Switch-a-View: View Selection Learned from Unlabeled In-the-wild Videos

Sagnik Majumder[1] Tushar Nagarajan[1] Ziad Al-Halah[2] Kristen Grauman[1]
[1]University of Texas at Austin [2]University of Utah

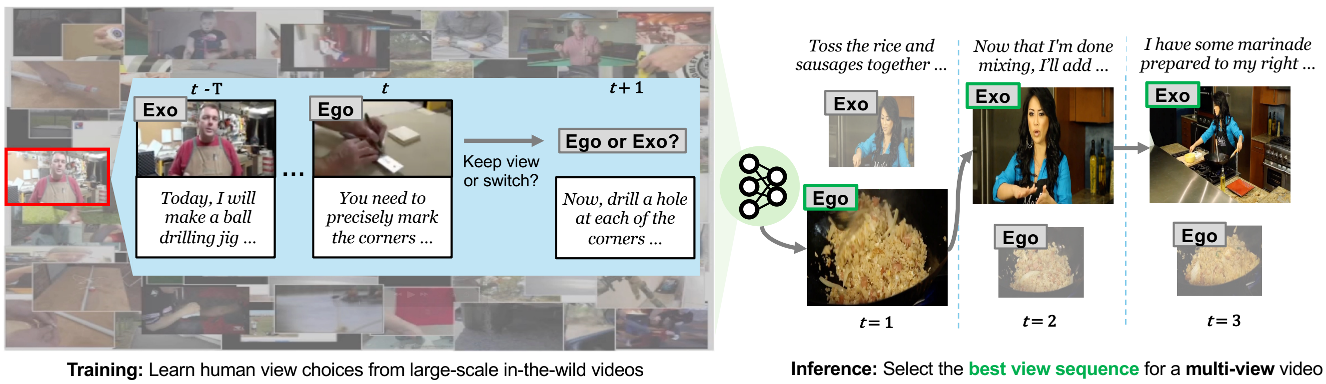

Figure 1. Given a multi-view narrated how-to video, can we select the sequence of camera viewpoints that best show the activity—automating the camerawork that is today done with manual editing? While direct supervision for this task is impractical, our SWITCH-A-VIEW approach shows how to learn *typical viewpoint choice patterns* from large-scale unlabeled in-the-wild instructional videos (left), then translate those patterns to novel multi-view videos (right), yielding an informative how-to that hops between the most useful ego/exo viewpoints.

Abstract

We introduce SWITCH-A-VIEW, *a model that learns to automatically select the viewpoint to display at each timepoint when creating a how-to video. The key insight of our approach is how to train such a model from unlabeled—but human-edited—video samples. We pose a pretext task that pseudo-labels segments in the training videos for their primary viewpoint (egocentric or exocentric), and then discovers the patterns between the visual and spoken content in a how-to video on the one hand and its view-switch moments on the other hand. Armed with this predictor, our model can be applied to new multi-view videos to orchestrate which viewpoint should be displayed when. We demonstrate our idea on a variety of real-world videos from HowTo100M and Ego-Exo4D, and rigorously validate its advantages.*

1. Introduction

Video is an amazing medium for communication, and today's widely used Internet platforms make it easy to create and share content broadly. Instructional or "how-to" video is particularly compelling in this setting: YouTube, TikTok, and similar sites have democratized the ability to share our talents with others, by both showing and telling how to perform some special skill. From how to plant a garden, how to make yogurt, how to fold origami, or how to give a dog a haircut, there is no shortage of how-to nuggets produced and consumed by users of many ages and backgrounds.

Creating an effective how-to video, however, is not trivial. From potentially hours of footage from multiple cameras capturing all aspects of the instructional activity, a creator needs to edit down to the essential steps of their demonstration *and* decide on the camera viewpoint (view) for each temporal segment that best reveals what they want to show. For example, when showing how to cut the dog's hair, the instructor might first appear standing beside the dog—the camera more distant—then the camera may zoom close up to her using scissors and describing how to trim near the ear, then zoom back out while she shows progress across the dog's body. How-to videos often exhibit this sequential mix of "exocentric" and "egocentric-like" viewpoints to effectively recap the procedure with clear visuals.

The status quo is to either orchestrate camerawork live while filming, or do post-recording editing among the multiple available cameras—both of which are labor intensive. Work in automatic cinematography [7, 15, 16, 22, 49, 56], though inspiring, relies on heuristics or domain-specific models that are not equipped to address automatic editing of video demonstrations. How could we train an "AI how-to cameraman", which, given a stream of two or more simultaneous camera views, could hop between them intelligently?

Supervising this learning task presents a problem. There are vast amounts of positive examples of well-edited how-to videos, but those edited results hide the "negatives"—the viewpoints that were *not* chosen for inclusion in the final video at any given time point. Those are left on the cutting room floor. This makes it unclear how to translate the editing patterns in in-the-wild edited video to new data.

To tackle this learning challenge, we design a pretext task for learning human view preferences from varying-view instructional videos on the Web. *Varying-view* means that the source training videos display an arbitrary number of view switches over the course of the video (e.g., from ego to exo and back as in our example above), and contain only *one* viewpoint at any time. We introduce a model called SWITCH-A-VIEW that learns from such data; it uses past frames in concert with the how-to narrations spoken by the demonstrator, which are widely available in instructional videos, to learn a binary classifier indicating whether the viewpoint is going to switch or not at the current time step. Then, we deploy this pretext-trained model in *multi-view*, narrated video settings with *limited* best view labels, and decide how to orchestrate the view selection of such videos over time. In this way, our approach captures the view-switch patterns from widely diverse unlabeled in-the-wild videos, then translates those trends to automatically direct the camerawork in new instances. See Fig. 1.

We train and evaluate our approach on HowTo100M [34], an extensive repository of real-world how-to videos, and further show generalization to multi-view Ego-Exo4D [18] videos. Our findings confirm that human judges exhibit substantial agreement on what constitutes a "best view" in a how-to video, establishing that it is possible to rigorously evaluate this task. Furthermore, our results show SWITCH-A-VIEW outperforms the state-of-the-art in multi-view video view selection [32] as well as proprietary VLMs like Gemini 2.5 Pro and GPT-4o [12, 23] and other baselines.

2. Related work

Automatic cinematography. In automatic cinematography, systems automate the process of creating an effective video presentation given a video scene, such as controlling camera movements, angles, and transitions. Prior work targets classroom environments [16, 21, 56], group activities [2], or (pseudo-)panoramic recordings [6, 7, 9, 15, 49, 50, 55]. Different from all of the above, we tackle view selection in multi-view *instructional* scenarios. Moreover, we seek a lighter-weight supervision solution: whereas prior work uses supervised discriminative methods requiring large-scale best view labels [7, 22, 49] or bootstraps view selector training using large-scale multi-view videos annotated with view-agnostic narrations [32], we aim to learn view selection from readily available in-the-wild *unlabeled* instructional videos. Furthermore, our model is multimodal, integrating both the video content as well as its transcribed speech.

View selection in active perception. More distant from our problem, work in active perception and robotics considers how agents can intelligently select their visual input stream. This includes next-best-view selection, where an embodied agent learns to actively place a camera for recognition [1, 8, 13, 24, 40] or segmentation [44, 45]. Whereas the objective in such work is to spend less time or compute for an agent to see sufficient content, our goal is instead to choose the sequence of informative camera views for human consumption, from among the available viewpoints.

Weak supervision from Web data. Large-scale instructional data from the Web has been shown to provide weak supervision for understanding instructional activities, by aligning frames [33] and narrations [29, 33] with their step descriptions from instructional Web articles (e.g., WikiHow), or through modeling the temporal order and interdependence of steps [3, 58, 59]. Unlike any of these methods, we tackle a distinct problem of weakly supervised view-switch detection in instructional videos, with the end goal of using the detector for view selection.

Video summarization. Temporal video summarization [4, 20, 35, 37, 42] entails creating a short but informative summary of a long video by subsampling keyframes or clips from it. While early methods are largely unsupervised [25, 30, 38, 47], more recent works derive supervision from manual labels [17, 19, 27, 41, 48, 57]. Limited work explores summarization in the context of multiple input videos [10, 14, 37, 43]. Video summarization and viewpoint selection are two entirely distinct tasks. Video summarization aims to downsample the video in time to the essential parts, whereas our task essentially requires downsampling the video in *space* to isolate the most informative viewpoint.

3. Approach

Our goal is to train a model to predict the "best view sequence" for multi-camera instructional videos — the sequence of camera viewpoints (views) that a human would most likely select to demonstrate an instructional activity (e.g., a close-up view of ingredients in a cooking video, moving to a wide-shot view when the chef speaks and gestures). To tackle this, we train a model for the proxy task of detecting "view switches" in varying-view instructional videos, which we then bootstrap to form a view selection model.

First, we formally define our pretext task (Sec. 3.1). Next, we describe how to source pseudo-labels for our pretext task by automatically classifying views in varying-view videos (Sec. 3.2). We then describe our method and how to train it to predict view-switches (Sec. 3.3). Finally, we describe

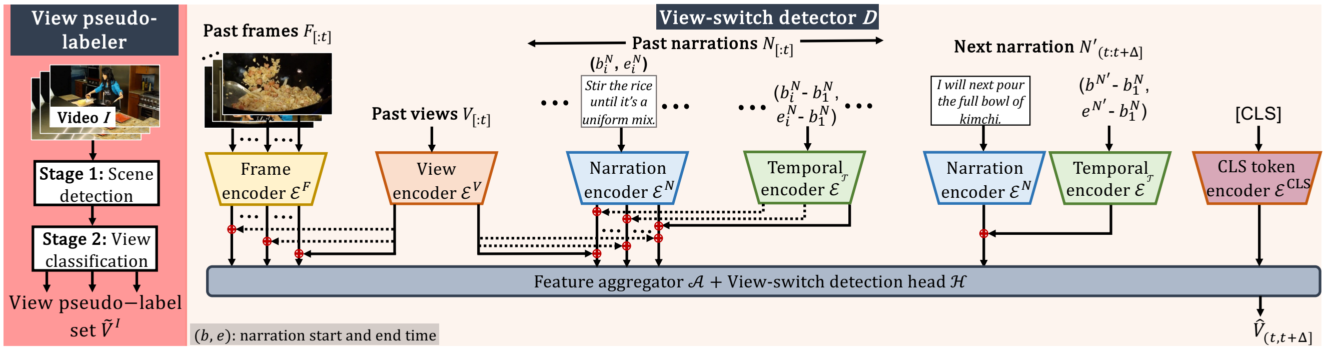

Figure 2. Given varying-view instructional videos—videos composed of a sequence of views chosen by human(s) to accurately show the instructional activity at all times— our goal is to train a view-switch detector D that can predict if the view should switch or not, at any time in a new video. Our hypothesis is that such a detector, when trained on large-scale and in-the-wild videos, can capture human view preferences and facilitate learning best view selection in multi-view settings with limited labels. However, such in-the-wild videos lack view labels. To train nevertheless, we propose an approach comprising **(a)** a view pseudo-labeler (left) that given a varying-view instructional video I, automatically classifies views in it and generates a pseudo-label set $\tilde{V}^I$, and **(b)** a view-switch detector D (right) that given the pseudo-labels $\tilde{V}^I$ and any time t in I, learns to predict the next view. The prediction is conditioned on the past frames, past narrations, and the next narration, where narrations are naturally occurring spoken content from the how-to demonstrator.

how our view-switch detector can bootstrap learning a view selection model (Sec. 3.4 and 3.5) with limited labels.

3.1. View-switch detection as a pretext task

We introduce our pretext task: view-switch detection in varying-view instructional videos. Consider a varying-view instructional video I, where the view changes back and forth over time between a close-up / *egocentric-like* (ego) view, and a wide shot / *exocentric-like* (exo) view.[1] This results in a sequence of varying views V. The instructional video also contains a sequence of narrations N, where each narration N_i has a start and end time, (b_i, e_i), and provides commentary transcribed to text. These narrations are free-form spoken language from the demonstrator, which capture their actions ("hammer the nail in there") as well as side comments ("sometimes I use my sander instead", "thanks for watching!").

We formulate the view-switch detection task as a two-class *view prediction* problem, where at any time t in the video, the model must detect if the view should be of type *ego* or *exo*, to best showcase the activity over the *next* Δ seconds. More specifically, we require a model D that predicts the human-preferred view $V_{(t,t+\Delta]}$ given the past video, narrations and views, as well as the next narration. Formally,

$$D(F_{[:t]}, N_{[:t]}, V_{[:t]}, N'_{(t,t+\Delta]}) = V_{(t,t+\Delta]},$$

where $F_{[:t]}$ is the past frames, $N_{[:t]}$ is the past narrations and $V_{[:t]}$ is the past views. $N'_{(t,t+\Delta]}$ is the next narration, if it overlaps with the prediction interval, and an empty string otherwise. Importantly, this formulation provides a path from the *next-view prediction task* to the *view-switch task*: since the most recent past view is observed, estimating the desired next view—and comparing it with the latest past view—is equivalent to predicting whether the view switched.

While past narrations provide high-level cues about past activity steps, past frames offer more fine-grained information about the steps and how they were viewed. They together form the past context that can help anticipate the next view. The next narration is essential to disambiguate between various potential actions that the demonstrator may do next, and the language directly hints at the appropriate views (e.g., the person says "next, let's take a closer look at ..." suggesting an ego view). Thus, combining these inputs will offer valuable cues to our detector.[2]

Critically, we aim to train this detector on large-scale, in-the-wild instructional videos [34, 51]. We show that training for this pretext task can enable view selection models for multi-camera settings, with *limited* supervision. In short, representations developed to detect when to "switch view" can be repurposed with minimal modification to select the "best view" to switch to, since they contain rich knowledge of human-selected view-switch patterns in a large variety of in-the-wild scenarios. Next, we show how to source pseudo-labels to train such models.

3.2. Sourcing "view-switch" pseudo-labels

Instructional videos [34, 51] are an ideal source of varying-view data, however they do not come paired with information about what camera viewpoint is chosen for each segment. We therefore design a strategy to automatically identify and pseudo-label their underlying view sequences. We do this in two stages (Fig. 2 left).

[1] We adopt this ego/exo view taxonomy given their importance and prevalence in instructional video datasets [18, 34, 51].

[2] Note that next-step narrations are also available at inference time, when we have multi-view video content and a full narration track, and we aim to perform view selection.

First, given a video I, we use an off-the-shelf scene detector (PySceneDetect [5]) to compute scene boundaries. Using this, we split the video into a sequence of contiguous shots. Next, we classify each frame in the video using a pre-trained ego vs. exo view classifier, and then aggregate the class predictions into a shot-level pseudo-label. Specifically, given a shot from I, we first split it into a sequence of fixed-length clips. Next, we feed each clip to the view classifier that produces the probability that the clip is from an ego vs. exo view. We then compute the pseudo-label for the whole shot by averaging the view probabilities across all its clips. We repeat these steps for all shots, and assign each frame in I the same pseudo-label as the shot it lies in, to finally obtain a pseudo-label set $\hat{V}^I$. Combining the classifier with the scene detector reduces the overall noise in the pseudo-labels due to classification failures at scene boundaries. We use a learned model [28] for ego-exo view classification, trained on the Charades-Ego [46] dataset. See Supp. for details.

3.3. View-switch detector design

Given a video I and any time t in it, our view-switch detector D must successfully predict the view for the future time interval $(t, t+\Delta]$. It must do so using the frames, narrations and views from the past, and also the next narration, if it overlaps with the prediction interval (c.f. Sec. 3.1). See Fig. 2 right. In the following, we provide details on how our method extracts features from each input and then aggregates them for making a view prediction.

Frame encoding. We begin by using a frame encoder $\mathcal{E}^F$ to embed the past frames $F_{[:t]}$ and produce a visual feature sequence f, where each frame F_i has a feature f_i. We further enhance each feature f_i by using a viewpoint encoder $\mathcal{E}^V$ to embed the corresponding view V_i^F into a view feature and adding it to f_i. We also encode frame F_i's temporal position relative to the start time of the most recent narration using a temporal encoder $\mathcal{E}^\mathcal{T}$ and add the encoding to the enhanced frame feature. Producing a feature per frame and augmenting it with view and temporal information helps us create a *fine-grained*, and *view-* and *temporally-aware* representation predictive of the next view.

Narration encoding. Next, we encode each past narration from $N_{[:t]}$, and the next narration $N'_{(t,t+\Delta]}$ by using an LLM encoder. This generates a text feature sequence n for the past narrations and a single text feature n' for the next narration.

Similar to our encoding of past frames, we also make the features for past narrations *view-aware*. To do so, we first produce a per-view count of the frames that lie in the interval of each past narration N_i. We then estimate the dominant viewpoint for the narration—called narration view, henceforth—by setting it to the most frequent view per the per-view frame count. Next, we use our view encoder $\mathcal{E}^V$ to embed the narration view into a view feature. Finally, we update the narration feature n_i by adding it with the view feature.

Moreover, for both past and next narrations, we provide their temporal information to our model so that it can infer the alignment between the frames and the narrations, and use it to improve its cross-modal reasoning. To this end, we first normalize the start and end time pair for each past narration N_i and next narration N', to be relative to the start time of the first past narration. We then compute the mean time of each pair. These means convey the temporal locations of the narrations relative to each other. Next, we encode each relative mean with the temporal encoder $\mathcal{E}^\mathcal{T}$ and obtain a temporal feature. Finally, we update the narration features, n and n', by adding them with their temporal features.

Feature aggregation and view classification. To aggregate the visual and narration features, we first add modality features to the frame features f, and narration features, n and n', respectively. These are modality-specific learnable embeddings that help distinguish between the visual and text modalities, and successfully do cross-modal reasoning.

We also introduce a [CLS] token in our model, and embed it with an encoder $\mathcal{E}^{\text{CLS}}$ to produce a feature c, so that the output of our feature aggregator, which corresponds to the [CLS] token, can be used to estimate the next view. Next, we feed the frame features f, the past narration features n, the next narration feature n', and the [CLS]-token feature c into a feature aggregator $\mathcal{A}$. $\mathcal{A}$ comprises a transformer [53] encoder that performs self-attention on all features and extracts multi-modal cues that are predictive of the next view. Finally, we take the output feature of $\mathcal{A}$, which corresponds to the [CLS] token, and pass it to a view classification head $\mathcal{H}$ to get an estimate $\hat{V}_{(t,t+\Delta]}$ of the next view $V_{(t,t,+\Delta]}$. Formally,

$$\hat{V}_{(t,t+\Delta]} = \mathcal{H}(\mathcal{A}(f, n, n', c)[j_{\text{CLS}}]), \tag{1}$$

where j_{CLS} is the feature index for the [CLS] token.

3.4. Repurposing switch detection for view selection

Recall that in view selection, given a multi-view instructional video I and any time t in it, the goal is to predict the view that is preferred by humans for showing the activity in an interval $[t, t+\Delta]$. We introduce a view selector S for tackling this task. S is a modification of our view-switch detector D, such that S additionally has access to the *frames from the simultaneously captured ego and exo views* during the prediction interval $[t, t+\Delta]$.

To this end, we first use our frame encoder $\mathcal{E}^F$ to embed the ego frames $F^G_{[t,t+\Delta]}$ and exo frames $F^X_{[t,t+\Delta]}$ into visual features f^G and f^X, respectively. Next, we append f^G and f^X to the input sequence of our feature aggregator $\mathcal{A}$. Finally, we treat $\mathcal{A}$'s output feature for its [CLS] token input, as a representation of the best view for $[t, t+\Delta]$, and feed it to the detector's view classification head $\mathcal{H}$ to get an estimate $\ddot{V}_{[t,t+\Delta]}$ of the best view $V_{[t,t+\Delta]}$.

To learn view selection we initialize S with our detector's parameters, trained on the view-switch detection task, and finetune it using a small set of samples labeled for view selection. This design enables us to effectively use the knowledge from pretraining and learn view selection with limited labels. Next, we provide details for training and finetuning.

3.5. Model training objective

We train our view-switch detector D with a view classification loss $\mathcal{L}^D$. We set $\mathcal{L}^D$ to

$$\mathcal{L}^D = \mathcal{L}_{\text{CE}}(\hat{V}_{(t,t+\Delta]}, \tilde{V}_{(t,t+\Delta]}), \tag{2}$$

where $\hat{V}_{(t,t+\Delta]}$ is our estimated view (c.f. Sec. 3.3) and $\tilde{V}_{(t,t+\Delta]}$ is the pseudo-label from our view pseudo-labeler (c.f. Sec. 3.2).

To train our view selector S, we obtain a *small* training set of best view labels, B, such that $B = \{V_{[t_1,t_1+\Delta]}, \ldots, V_{[t_W,t_W+\Delta]}\}$, and W is the label count in B. For each best view label $V_{[t_w,t_w+\Delta]} \in B$, and the corresponding view estimate $\ddot{V}_{[t_w,t_w+\Delta]}$, per our view selector S (c.f. Sec. 3.4), we set our view selection loss $\mathcal{L}^S$ to a cross-entropy loss, such that

$$\mathcal{L}^S = \mathcal{L}_{\text{CE}}(\ddot{V}_{(t_w,t_w+\Delta]}, V_{(t_w,t_w+\Delta]}). \tag{3}$$

Once trained, our framework can accurately choose the preferred view in novel multi-view videos.

4. Datasets and annotations

Datasets. We use two datasets in our experiments. **HT100M [34]** is a large-scale dataset of narrated, in-the-wild instructional videos. These videos are view-varying in nature, and the views can be broadly categorized as ego or exo. This, along with the diversity and realism of HT100M, makes it ideal for our view-switch detection task. **Ego-Exo4D [18]** contains multi-view videos, where each video is captured with five time-synced cameras—one is an ego camera worn by a human performing an instructional activity, and the other four are stationary exo cameras placed around the scene. Moreover, the narrate-and-act (N&A) subset of Ego-Exo4D has videos of humans narrating and performing an activity, where the narrations are free-form and match in style with HT100M, making it compatible with our task of view selection with limited labels.

Training data. To train the view-switch detector, we use 3,416 hours of HT100M videos spanning a diverse set of activities (cooking, DIY, household, etc.) and pseudo-label shots from these videos (c.f. Sec. 3.2). See Supp. for details.

Evaluation data. For evaluation, we use both HT100M and Ego-Exo4D [18], where the view-switch detection evaluation on Ego-Exo4D is *zero-shot*. While the training sets are automatically generated and pseudo-labeled, we ensure a gold-standard *test set* free of noise by manually annotating videos for our tasks. To this end, we recruit trained annotators to manually annotate the view types for HT100M and the human-preferred views for Ego-Exo4D, as follows.

For HT100M, we identify 975 hours of videos that do not overlap with our train videos above. We segment 4,487 fixed-length clips, each with length set to the prediction interval Δ (c.f. Sec. 3.1). Next, we ask trained annotators to label these clips as either ego or exo. See Supp. for full annotation instructions and more details.

For Ego-Exo4D, we create a test set containing 2.7 hours of N&A videos spanning six activity categories (cooking, bike repair, rock climbing, dancing, soccer, basketball). For each video, we use its "best-exo-view" annotation from Ego-Exo4D to generate an ego-exo view pair comprising the single ego and the best exo view. As before, we create Δ length clips from each view. We then couple the pair with its closest atomic activity description (time-stamped manual descriptions of the camera wearer's activity [18]) and ask our annotators to label the *view* between the two that *best* demonstrates the activity described in the narration (see Supp. Fig. 3). Importantly, this means that annotators specifically select the "best" view as the one that most clearly illustrates the *current actions of the camera wearer*, consistent with our how-to video view selection goal.

Annotator agreement on best view. To ensure annotation quality for *both* datasets, in addition to providing detailed annotation guidelines and concrete examples (available in Supp.), we require annotators to take qualifiers with stringent passing criteria and we solicit 9 annotators' responses for each instance. We accept an annotation only if the inter-annotator agreement is at least 78%, meaning at least 7 out of 9 annotators agree. This resulted in a Cohen's kappa coefficient [11] of 0.65 for HT100M and 0.70 for Ego-Exo4D—both of which constitute "substantial" agreement [26]. This solid agreement assures the quality of our test set; despite there being some room for subjectivity in deciding the best view for a how-to, this data shows human judges are indeed able to substantially agree.

This results in a final total of 3,151 and 5,049 test instances (fixed-length clip-narration pairs from above), sampled from 3,677 HT100M and 33 Ego-Exo4D test videos, respectively. In Supp. we filter with even higher agreement thresholds, yielding even more selective (but smaller) test sets; trends for our method vs. baselines remain consistent.

Data for view selection with limited labels. We train and evaluate our view selector on a small dataset comprising Ego-Exo4D [18] videos. For our training data, we follow

Model	HowTo100M [34] Accuracy	AUC	AP	Ego-Exo4D [18] Accuracy	AUC	AP
All-ego/exo	50.0	50.0	50.0	50.0	50.0	50.0
Random	52.0	52.0	51.0	49.3	49.3	49.7
Last-frame	42.3	42.3	53.4	50.0	50.0	50.0
First-person pronoun detector	47.8	47.8	46.4	50.3	50.3	50.1
Retrieval [54]-F	53.4	53.4	53.2	**52.6**	52.6	53.6
Retrieval [54]-N	52.1	52.1	51.8	52.0	52.0	50.6
Retrieval [54]-N'	52.6	52.6	52.9	52.1	52.1	52.6
SWITCH-A-VIEW (Ours)	**59.4**	**63.8**	**60.5**	51.2	**56.4**	**55.4**

Table 1. View-switch detection results. Evaluation on Ego-Exo4D [18] is zero-shot. All values are in %, and higher is better.

our annotation protocol for evaluating view-switch detection on Ego-Exo4D, and collect view annotations for a total of 3.5 hours of training videos. This results in a total of 6,634 train instances. For evaluation, we use our test set from view-switch detection. This reuse is possible since a label indicates both the type (ego/exo) of the desired next view for view-switch detection as well as the desired current view for view selection. Train and test videos for this task are disjoint. See Supp. for details.

5. Experiments

Implementation. We set the durations of past frames to 8 seconds—corresponding to 0.23 and 2.31 switch(es) per second for HT100M and Ego-Exo4D, respectively—and past narrations to 32 seconds, and the prediction interval to $\Delta = 2$ seconds. We set the sample count for view selection to $W = 5000$. We evaluate view-switch detection on HowTo100M [34] by obtaining the views for the past frames (c.f. Sec. 3.3) from our pseudo-labeler. For Ego-Exo4D, we adopt a teacher-forcing setup and evaluate both tasks by using the ground-truth annotations for past frames and views. We implement our view-switch detector D and view selector S using the DINOv2 [36] encoder for our frame encoder $\mathcal{E}^F$, the Llama 2 [52] encoder for our narration encoder $\mathcal{E}^N$, a 8-layer transformer encoder [53] for our feature aggregator $\mathcal{A}$, a 2-layer MLP for the view classification head $\mathcal{H}$, and learnable embedding layers for our view encoder $\mathcal{E}^V$ and temporal encoder $\mathcal{E}^\mathcal{T}$.

Baselines. We provide strong baselines comprising SOTA models and representations, as well as relevant heuristics. For *view-switch detection*, we compare against

- **InternVideo2 retrieval [54]:** a set of baselines that given the most recent past frame (**Retrieval [54]-**F), most recent past narration (**Retrieval [54]-**N), or next narration (**Retrieval [54]-**N'), first encodes [54] them into fine-grained features that capture multi-frame temporal contexts, then uses feature similarity to retrieve a nearest neighbor of the same input type from the train set, and finally outputs the next view for F or N, or the corresponding view for N', as its prediction.[3]

Model	Accuracy	AUC	AP
Human performance (Upper bound)	82.3	83.5	81.7
All-ego/exo	50.0	50.0	50.0
Random	49.3	49.3	49.7
Last-frame	50.0	50.0	50.0
First-person pronoun detector	50.3	50.3	50.1
Retrieval [54]-F	52.3	52.3	53.6
Retrieval [54]-N	51.9	51.9	51.0
Retrieval [54]-N'	52.4	52.4	52.4
View-narration [54] Similarity	52.5	52.4	53.9
Finetuned X-CLIP [7]			
Random negative sampling	52.1	52.0	53.1
Text-conditioned negative sampling	52.8	52.7	53.6
Proprietary VLMs			
Gemini 2.5 Pro	51.2	51.2	51.0
GPT-4o	53.3	53.3	52.3
LangView [32]			
-smallData	52.1	52.6	53.2
-bigData (privileged)	53.3	54.8	54.5
Ours w/o pretraining	50.1	51.6	51.3
SWITCH-A-VIEW (Ours)	**54.0**	**57.3**	**56.0**

Table 2. Results and ablation for view selection with limited labels. All values are in %; higher is better. Significance $p \leq 0.05$.

- **All-ego, All-exo, Random, Last-frame**: these are heuristics that use the ego view (**All-ego**), the exo view (**All-exo**), a randomly chosen (**Random**) view, or the view of the most recent past frame (**Last-frame**), as their prediction.
- **First-person pronoun detector**: a heuristic that predicts exo when it detects first-person pronouns like “I", “We", “My" or “Our" in the next narration, as human editors often use a wide shot that reveals their face or full body, when using such pronouns.

For *view selection with limited labels*, in addition to the baselines listed above, we compare against the following:

- **LangView [32]**: a SOTA view selector that uses multi-view videos and human-annotated narrations for weakly supervised pretraining. We finetune this model with our Ego-Exo4D labels (Sec. 4). We evaluate two versions of this baseline: LangView-*bigData* and LangView-*smallData*, which use large-scale Ego-Exo4D [32] videos, and our same small subset (Sec. 4), respectively, for pretraining. Note that the *bigData* variant enjoys access to **98**× more training samples than our method, an advantage for the baseline.
- **View-narration [54] Similarity (VN-Sim)**: separately computes the cosine similarity between the InternVideo2 features [54] for each view and the next narration, and picks the view most similar to the narration.
- **Finetuned X-CLIP [31]**: a finetuned CLIP [39]-style model that aligns the frames from the target view and

[3] See Supp. for parallel evaluation with CLIP [39]-style encoders, which generally underperformed InternVideo2 [54] encoders.

the future narration. We explore two negative sampling strategies when finetuning: random and text-conditioned.

- **Proprietary VLMs**: we feed Gemini 2.5 Pro [12] and GPT-4o [23] all our view selection inputs and task them with choosing the best view by providing a text prompt similar to our guidelines for collecting human annotations (see Sec. 5 and Supp.).

LangView evaluates how our model fares against SOTA view selection, while the retrieval, view-narration and finetuned CLIP [39]-style baselines analyze whether SOTA video-language embeddings, whether frozen or finetuned, are sufficient for this task. The heuristics verify the challenging nature of the tasks. The proprietary VLMs evaluate if employing large-scale generalist models is enough.

Evaluation metrics. We consider three metrics: 1) **Accuracy**, which directly measures the agreement between our predictions and labels; 2) **AUC**, the area under the ROC curve; and 3) **AP**, the average precision (AP) of the precision vs. recall curve. We use AUC and AP to account for the possible class imbalance[4] in our collected annotations. Moreover, for each metric, we separately compute its value for the *same-view* and *view-switch* instances in our test sets, and report the mean. This lets us account for differences in the same-view and view-switch frequency, and obtain unbiased performance measures.

View-switch detection. In Table 1, we report our view-switch detection results. The heuristics generally perform the worst on both datasets, underlining the challenging nature of the task. The Retrieval [54] baselines improve over them, indicating that our model inputs do provide cues about the view type. Among the Retrieval baselines, retrieving using the most recent past frame performs the best, showing that the past frames offer fine-grained task-relevant information beyond the narration words. Moreover, retrieving with the next narration is better than retrieving with the most recent past narration, revealing that the next narration carries more pertinent details about the desired view. This is likely because the next narration is better aligned with the time interval for which the view is being predicted.

Our method outperforms all baselines on both datasets, with the AUC margin over the best baseline, Retrieval [54]-F, being as high as 10.4% on HowTo100M (HT100M) [34] and 3.8% on Ego-Exo4D [18]. Our improvement over the Retrieval baselines show that computing feature [54]-level similarities are not enough for this task. Instead, learning it by leveraging complementary cues from both narrations and frames is critical. Moreover, our zero-shot results on Ego-Exo4D speak to our model's efficacy vis-a-vis learning human view patterns from large-scale and in-the-wild videos, which generalize to different scenarios, without any training.

[4]Same-view instance count = 1.6x view-switch instance count for HT100M and 3.9x for Ego-Exo4D

	HowTo100M [34]			Ego-Exo4D [18]		
Model	Accuracy	AUC	AP	Accuracy	AUC	AP
N-only	53.5	54.4	52.3	50.0	48.7	49.0
N'-only	55.4	57.8	56.2	49.8	49.8	50.0
F-only	53.3	54.5	54.7	51.0	53.4	53.2
(F, N')-only	55.5	60.1	<u>58.1</u>	**52.1**	54.2	52.6
(N, N')-only	<u>57.5</u>	59.3	56.6	50.0	53.0	52.6
(F, N)-only	56.0	<u>60.9</u>	57.4	51.8	<u>54.9</u>	<u>54.2</u>
Ours	**59.4**	**63.8**	**60.5**	<u>51.2</u>	**56.4**	**55.4**

Table 3. Ablation study for view-switch detection. All values are in %, and higher is better. Significance $p \leq 0.05$.

View selection. Table 2 shows our results on view selection with limited labels. For the heuristics and Retrieval [54] baselines, we observe the same performance trends as view-switch detection. The View-narration [54] Similarity (VN-Sim) baseline marginally improves over these methods, indicating the frames from candidate views when combined with the corresponding narration (N') provide direct cues about the preferred view. LangView [32]'s results benefit from its language-guided training, generally outperforming VN-Sim.

Our method significantly improves over all baselines, with the AUC margin over the best baseline, LangView [32]-bigData, being 2.5%. Our gains over VN-Sim and Fine-tuned X-CLIP [31] underscore that using feature similarity to match the activity described in the next narration with a candidate view does not suffice, and instead a model like ours, which can leverage multi-modal cues from the combination of both past and candidate frames, and past and next narrations, is valuable for this task. Our improvement over the proprietary VLMs—despite their much larger size and training data—shows that task-specific experts are necessary to tackle our challenging task. Training our model from scratch with only the small set of best view labels ("ours w/o pretraining") is significantly weaker, showing that our view-switch pretraining idea is doing the heavy lifting.

Our gains over the SOTA LangView [32] show that learning view selection from language is less effective than that from large-scale human-edited videos, even when the videos and language are available at scale (bigData). Moreover, the insights of LangView and this work are complementary. We find if we fine-tune SWITCH-A-VIEW with LangView's narration-based pseudo-labels, in *addition* to our labels (Sec. 4), we achieve further gains. See Supp. for details.

Ablations. Table 3 shows our ablation results for view-switch detection. Dropping any one input to our model degrades performance, indicating that each input plays a role. Dropping two inputs hurt the performance even more, showing that more inputs are better in any combination, suggesting our model design extracts complementary cues from them in all configurations. Moreover, using past frames instead of narrations improves performance, re-affirming

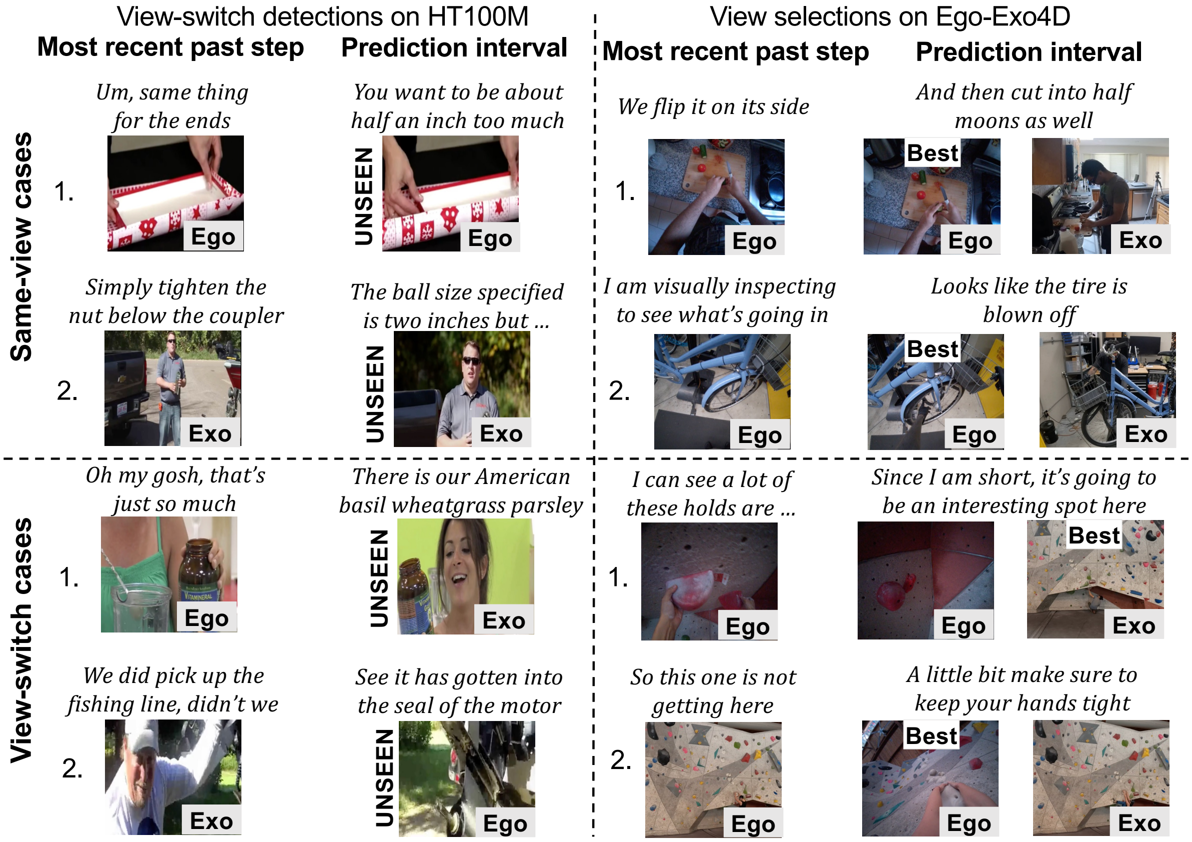

Figure 3. Left: successful view-switch detections by our model on same-view (**top**) and view-switch cases (**bottom**). Our model correctly detects view switches by anticipating the next step using past frames (same-view sample 1, and view-switch sample 2) or leveraging the content of the next narration (same-view sample 2, and view-switch sample 1 and 2). **Right:** successful view selections by our model on same-view (**top**) and view-switch cases (**bottom**). For view selection as well, our model can predict the desired next view by relying on the next narration (same-view sample 1, and view-switch sample 1 and 2), or anticipate it using the past narrations (same-view sample 1 and 2), or the past frames (same-view sample 1). These examples show that all three inputs play a role in our model predictions.

that vision provides fine-grained features necessary for high performance. Finally, using N' instead of N improves performance in some cases, showing the next narration's role.

See Supp. for more analysis, including the effect of the past frame and narration durations, and sample count on model performance, and its scenario-level breakdown.

Qualitative examples. Fig. 3-left shows our model's successful view-switch detections on both same-view (**top**) and view-switch cases (**bottom**); see caption for details. We also notice some common **failure modes** with our model. For view-switch detection, our model sometimes fails when there is no next narration overlapping with the prediction interval, and neither the past frames nor narrations are predictive of the next view. In another failure type, the past views are wrongly categorized by our pseudo-labeler for HowTo100M [34] or by professional annotators for Ego-Exo4D [18]. This leads to our model getting confused and predicting the wrong next view. For view selection, in addition to these failures, our model can fail when both views look equally good. See Supp. for video examples.

6. Conclusion and future work

We introduced an approach for learning to select views from instructional video by bootstrapping human-edited (but unlabeled) in-the-wild content. Results show the method's efficacy and set the benchmark for this new task.

A potential limitation of our model is its clip-level predictions, which can lead to rapid switches between viewpoints over time. While hard cuts are in fact necessary at times to maximize informativeness, the trade-off between view information and perceived viewing ease is interesting future work. Other challenges uncovered by our work are the distribution gap between between edited in-the-wild and multi-view videos and the complexity of learning view selection from limited labels. In addition, we plan to generalize to *continuous* view selection, potentially by integrating ideas from new view synthesis, and we will explore modeling user attention for personalized view selection.

Acknowledgements

UT Austin is supported in part by the UT Austin IFML NSF AI Institute and the UT Austin MLL Center for Generative AI. We would also like to thank Zihui Xue for suggesting the PySceneDetect scene detector and helpful discussions regarding the view classifier during the early stages of designing our view pseudo-labeler.

References

[1] Phil Ammirato, Patrick Poirson, Eunbyung Park, Jana Košecká, and Alexander C Berg. A dataset for developing and benchmarking active vision. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1378–1385. IEEE, 2017. 2

[2] Ido Arev, Hyun Soo Park, Yaser Sheikh, Jessica Hodgins, and Ariel Shamir. Automatic editing of footage from multiple social cameras. *ACM Trans. Graph.*, 33(4), 2014. 2

[3] Kumar Ashutosh, Santhosh Kumar Ramakrishnan, Triantafyllos Afouras, and Kristen Grauman. Video-mined task graphs for keystep recognition in instructional videos. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[4] Taivanbat Badamdorj, Mrigank Rochan, Yang Wang, and Li Cheng. Contrastive learning for unsupervised video highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14042–14052, 2022. 2

[5] Brandon Castellano. Pyscenedetect. `https://github.com/Breakthrough/PySceneDetect`. 4

[6] Seunghoon Cha, Jungjin Lee, Seunghwa Jeong, Younghui Kim, and Junyong Noh. Enhanced interactive 360° viewing via automatic guidance. *ACM Trans. Graph.*, 39(5), 2020. 2

[7] Jianhui Chen, Keyu Lu, Sijia Tian, and Jim Little. Learning sports camera selection from internet videos. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1682–1691. IEEE, 2019. 1, 2

[8] Ricson Cheng, Ziyan Wang, and Katerina Fragkiadaki. Geometry-aware recurrent neural networks for active visual recognition. *Advances in Neural Information Processing Systems*, 31, 2018. 2

[9] Shih-Han Chou, Yi-Chun Chen, Kuo-Hao Zeng, Hou-Ning Hu, Jianlong Fu, and Min Sun. Self-view grounding given a narrated 360 {\deg} video. *arXiv preprint arXiv:1711.08664*, 2017. 2

[10] Wen-Sheng Chu, Yale Song, and Alejandro Jaimes. Video co-summarization: Video summarization by visual co-occurrence. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3584–3592, 2015. 2

[11] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960. 5

[12] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 2, 7

[13] Ruoyi Du, Wenqing Yu, Heqing Wang, Ting-En Lin, Dongliang Chang, and Zhanyu Ma. Multi-view active fine-grained visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1568–1578, 2023. 2

[14] Mohamed Elfeki, Liqiang Wang, and Ali Borji. Multi-stream dynamic video summarization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 339–349, 2022. 2

[15] J. Foote and D. Kimber. Flycam: practical panoramic video and automatic camera control. In *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No.00TH8532)*, pages 1419–1422 vol.3, 2000. 1, 2

[16] Michael Gleicher and James Masanz. Towards virtual videography (poster session). In *Proceedings of the Eighth ACM International Conference on Multimedia*, page 375–378, New York, NY, USA, 2000. Association for Computing Machinery. 1, 2

[17] Boqing Gong, Wei-Lun Chao, Kristen Grauman, and Fei Sha. Diverse sequential subset selection for supervised video summarization. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014. 2

[18] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. *arXiv preprint arXiv:2311.18259*, 2023. 2, 3, 5, 6, 7, 8

[19] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *Computer Vision – ECCV 2014*, pages 505–520, Cham, 2014. Springer International Publishing. 2

[20] Bo He, Jun Wang, Jielin Qiu, Trung Bui, Abhinav Shrivastava, and Zhaowen Wang. Align and attend: Multimodal summarization with dual contrastive losses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14867–14878, 2023. 2

[21] Rachel Heck, Michael Wallick, and Michael Gleicher. Virtual videography. In *Proceedings of the 14th ACM International Conference on Multimedia*, page 961–962, New York, NY, USA, 2006. Association for Computing Machinery. 2

[22] Hou-Ning Hu, Yen-Chen Lin, Ming-Yu Liu, Hsien-Tzu Cheng, Yung-Ju Chang, and Min Sun. Deep 360 pilot: Learning a deep agent for piloting through 360deg sports videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3451–3460, 2017. 1, 2

[23] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 2, 7

[24] Dinesh Jayaraman and Kristen Grauman. End-to-end policy learning for active visual categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1601–1614, 2019. 2

[25] Hong-Wen Kang, Y. Matsushita, Xiaoou Tang, and Xue-Quan Chen. Space-time video montage. In *2006 IEEE Computer*

Society Conference on Computer Vision and Pattern Recognition (CVPR'06), pages 1331–1338, 2006. 2

[26] J Landis and G Koch. The measurement of observer agreement for categorical data. *Biometrics*, 1977. 5

[27] Yandong Li, Liqiang Wang, Tianbao Yang, and Boqing Gong. How local is the local diversity? reinforcing sequential determinantal point processes with dynamic ground sets for supervised video summarization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 151–167, 2018. 2

[28] Yanghao Li, Tushar Nagarajan, Bo Xiong, and Kristen Grauman. Ego-exo: Transferring visual representations from third-person to first-person videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6943–6953, 2021. 4

[29] Xudong Lin, Fabio Petroni, Gedas Bertasius, Marcus Rohrbach, Shih-Fu Chang, and Lorenzo Torresani. Learning to recognize procedural activities with distant supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13853–13863, 2022. 2

[30] Zheng Lu and Kristen Grauman. Story-driven summarization for egocentric video. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, page 2714–2721, USA, 2013. IEEE Computer Society. 2

[31] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *Proceedings of the 30th ACM international conference on multimedia*, pages 638–647, 2022. 6, 7

[32] Sagnik Majumder, Tushar Nagarjan, Ziad Al-Halah, Reina Pradhan, and Kristen Grauman. Which viewpoint shows it best? Language for weakly supervising view selection in multi-view videos. In *CVPR*, 2025. 2, 6, 7

[33] Effrosyni Mavroudi, Triantafyllos Afouras, and Lorenzo Torresani. Learning to ground instructional articles in videos through narrations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15201–15213, 2023. 2

[34] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2630–2640, 2019. 2, 3, 5, 6, 7, 8

[35] Medhini Narasimhan, Arsha Nagrani, Chen Sun, Michael Rubinstein, Trevor Darrell, Anna Rohrbach, and Cordelia Schmid. Tl; dw? summarizing instructional videos with task relevance and cross-modal saliency. In *European Conference on Computer Vision*, pages 540–557. Springer, 2022. 2

[36] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 6

[37] Rameswar Panda and Amit K Roy-Chowdhury. Collaborative summarization of topic-related videos. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pages 7083–7092, 2017. 2

[38] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *Computer Vision – ECCV 2014*, pages 540–555, Cham, 2014. Springer International Publishing. 2

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 6, 7

[40] Santhosh K Ramakrishnan, Dinesh Jayaraman, and Kristen Grauman. Emergence of exploratory look-around behaviors through active observation completion. *Science Robotics*, 4 (30):eaaw6326, 2019. 2

[41] Mrigank Rochan, Linwei Ye, and Yang Wang. Video summarization using fully convolutional sequence networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 347–363, 2018. 2

[42] Mrigank Rochan, Mahesh Kumar Krishna Reddy, Linwei Ye, and Yang Wang. Adaptive video highlight detection by learning from user history. In *Computer Vision – ECCV 2020*, pages 261–278, Cham, 2020. Springer International Publishing. 2

[43] Abhimanyu Sahu and Ananda S. Chowdhury. Shot level egocentric video co-summarization. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2887–2892, 2018. 2

[44] Soroush Seifi and Tinne Tuytelaars. Attend and segment: Attention guided active semantic segmentation. In *European Conference on Computer Vision*, pages 305–321. Springer, 2020. 2

[45] Soroush Seifi, Abhishek Jha, and Tinne Tuytelaars. Glimpse-attend-and-explore: Self-attention for active visual exploration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16137–16146, 2021. 2

[46] Gunnar A Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Charades-ego: A large-scale dataset of paired third and first person videos. *arXiv preprint arXiv:1804.09626*, 2018. 4

[47] Michael Smith and Takeo Kanade. Video skimming for quick browsing based on audio and image characterization. Technical Report CMU-CS-95-186, Pittsburgh, PA, 1995. 2

[48] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5179–5187, 2015. 2

[49] Yu-Chuan Su, Dinesh Jayaraman, and Kristen Grauman. Pano2vid: Automatic cinematography for watching 360 videos. In *Asian Conference on Computer Vision*, pages 154–171. Springer, 2016. 1, 2

[50] Xinding Sun, J. Foote, D. Kimber, and B. S. Manjunath. Region of interest extraction and virtual camera control based on panoramic video capturing. *Trans. Multi.*, 7(5):981–990, 2005. 2

[51] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1207–1216, 2019. 3

[52] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 6

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4, 6

[54] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. Internvideo2: Scaling foundation models for multimodal video understanding. In *European Conference on Computer Vision*, pages 396–416. Springer, 2024. 6, 7

[55] Bo Xiong and Kristen Grauman. Snap angle prediction for 360° panoramas. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2

[56] Cha Zhang, Yong Rui, Jim Crawford, and Li-Wei He. An automated end-to-end lecture capture and broadcasting system. *ACM Trans. Multimedia Comput. Commun. Appl.*, 4(1), 2008. 1, 2

[57] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Summary transfer: Exemplar-based subset selection for video summarization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1059–1067, 2016. 2

[58] Yiwu Zhong, Licheng Yu, Yang Bai, Shangwen Li, Xueting Yan, and Yin Li. Learning procedure-aware video representation from instructional videos and their narrations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14825–14835, 2023. 2

[59] Honglu Zhou, Roberto Martín-Martín, Mubbasir Kapadia, Silvio Savarese, and Juan Carlos Niebles. Procedure-aware pretraining for instructional video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10727–10738, 2023. 2