

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

Supplementary Material

Sagnik Majumder¹ Tushar Nagarajan¹ Ziad Al-Halah² Kristen Grauman¹

¹University of Texas at Austin ²University of Utah

In this supplementary material we provide additional details about:

- Video for qualitatively illustrating of our main idea and also qualitatively evaluating of our view-switch detections and view selections (Sec. 1), as mentioned in ‘Qualitative examples’ in Sec. 5 in main
- Analysis of the impact of our shot-level pseudo-labeling on view-switch detection performance (Sec. 2), as referenced in Sec. 3.2 in main
- Annotation filtering and model evaluation with higher inter-annotator agreement thresholds (Sec. 3), as noted in ‘Annotator agreement on best view’ in Sec. 4 in main
- View-selection results upon finetuning our view-switch detector jointly with narration-based pseudo-labels and our best view labels (Sec. 4), as mentioned in ‘View selection’ in Sec. 5 in main
- Analysis of the impact of the duration of our past frames on view-switch detection performance (Sec. 5), as noted in ‘Ablations’ in Sec. 5 in main
- Analysis of the impact of the duration of our past narrations on view-switch detection performance (Sec. 6), as referenced in ‘Ablations’ in Sec. 5 in main
- Analysis of the impact of the number of training samples on view selection performance (Sec. 7), as mentioned in ‘Ablations’ in Sec. 5 in main
- Scenario-level breakdown of view selection performance (Sec. 8), as noted in ‘Ablations’ in Sec. 5 in main
- Feature similarity baseline evaluation with CLIP [11]-style encoders (Sec. 9), as mentioned in ‘Baselines’ in Sec. 5 in main
- Dataset details (Sec. 10) in addition to the ones provided in Sec. 3.2, and ‘Training data’, ‘Evaluation data’ and ‘Data for view selection with limited labels’ in Sec. 4 in main
- Annotation details (Sec. 11) in addition to the ones provided in ‘Evaluation data’ and ‘Annotator agreement on best view’ in Sec. 4 in main
- Additional implementation details (Sec. 12) to supplement the ones mentioned in ‘Implementation’ in Sec. 5 in main

Model	Accuracy	AUC	AP
Ours w/o shot-level pseudo-labeling	51.5	51.9	52.3
Ours	59.4	63.8	60.5

Table 1. Analysis of the impact of our shot-level pseudo-labeling strategy on view-switch detection performance on the HowTo100M [9] dataset. All values are in %, and higher is better. Significance $p \leq 0.05$.

1. Supplementary video

The supplementary video, available at https://vision.cs.utexas.edu/projects/switch_a_view/, qualitatively illustrates our task, View Selection with Limited Labels, and our main idea towards tackling that task, Weakly-Supervised Learning from Unlabeled In-the-wild Videos. We also show successful predictions by our model for both view-switch detection and view selection. For view selection, we additionally provide multi-step selection examples, where our model selects the best view over multiple consecutive steps. Finally, we illustrate our model’s common failure modes (‘Qualitative examples’ in Sec. 5 in main) with qualitative examples.

2. Shot-level pseudo-labeling

In Table 1, we report the results for an additional ablation study, in which we analyze the impact of our shot-level pseudo-labeling strategy (Sec. 3.2 in main) on view-switch detection with the HowTo100M [9] dataset. Upon replacing our shot-level pseudo-labeling strategy with a clip-level pseudo-labeling strategy (Sec 3.2 in main), we observe a drastic drop in model performance. This demonstrates that our pseudo-labeler is able to mitigate noisy clip-level predictions, particularly at scene boundaries.

3. Inter-annotator agreement threshold

In main, we evaluated our models with an inter-annotator agreement thresholds of 78%, meaning at least 7 out of 9 agree for each annotation instance (‘Annotator agreement on best view’ in Sec. 4 in main). Here, we evaluate even higher

Model	View-switch detection on HT100M			View selection on Ego-Exo4D		
	78%	89%	100%	78%	89%	100%
Retrieval [17]- F	53.2	53.6	53.6	—	—	—
View-narration [17] Similarity	—	—	—	53.9	54.2	53.7
LangView [8]-bigData	—	—	—	54.5	54.9	54.1
Ours	60.5	60.8	60.7	56.0	56.2	55.3

Table 2. Model performance (AP) vs. inter-annotator agreement threshold. — indicates that the baseline is not applicable for the particular task. All values are in %, and higher is better. Significance $p \leq 0.05$.

agreement thresholds of 89%—at least 8 out of 9 annotators agree, and 90%—all annotators agree. For HT100M [9], the number of samples drops from 3,151 to 1,840 at 80%, and 1,345 at 90%. For Ego-Exo4D [4], the same goes down from 5,049 to 3,421 at 80%, and 1,887 at 90%, respectively. Table 2 reports the results. Even at these higher and more challenging inter-annotator agreement thresholds, our model outperforms the strongest baseline—Retrieval [17]- F for view-switch detection, and LangView [8]-bigData for view selection—on both tasks.

4. Finetuning jointly with narration-based pseudo-labels and best view labels, for view selection

Here, we provide details on how we finetune our view-switch detector jointly with narration-based pseudo-labels [8] and our best view labels (‘Data for view selection with limited labels’ in Sec. 4 in main), for doing view selection. Essentially, we modify our view selector training loss \mathcal{L}^S (Sec. 3.5 in main) as follows:

$$\mathcal{L}^S = \mathcal{L}_{\text{CE}}(\tilde{V}_{(t_w, t_w + \Delta)}, V_{(t_w, t_w + \Delta)}) + \alpha * \mathcal{L}^{N'}, \quad (1)$$

where $\mathcal{L}^{N'} = \mathcal{L}_{\text{CE}}(\tilde{V}_{(t_w, t_w + \Delta)}, \tilde{V}_{(t_w, t_w + \Delta)}^{N'})$ is the cross-entropy loss between the predicted views $\tilde{V}_{(t_w, t_w + \Delta)}$ (Sec. 3.4 in main) and narration-based pseudo-labels [8] $\tilde{V}_{(t_w, t_w + \Delta)}^{N'}$, generated using next narrations N' (Sec. 3.1 in main), and α is the weight on $\mathcal{L}^{N'}$, which we set α to 0.3 on the basis of validation. See Fig. 1b for quantitative results.

5. Duration of past frames

In Table 3, we report our view-switch detection performance numbers for different durations of past frames, denoted by T^F , using the HowTo100M [9] dataset. We notice that our model performance declines monotonically as we move from our choice of $T^F = 8$ seconds (‘Implementation’ in Sec. 5 in main) to both smaller and larger values. While very short visual contexts fail to capture long-range temporal patterns in human-preferred view sequences, very long visual contexts might contain spurious signals that affect model performance. $T^F = 8$ seconds balances this trade-off and leads to the best model performance, per this study.

Model	Accuracy	AUC	AP
$T^F = 2$	59.4	63.1	59.6
$T^F = 4$	<u>59.0</u>	<u>63.4</u>	<u>60.2</u>
$T^F = 8$ (Ours)	59.4	63.8	60.5
$T^F = 16$	55.4	59.1	57.0
$T^F = 32$	52.8	55.0	53.6

Table 3. Analysis of the impact of the duration of past frames, denoted with T^F , on view-switch detection performance on the HowTo100M [9] dataset. All values are in %, and higher is better. Significance $p \leq 0.05$.

Model	Accuracy	AUC	AP
$T^N = 2$	56.1	55.9	56.2
$T^N = 4$	52.4	53.9	53.4
$T^F = 8$	55.5	60.2	58.0
$T^N = 16$	<u>56.1</u>	<u>60.2</u>	<u>58.0</u>
$T^N = 32$ (Ours)	59.4	63.8	60.5

Table 4. Analysis of the impact of the duration of past narrations, denoted with T^N , on view-switch detection performance on the HowTo100M [9] dataset. All values are in %, and higher is better. Significance $p \leq 0.05$.

6. Duration of past narrations

In Table 4, we report our view-switch detection results for different durations of past narrations, denoted by T^N , with HowTo100M [9]. Upon reducing T^N to values lower than our choice of 32 seconds (‘Implementation’ in Sec. 5 in main), our model performance declines monotonically. This shows that a longer past narration context helps better learn correlations between the text in the narrations and desired view types.

7. Sample count for training view selector

In Fig. 1a, we study the effect of sample count on our view selection performance. Our model already improves performance with as few as 1000 samples. This plot also highlights the low-shot success of our model versus the best baseline, LangView [8]-bigData.

8. Scenario-level analysis of view-selection performance

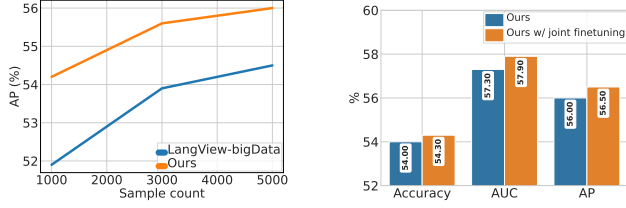


Figure 1. (a) Effect of sample count on our view selection (VS) performance; (b) Impact of joint finetuning with narration-based pseudo-labels [8] and best view labels on view selection (VS)

Model	HowTo100M [9]			Ego-Exo4D [4]		
	Accuracy	AUC	AP	Accuracy	AUC	AP
Retrieval [11]- F	53.0	53.0	52.7	52.1	52.1	53.4
Retrieval [11]- N	52.3	52.3	51.8	51.8	51.8	50.7
Retrieval [11]- N'	52.6	52.6	53.2	52.0	52.0	52.5
Ours	59.4	63.8	60.5	51.2	56.4	55.4

Table 5. View-switch detection results. Evaluation on Ego-Exo4D [4] is zero-shot. All values are in %, and higher is better. Significance $p \leq 0.05$.

Fig. 2 shows the breakdown of view selection performance per scenario, where only the scenarios with a minimum of 10 instances after filtering low-quality annotations (‘Data for view selection with limited labels’ in Sec. 4 in main) are shown. Compared to the best-performing baseline, LangView [8]-bigData, our model’s performance goes up both in absolute and relative terms, from the procedural scenarios like Cooking or Bike Repair, to physical scenarios like Rock climbing. This demonstrates that our model is better able to handle more scenarios with more physical activity, and consequently, more view changes, than the best baseline.

9. Feature similarity baselines with CLIP [11] encoders

In ‘View-switch detection’ and ‘View selection’ in Sec. 5 in main, we evaluated feature similarity baselines with In-

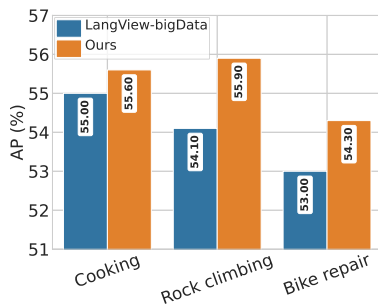


Figure 2. Per-scenario breakdown of our and the strongest baseline, LangView-bigData’s view selection performance, measured with AP (%).

Model	Accuracy	AUC	AP
Retrieval [11]- F	52.1	52.1	53.4
Retrieval [11]- N	51.8	51.8	50.7
Retrieval [11]- N'	52.0	52.0	52.5
View-narration [7] Similarity	52.5	52.2	53.4
Ours	54.0	57.3	56.0

Table 6. Results and ablation study for view selection with limited labels. All values are in %, and higher is better. Significance $p \leq 0.05$.

ternVideo2 [17] encoders. Here, we provide the parallel experiment with CLIP [11]-style encoders in Table 5 for view-switch detection, and Table 6 for view selection. Specifically, while for the retrieval baselines, we use the unmodified CLIP encoders, we use X-CLIP [7] encoders in the View-narration Similarity baseline for encoding multiple frames in the ego and exo views (Sec. 3.4 in main). With CLIP, the similarity baselines generally perform worse. This happens possibly because, unlike InternVideo2, CLIP features are not very fine-grained and/or do not capture temporal context in the case of retrieval baselines. Furthermore, our model outperforms all feature similarity baselines, even when implemented with CLIP, highlighting its advantages over the baselines across different encoder choices.

10. Additional dataset details

Here, we give further dataset details, in addition to the ones provided in ‘Training data’, ‘Evaluation data’ and ‘Data for view selection with limited labels’ in Sec. 4 in main.

Charades-Ego [12] datasets for training view classifier in our pseudo-labeler. As mentioned in Sec. 3.2 in main in main, we train the view classifier of our pseudo-labeler on the Charades-Ego [13] dataset. To do so, we create a dataset containing 5,551 train, 615 val, and 1,597 test videos, where all videos are randomly sampled and the splits are completely disjoint. Moreover, we train and test our model by sampling fixed-length clips, where the clip length is set to 2 seconds.

HowTo100M [9] datasets for view-switch detection. As noted in ‘Training data’ and ‘Evaluation data’ in Sec. 4 in main in main, we train our view-switch detector on HowTo100M (HT100M) [9] and also use it as a dataset for evaluating view-switch detection. To do so, we sample a maximum of 500 videos from each category in the second level of the HT100M video classification hierarchy. This results in a total of 4,391 hours of HT100M videos.

In addition to the details provided in ‘Evaluation data’ in Sec. 4 in main, for creating the HT100M test set for evaluating view-switch detection, we also include the clips

right after all view-switch boundaries, as identified by our pseudo-labeler, in the test set. This ensures that the test set is not totally dominated by the more frequently-occurring same-view instances, which can affect the estimation of our unbiased mean performance (‘Evaluation metrics’ in Sec. 5 in main). Finally, we provide the clip just before each clip being labeled, to the annotators in order to identify instances where the *ground-truth* view stays the same (same-view) and where it switches (view-switch). This allows us to separately evaluate these two alternate but important scenarios, and report *unbiased* mean performance (‘Evaluation metrics’ in Sec. 5 in main).

Ego-Exo4D [4] datasets. We create our datasets for Ego-Exo4D [4] (‘Evaluation data’ and ‘Data for view selection with limited labels’ in Sec. 4 in main) for evaluating view-switch detection, and training and evaluating view selection, by sampling clips from each video at a regular interval of 1 second.

11. Additional annotation details

Here, we provide further annotation details, in addition to what are provided in (‘Evaluation data’ and ‘Annotator agreement on best view’ in Sec. 4 in main). We start with the details that are common for both both HT100M [9] and Ego-Exo4D [4].

We use Amazon Mechanical Turk (MTurk) to collect annotations for both datasets. Before assigning an MTurk worker our job, we ensure that their prior annotation approval rate is more than 98%. We also require them to take short qualifiers (‘Annotator agreement on best view’ in Sec. 4 in main), each of which contains 10 annotation instances. We design these qualifiers such that they are very similar to our main jobs. Furthermore, we handpick the annotation instances in the qualifiers such we that know the ground-truth for each of them. This lets us easily compare an annotator’s choices against the ground-truths in the qualifiers, and consequently, gauge the their reliability. We only accept annotators who pass these qualifiers with a success rate of at least 90%.

Next, we provide dataset-specific annotation details.

HT100M. In Fig. 3, we show our annotation interface for HT100M (‘Evaluation data’ and ‘Annotator agreement on best view’ in Sec. 4 in main). In short, we provide a set of detailed instructions, which lists the different characteristics of both ego and exo clips¹, and give examples for each characteristic. Additionally, we provide a more concise summary of the lists of per-view attributes on each annotation page to give a quick recap of the annotation task, to the workers.

¹We refer to ego and exo views as “closeup” and “wide” shots, respectively, in order to easily explain the annotation process to the workers.

Finally, we provide video examples of both ego and exo clips, to further guide the annotation process.

Ego-Exo4D. In Fig. 4, we show our annotation interface for Ego-Exo4D (‘Evaluation data’ and ‘Annotator agreement on best view’ in Sec. 4 in main). In summary, we provide a set of detailed instructions on each annotation page, which describes the kind of information captured by the two views (ego and exo) and the role of the associated atomic description, and also specify that we expect the annotator to pick a view that best shows the activity mentioned in the atomic description and hence, useful for instructional purposes. To further assist with the annotation process, we provide examples showing pairs of clips from both ego and exo views, their associated atomic descriptions, and how an annotator should reason about which view is better for viewing the activity, in the context of its corresponding atomic description.

12. Additional implementation details

Here, we provide additional implementation details.

View assignment to past frames and narrations for Ego-Exo4D [4]. In Sec. 3.2 in main, we provide details about how we assign our view pseudo-labels to past frames and past narrations, for using our model (view-switch detector or view selector) with HowTo100M [9]. Here, we describe our process for assigning views to the past frames and narrations for Ego-Exo4D [4].

Note that all frames or narrations in an Ego-Exo4D video might not be assigned a ground-truth best view during our annotation process, because the annotations for some couplets of pairs of clips from the two views, and their associated atomic descriptions (‘Evaluation data’ in Sec. 4 in main), might get discarded due to our annotation quality control measures (‘Annotator agreement on best view’ in Sec. 4 in main, and above). To tackle this, we set the best view for each past frame and past narration to the best view ground-truth for the nearest couplet of clip pair and its associated atomic description, for which the ground-truth has not been discarded.

12.1. View pseudo-labeler

Scene detector. As mentioned in Sec. 3.2 in main, we use PySceneDetect [1], an off-the-shelf scene detector, for detecting scene boundaries in HowTo100M [9] videos. Specifically, we use the image-content-based detector. Moreover, we set the weights for pixel colors to 1.0 and the same for object edges to 0.0, and the minimum shot length to 2 seconds.

View classifier. For our view classifier (Sec. 3.2 in main), we use the slow branch of a SlowFast [2] model that has

Instructions

Please SCROLL DOWN and CLICK ON "More instructions" for VIDEO EXAMPLES before entering your responses!!

1. Watch a few videos of a person doing an activity and for each video, check if a **majority** of its frames is from a **closeup** or a **wide shot**.

By closeup, we refer to shots that

- have the camera placed close to the entities involved in the activity (e.g., hands or legs, objects or tools being used, etc.)
- provide a zoomed-in view of the same
- might look like they have been captured from a first-person perspective, i.e., the point of view of the person doing the activity
- might have high camera motion and look like the camera is hand-held or head-worn by somebody

By wide shots, we refer to shots that

- have the camera placed at a larger distance
- might show entities not involved in the activity (e.g., talking head or torso of the person, background objects, etc.)
- might look like they have been captured from a third-person perspective, i.e., the point of view of somebody watching the activity

Check examples in "More instructions" for examples of these two shot types.

2. Additionally, check if you are **sure** or **unsure** about the shot types you chose in the first response.

Please first click the "Instructions" button on the top-left and read the instructions if **this is your first time**.

Please watch the short videos and answer the questions. If videos are not playing, please use Chrome or Mozilla.

TASK: choose if a **majority** of the frames in the videos from a **closeup** or a **wide shot**


To recap, by closeup, we refer to shots that

- have the camera placed close to the entities involved in the activity
- provide a zoomed-in view of the same
- might look like they have been captured from the point of view of the person doing the activity
- might have high camera motion and look like the camera is hand-held or head-worn by somebody

By wide shots, we refer to shots that

- have the camera placed at a larger distance
- might show entities not involved in the activity
- might look like they have been captured from the point of view of somebody watching the activity

Video 1



Q1*. Is a **majority** of the frames in the videos from a **closeup** or a **wide shot**?

☐ Closeup shot ☐ Wide shot

Figure 3. Sample interface for collecting HT100M [9] annotations (‘Evaluation data’ and ‘Annotator agreement on best view’ in Sec. 4 in main). Additionally, we also provide video examples for both ego (closeup shot) and exo (wide shot) clips, to help the annotators.

a ResNet3D [15]-50 architecture and is pretrained on the Kintetics-400 [5] dataset, as the visual encoder. This encoder takes 8 uniformly sampled frames from each 2-second clip (c.f. Sec. 11), embeds them into a visual feature, and passes the visual feature to a linear classification head, which is implemented as a single linear layer. We initialize the parameters for all model components that are trained from scratch, using a parameter initialization strategy for masked auto-encoders for videos [3]. We train the model until convergence by using an AdamW [6] optimizer, a batch size of 32, and initial learning rates of 10^{-5} and 10^{-4} for the visual encoder, and the classification head, respectively.

12.2. View-switch detector

Here, we provide more implementation details of view-switch detector, in addition to those provided in ‘Implementation’ in Sec. 5 in main. Our detector’s DINOv2 [10] frame encoder has 12 layers and takes in frames sampled at 4 fps, and produces a 768-dimensional feature for each frame. Our detector’s Llama 2 [14] text encoder begins by producing a token sequence for each input narration by tokenizing its text, and padding the tokenizer output to match the length of the longest token sequence in a batch, or truncating it to reduce its length to 512, depending on which length is shorter. Moreover, for the past narrations, it ensures that their total token count does not cross 1024, by truncating

wherever necessary. It then encodes each token from the past narration sequence, or the next narration, into a 4096-dimensional features. Next, it projects each such feature into a 768-dimensional feature using a linear layer. We implement our modules for encoding views for both frames and past narrations, and their relative temporal positions, as learnable embeddings that produce 768-dimensional features. Specifically, for encoding views, we use a learnable feature dictionary with 0 (ego) and 1 (exo) as keys. For encoding relative temporal positions, we first discretize the durations in seconds, by using bins of size 0.1 second, and then encode them with learnable feature dictionaries, in which the number of keys is set to the maximum number of bins. To aggregate all the above-mentioned 768-dimensional features, we use a 8-layer and 8-head transformer encoder [16] that adds a positional embedding [16] to each feature and performs self-attention on them. Finally, the 2-layer MLP for our view classification is a stack of two hidden linear layers with the output feature size of 256 and 64, respectively, and a final linear layer that estimates the next view. We initialize the parameters for all model components that are trained from scratch, using a parameter initialization strategy for masked auto-encoders for videos [3]. We freeze the pre-trained components of our view-switch detector and train the rest of the model until convergence with the AdamW [6]

Please first click the **"Instructions"** button on the top-left and read the instructions if **this is your first time**.

Please watch the short videos and answer the questions. If videos are not playing, please use Chrome or Mozilla.

TASK: Watch pairs of videos and choose which video--**Left** or **Right**--**more accurately** shows the **activity** mentioned in the **text** written below each pair, and is **also more useful** for **teaching** somebody **how to** do the activity ?

For each pair


- a person performs an activity, which is shown from two viewpoints
- the video on the **left** is captured from the **perspective of the person doing the activity**, and provides a **first-person view** of the activity
- the video on the **right** is captured with a camera placed next to the site of the activity, and provides a **third-person view** of the activity
- the **text** below each pair provides a **fine-grained description** of the activity.

By 'choose which video more accurately shows the activity in the text', we ask you to choose the video that more clearly shows the body parts and other objects involved in the activity, and these entities interact with each other, as described in the text, and is **also more useful** for **teaching** somebody **how to** do the activity.

Note that

- the letter '**C**', when used as the subject in the text, denotes the person doing the activity
- other uppercase letters, like '**X**', '**O**', etc., when used in the text, denote other persons involved in the activity

Pair 1



Text: C places the garlic in the bowl on the countertop, using his right hand

Q1*. Which video--**Left** or **Right**--do you think more accurately shows the activity mentioned in the text?
By accurate, we mean that the video you choose should more clearly show the human body parts and objects, and their interactions as mentioned in the text. (refer to instructions for examples).

☐ Left ☐ Right

Figure 4. Sample interface for collecting Ego-Exo4D [4] annotation (‘Evaluation data’ and ‘Annotator agreement on best view’ in Sec. 4 in main) . Additionally, we also provide examples showing pairs of clips from both ego and exo views, their associated atomic descriptions, and how to reason about which view is better for viewing an activity, to help the annotators.

optimizer, a batch size of 48, and a learning rate of 10^{-6} .

12.3. View selector

Our view selector has the exact same architecture as our view-switch detector. For training it, we first initialize its parameters with those of our pretrained view-switch detector. We then freeze the frame encoder and the text encoder, and finetune the rest of the model until convergence with the exact same optimizer and hyperparameters as the ones used in our view-switch detector training.

References

- [1] Brandon Castellano. Pyscenedetect. <https://github.com/Breakthrough/PySceneDetect>. 4
- [2] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 4
- [3] Christoph Feichtenhofer, Yanghao Li, Kaiming He, et al. Masked autoencoders as spatiotemporal learners. *Advances in neural information processing systems*, 35:35946–35958, 2022. 5
- [4] 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](#), [4](#), [6](#)
- [5] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. [5](#)
- [6] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. [5](#)
- [7] 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. [3](#)
- [8] 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](#), [3](#)
- [9] 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. [1](#), [2](#), [3](#), [4](#), [5](#)
- [10] 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. [5](#)
- [11] 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. [1](#), [3](#)
- [12] Gunnar A Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Actor and observer: Joint modeling of first and third-person videos. In *proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7396–7404, 2018. [3](#)
- [13] 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. [3](#)
- [14] 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. [5](#)
- [15] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. [5](#)
- [16] 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. [5](#)
- [17] 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. [2](#), [3](#)