Which Viewpoint Shows it Best? Language for Weakly Supervising View Selection in Multi-view Instructional Videos Supplementary Material

Sagnik Majumder[1,2] Tushar Nagarajan[2] Ziad Al-Halah[3] Reina Pradhan[1] Kristen Grauman[1,2]
[1]UT Austin [2]FAIR, Meta [3]University of Utah

In this supplementary material we provide additional details about:

- Video (with audio) for qualitative illustration of our task and qualitative assessment of our view predictions (Sec. 1), as referenced in 'Qualitative examples' in Sec. 4.2 in main in main
- Additional ablations of our model components (Sec. 2), as mentioned 'Ablations' in Sec. 4.2 in main
- Analysis of the view-specificity of our model's learned visual features (Sec. 3), as noted in 'Ablations' in Sec. 4.2 in main
- Analysis of the impact of rank our selector's sampled view on view selection performance (Sec. 4), as mentioned in 'Ablations' in Sec. 4.2 in main
- Examples of our view selector's attention heatmaps (Sec. 5), as noted in 'Ablations' in Sec. 4.2 in main
- Analysis of our pseudo-labeler (Sec. 6), as referenced in 'Ablations' in Sec. 4.2 in main
- View selection results on Ego-Exo4D [9] with a single exo camera (Sec. 7), as mentioned in 'Automatic evaluation' in Sec. 4.2 in main
- 3-fold evaluation of our view selector on Ego-Exo4D [9], as noted in 'Automatic evaluation' in Sec. 4.2 in main
- Analysis of the relation between our model performance and the distribution of different concepts in the grountturth train narrations (Sec. 10)
- Our pseudo-labeling cost (Sec. 9)
- Dataset details (Sec. 11) in addition to what is provided in 'Dataset' in Sec. 4.1 in main
- Implementation details (Sec. 12), as noted in 'Implementation' in Sec. 4.1 in main

1. Supplementary video

The supplementary video, available at https://vision.cs.utexas.edu/projects/which-view-shows-it-best, qualitatively depicts our task of view-selection in multi-view instructional videos. Moreover, we qualitatively illustrate our key idea, Language for Weakly Supervising View Selection, show our model's view selection quality at the level of both individual clips and long videos (comprising multiple clips), and compare our predictions with those of two best-performing baselines. Some long videos also have the audio commentary of the participant. Please use headphones to hear the audio correctly.

2. Additional ablations

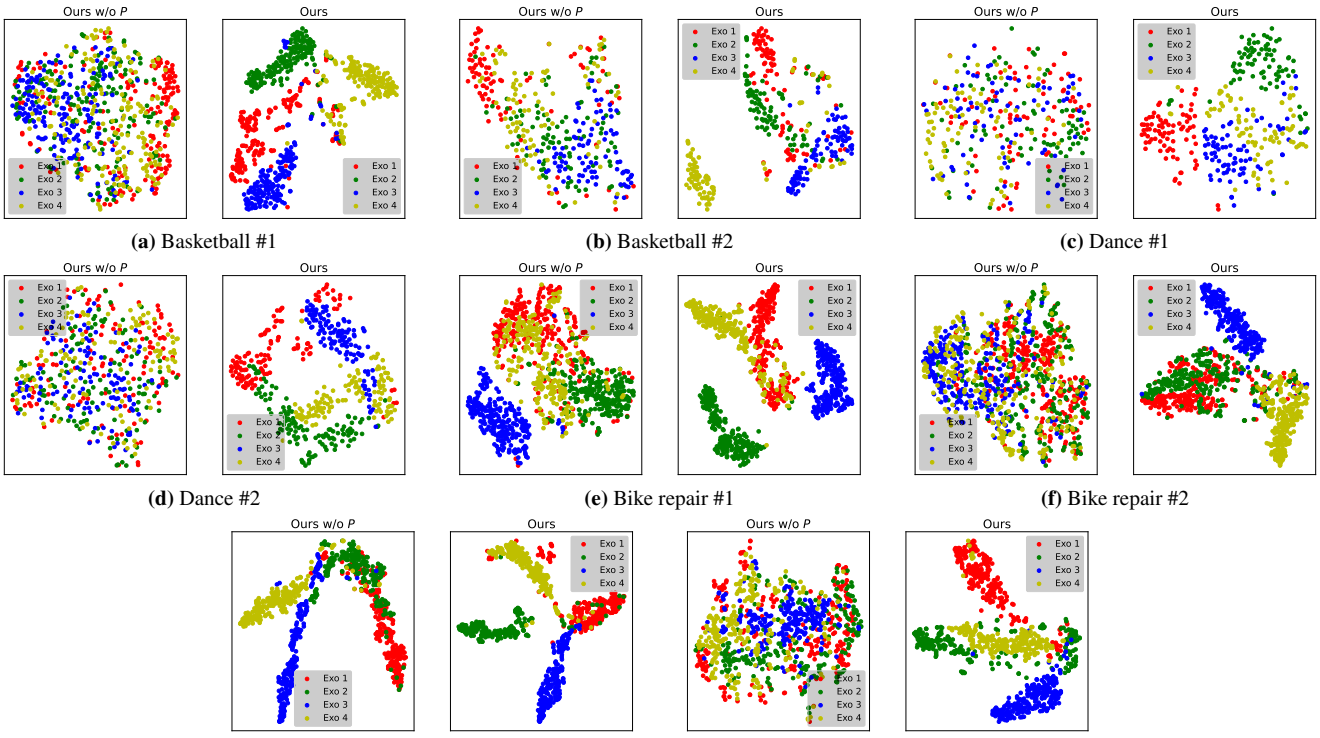
In 'Ablations' in Sec. 4.2 of main, we ablate different model components to understand their contribution to our view selection performance. Here, we provide additional ablations to further analyze our model. Table 1 shows the results. Upon keeping the off-the-shelf captioners [15, 27] frozen when generating our best view pseudo-labels using our pseudo-labeler L (Sec. 3.2 in main), the performance declines drastically, indicating that the generic captions generated by frozen off-the-shelf captioners are not at all suitable for activity understanding in instructional videos. Upon predicting the exact displacement of one camera center relative to another, instead of the rough direction between them, when predicting the inter-view relative poses using our relative camera pose predictor P (Sec. 3.3 in main), we against observe a significant drop in view selection performance. This happens possibly because predicting the exact difference in locations between two camera centers can be intractable in our setting, due to the unknown scale of objects and background.

3. View dependence of visual features

Fig. 1 shows the t-SNE visualizations of the visual features corresponding to the exo views of videos from different scenarios–basketball, dance, bike repair and cooking. The scenarios have varying levels of motion of the camera wearer's body and relevant objects–whereas basketball and dance involve moving large and fast movements of the full body and salient objects, bike repair and cooking primarily just involve hands and need less body and object motion. Our learned visual features for the exo cameras when grouped on the basis of the camera ID, produce tighter clusters across samples from different scenarios, compared

Model	*Captioning* CIDEr [25]	METEOR [2]	*Actions and objects* V-IoU	N-IoU	NC-IoU
Ours w/o captioner finetuning in our pseudo-labeler L	0.4	12.2	1.4	6.5	4.8
Ours w/o direction prediction between camera centers in our relative camera pose predictor P	12.9	48.1	32.5	36.8	31.6
Ours	**13.5**	**48.4**	**33.7**	**39.2**	**32.9**

Table 1. Ablation results on the large-scale Ego-Exo4D [9] dataset, in addition to what is provided in 'Ablations' in Sec. 4.2 in main. For the ablation that does not predict the direction between camera centers during relative pose prediction, we predict the exact differences in locations between camera centers instead. Significance, $p \leq 0.05$.



(a) Basketball #1 **(b)** Basketball #2 **(c)** Dance #1

(d) Dance #2 **(e)** Bike repair #1 **(f)** Bike repair #2

(g) Cooking #1 **(h)** Cooking #2

Figure 1. t-SNE [23] plots of exo visual features of sample Ego-Exo4D [9] videos from basketball, bike repair, dance and cooking scenarios. Our model, when trained with the relative camera pose predictor, produces visual features that form neater clusters when grouped on the basis of different exo views, highlighting their improved view sensitivity.

to the model variant trained without our relative camera pose estimation loss ('View selector training' in Sec. 3.4 in main). This demonstrates that our model's superior ability to learn view-dependent features cuts across different types of activity and different levels of body and object motion, which consequently leads to a stronger view selection performance.

4. Sampled view rank

Table 2 shows the impact of the rank of our sampled view on view selection performance. We observe that the lower the rank of our sampled view is, within our model's learned view order, the worse our view selection performance is. This shows that our model's learned ranking of views is highly correlated with the view quality, which indicates that our model successfully builds an implicit understanding of which views are more informative.

Model	*Captioning* CIDEr [25]	METEOR [2]	*Actions and objects* V-IoU	N-IoU	NC-IoU
Worst	10.9	45.1	29.2	35.8	30.7
Second best	11.9	46.4	30.9	35.8	30.6
Best (Ours)	**13.5**	**48.4**	**33.7**	**39.2**	**32.9**

Table 2. Effect of the rank of our sampled view on the view selection performance on Ego-Exo4D [9]. Significance, $p \leq 0.05$.

5. Attention heatmaps of our view selector

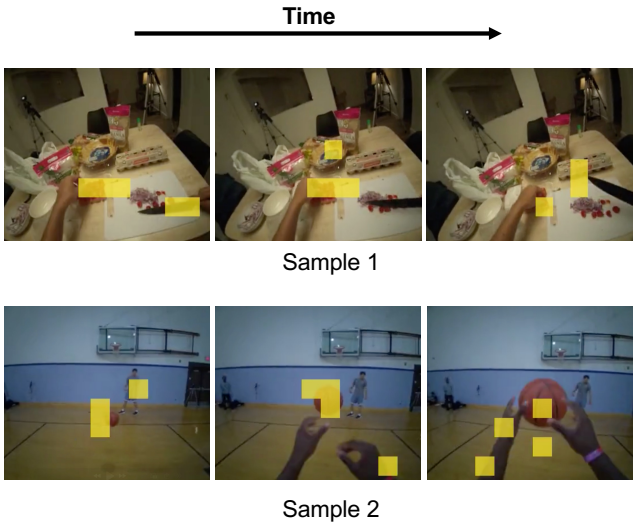


Figure 2. Our model's attention heatmaps on two best view clips from Ego-Exo4D [9]. Yellow patches indicate highest attention.

Ego-Exo4D [9]					*LEMMA [12]*	
Ego	Exo 1	Exo 2	Exo 3	Exo 4	Ego	Exo
20.4	19.8	20.3	19.6	19.9	63.6	36.4

Table 3. Probability distribution in % of our best view pseudo-labels.

Model	CIDEr	METEOR	V-IoU	N-IoU	NC-IoU
Ours w/ 2 captioners	13.3	**48.4**	**34.2**	38.1	32.5
Ours (w/ 3 captioners)	**13.5**	**48.4**	33.7	**39.2**	**32.9**

Table 4. Impact of captioner count on view selection performance, evaluated with Ego-Exo4D [9]. Significance, $p \leq 0.05$. See row 3 of Table 3, and Sec. 4.2, in main, for results with 1 captioner.

In Fig. 2, we provide examples of our model's attention heatmaps on Ego-Exo4D [9]. Our model tends to focus on the salient objects for an activity, even if they are dynamic, indicating its strong activity understanding ability.

6. Analysis of our best view pseudo-labeler

Here, we analyze different aspects of our pseudo-labeler L (Sec. 3.2 in main).

In Table 3, we report the distribution of our selected views for both Ego-Exo4D [9] and LEMMA [12] datasets. For Ego-Exo4D, our model produces a more or less uniform distribution over all views, indicating that depending on the activity and its level of body and object motion, our model can prefer the ego view or one of the exo views with almost equal likelihood. However, for LEMMA, our model tends to prefer the ego view much more than the exo view, re-emphasizing the prevalence of household activities that largely require the ego view for capturing their informative aspects ('Dataset' in Sec. 4.1 in main).

In addition to the ones provided in Fig. 2b in main, we show more pseudo-labeler outputs, comprising view ranks and predicted narrations, alongside the ground-truth narrations, in Fig. 3. In Fig. 4, we provide more such examples without narrations. We see very similar patterns in these additional samples—the better our pseudo-labeler considers a view to be, the more accurate the narration predicted from the view, is, in terms of capturing important activity details.

In Table 4, we compare our view selection performance on Ego-Exo4D [9], when using 3 vs. 2 captioners—see row 3 of Table 3, and Sec. 4.2, in main for results with 1 captioner, in our pseudo-labeler (Sec. 3.3 in main). Our view selection performance general improves with the increase in the captioner count in our pseudo-labeler, possibly because having more captioners vote on the best view reduces captioning noise and improves pseudo-label quality.

7. Ego-Exo4D with single exo camera

Here, we evaluate our view selector on the single exo camera variant of Ego-Exo4D [9] in order to emulate more typical instructional settings [12, 20] that consist of a single exo camera, but also retain the challenges in the Ego-Exo4D data arising from the diversity in scenarios, varying degrees of body and object motion, etc. Table 5 shows the results, where all metrics are first computed separately for each possible ego-exo view pair and then averaged over all pairs. Our model significantly outperforms all baselines across metrics, showing that it is robust to different camera setups even on challenging datasets with diverse activity scenarios and varying levels of motion of the objects and body parts involved in the activity.

8. 3-fold evaluation on Ego-Exo4D

In Table 6, we report the results from 3-fold evaluation with Ego-Exo4D [9]. Our model significantly outperforms Body-Area, the best baseline. This shows that our model's improvement over the baselines sustains across multiple test datasets.

9. Pseudo-labeling cost

We use 8 NVIDIA V100 GPUs for training and performing inference with the captioners in our pseudo-labeler (Sec. 3.2 in main). When pseudo-labeling Ego-Exo4D [9], it takes ~2.5 days with VideoLlama captioners, and 3 hours with VideoChat2. For LEMMA [12], the same takes 1 hour per captioner. Importantly, this is a one-time cost since we pseudo-label only once per dataset, and we do not use any captioner when training or evaluating our view selector.

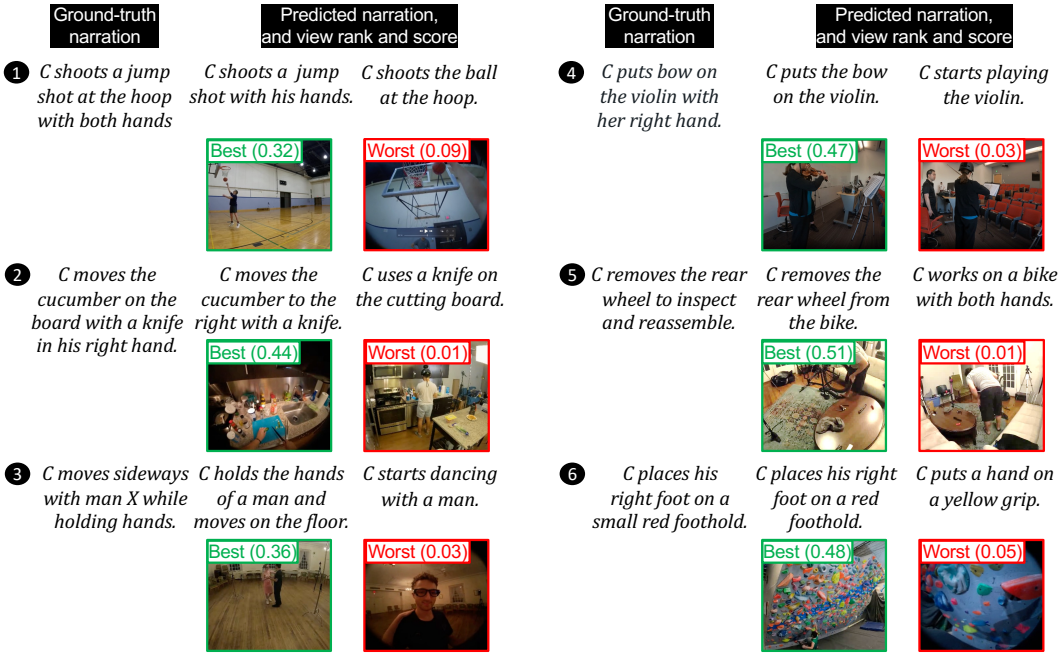


Figure 3. Examples of predicted narrations, and the ranks and scores of the views, per our pseudo-labeler L, shown alongside ground-truth narrations, in addition to what is provided in Fig. 2 in main.

Figure 4. Additional examples of best and worst views, and their scores, per our pseudo-labeler L.

10. Model performance vs. distribution of concepts in ground-truth train narrations

Fig. 5 plots our *test* gains over Body-area [13], the strongest baseline, versus the frequency (most to least) of occurrence of different concepts in the ground-truth *train* narrations. The lack of a strong correlation demonstrates that our view selection is not biased by the dominant concepts in the training narrations.

11. Dataset details

Here, we provide additional dataset details. For both Ego-Exo4D [9] and LEMMA [12], we uniformly sample 8 frames from each clip and resize each frame to 224×224. Further, we normalize each pixel in a frame by first dividing it by 255 so that its value lies in $[0, 1]$, then subtracting the pixel mean and finally dividing by the pixel standard deviation, where the pixel mean and standard deviation are channel-specific. We set the mean and standard deviation to $[0.48145466, 0.4578275, 0.40821073]$ and $[0.26862954, 0.26130258, 0.27577711]$, respectively, for our view selector and Video-Llama [27] captioners, and

Model	*Captioning*		*Actions and objects*		
	CIDEr [25]	METEOR [2]	V-IoU	N-IoU	NC-IoU
Ego	10.2	45.2	30.2	34.1	29.1
Random	9.8	44.5	29.0	34.9	28.5
Random-exo	9.6	43.8	28.0	34.2	27.4
Hand-object [5]	11.5	46.8	32.2	36.8	30.5
Body-area [13]	10.3	45.4	30.2	34.4	28.4
Joint-count [13]	9.9	44.6	28.6	34.1	28.1
Pixel-objectness [4, 26]	11.2	46.1	30.9	35.9	29.4
Longest-caption	0.0	0.0	0.0	0.0	0.0
Ours	**12.7**	**47.1**	**32.7**	**37.3**	**30.9**

Table 5. View selection with Ego-Exo4D, when the candidate viewpoints comprise the ego view and one exo view. All metrics, expressed in % are averaged over all possible ego-exo view pairs. Significance, $p \leq 0.05$.

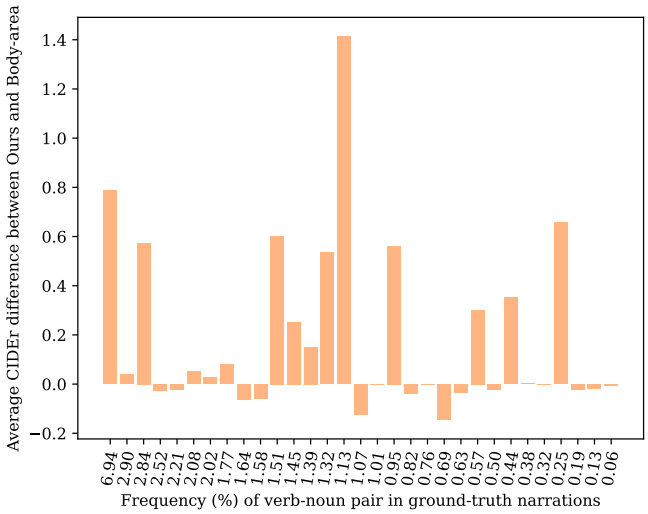


Figure 5. *Test* CIDEr difference between our model and the Body-area [13] baseline vs. verb-noun pair frequency in *train* narrations, sorted in decreasing order

Model	CIDEr	METEOR	V-IoU	N-IoU	NC-IoU
Body-area	10.5	46.6	30.0	35.2	30.4
Ours	**11.4**	**46.9**	**31.2**	**37.0**	**31.9**

Table 6. Average view selection results over three disjoint test splits from Ego-Exo4D [9]. Significance, $p \leq 0.05$.

$[0.485, 0.456, 0.406]$ and $[0.229, 0.224, 0.225]$, respectively, for our VideoChat2 [15] captioner, where the channels follow the RGB order.

We split the Ego-Exo4D videos into sequences of clips, each coupled with a narration, by adopting the "contextual variable length clip pairing strategy" strategy [16, 22], which generates temporal windows for extracting clip-narration pairs. To split the LEMMA videos into clips, we group contiguous frames using their verb and noun annotations ('Dataset' in Sec. 4.1 in main).

For Ego-Exo4D, we preprocess each narration by denoting each activity participant mentioned in the narration using 'Xi', where i is the participant's position in the sequence in which the participants appear in the time-sorted narrations for each full video (a take in Ego-Exo4D). The value of i starts from 0. We produce narrations for LEMMA by appending the verb and object annotations, where each narration has the following structure: 'verb1: object1_1, object1_2, ...; verb2: object2_1, object2_2, ...; ...' .

12. Implementation details

Here, we provide additional implementation details for different components of our framework, and our Pixel-objectness [4, 26] baseline.

12.1. Captioner

For our VideoLlama [27] and VideoChat2 [15] captioners, we use a model with the same architecture as proposed in the original paper and initialize the parameters from the check-

points released by the authors. We freeze the ViT [8] encoder and LLM (without LoRA [10], wherever it is used) in all captioners, and train all other modules with an AdamW [19] optimizer for a maximum of 1.6 million iterations. We use a cosine annealing learning rate schedule [18] with a linear warmup over 5000 iterations, where we set the starting learning rate to 10^{-6}, the peak learning rate to 3×10^{-5}, and the minimum learning rate during cosine annealing to 1×10^{-5}. We set the total batch size to 8, and the (β_1, β_2) and weight decay in AdamW to $(0.9, 0.999)$ and 5×10^{-2}, respectively. Furthermore, for VideoChat2, we turn off flash attention [6, 7]. Finally, we set the LLM prompt to 'What is the person wearing smart glasses doing in the video?' for Ego-Exo4D [9] and 'What is the person wearing a head-mounted camera in the video doing?' for LEMMA [12].

12.2. View selector

We use the EgoVLPv2 [21] vision encoder, pretrained on Ego-Exo4D [9], to obtain visual features f in our view selector S (Sec. 3.3 in main). The EgoVLPv2 encoder is a 12-layer TimeSformer [3] model, where we set the prediction head ($head$), prediction logits (pre_logits) and fully-connected layer (fc) to identity functions from PyTorch. We attach a shared convolution layer to the encoder for producing shared features for both view classification in W (Sec. 3.3 in main) and pose prediction in P (Sec. 3.3 in main). The shared convolution has a kernel size, padding and stride of 1, 768 input channels and 192 output channels. The output of the shared convolution goes into a view selection head and a pose prediction head.

The view selection head begins with the following layers: 1) a Batch Normalization [11] layer with 192 input channels, 2) a ReLU [1] activation, 3) a convolution layer with a kernel size of 4, stride of 2, padding of 1, and 192 and 96 input and output channels, respectively, 4) a Batch Normalization layer with 96 input channels, 5) a ReLU activation, and 6) a convolution layer with a kernel size of 4, stride of 2, padding of 0, and 96 and 24 input and output channels, respectively. We feed the output of the last convolution from above to a a transformer [24] encoder, which comprises 2 layers with 8 heads and 768 channels. Each layer uses a dropout of 0.1 and uses sinusoidal positional encodings [24]. We then feed the output of the transformer encoder to a 2-layer MLP that comprises 1) a linear layer with 768 input channels and 128 output channels, 2) a Batch Normalization layer with 128 input channels, 3) a ReLU activation, 4) a dropout layer with the dropout probability set to 0.1, and 5) a linear layer with 128 input channels and the output channel count set to the number of views.

The pose prediction head comprises a convolution-only and linear-layer-only component. The convolution-only component comprises 1) a Batch Normalization [11] layer with $192 \times 2 = 384$ input channels, 2) a ReLU [1] activation, 3) a dropout layer with the dropout probability set to 0.1, and 4) a convolution layer with a kernel size of 4, stride of 2, padding of 1, and 384 and 48 input and output channels, respectively. The linear-layer-only component is comprised of 1) a Batch Normalization layer with 2352 input channels, 2) a ReLU activation, 3) a dropout layer with the dropout probability set to 0.1, 3) a linear layer with 2352 input dimensions and 53 output dimensions. We feed the outputs of the convolution-only component to the linear-layer-only component.

We employ $resize$ and $reshape$ operations from PyTorch wherever necessary.

We train our view selector using AdamW [19] with a learning rate of 10^{-5} for the EgoVLPv2 [21] vision encoder and 10^{-4} for the rest of the model. We set the total batch size to 24, and the (β_1, β_2) and weight decay in AdamW to $(0.9, 0.999)$ and 10^{-5}, respectively.

For all our model components, we stop training once the validation loss starts increasing.

12.3. Baseline: Snap angles [4, 26]

This baseline ('Baselines' in Sec. 4.1 in main) is an upgrade to the most relevant existing methods [4, 26] in the literature. It predicts the view with the highest count of pixels belonging to foreground [4, 26] and salient [4] objects but not lying near the frame boundaries [26], as the best view. To do so, we treat the set of all objects mentioned in the training narrations as foreground and salient, and query a model composed of GroundingDino [17] and Segment Anything (SAM) [14] with this set to detect its constituent pixels. Specifically, we first feed GroundingDino with the foreground-and-salient object set to compute the corresponding bounding boxes. Next, we feed these bounding boxes to SAM to mark all pixels of relevance. Finally, for each view, we compute a score that is a weighted sum of its average foreground-and-salient pixel count across all frames and a penalty term that lowers the count by the inverse of the view's frame count, for every pixel found within a certain distance from the frame boundaries. We set the weights on the foreground-and-salient pixel count to 1.0, and the penalty term to 0.1 and 0.02 for Ego-Exo4D [9] and LEMMA [12], respectively, through validation, and the distance for using a foreground-and-salient pixel in computing the penalty term, to 6.25% [26] of the frame size.

References

[1] Abien Fred Agarap. Deep learning using rectified linear units (relu), 2018. cite arxiv:1803.08375Comment: 7 pages, 11 figures, 9 tables. 6

[2] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine*

Translation and/or Summarization, pages 65–72, Ann Arbor, Michigan, 2005. Association for Computational Linguistics. 2, 5

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, page 4, 2021. 6

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

[5] Tianyi Cheng, Dandan Shan, Ayda Sultan Hassen, Richard Ely Locke Higgins, and David Fouhey. Towards a richer 2d understanding of hands at scale. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 5

[6] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023. 6

[7] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022. 6

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 6

[9] 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. 1, 2, 3, 4, 5, 6

[10] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 6

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, page 448–456. JMLR.org, 2015. 6

[12] Baoxiong Jia, Yixin Chen, Siyuan Huang, Yixin Zhu, and Song-chun Zhu. Lemma: A multi-view dataset for le arning m ulti-agent m ulti-task a ctivities. In *European Conference on Computer Vision*, pages 767–786. Springer, 2020. 3, 4, 6

[13] Tao Jiang, Peng Lu, Li Zhang, Ning Ma, Rui Han, Chengqi Lyu, Yining Li, and Kai Chen. Rtmpose: Real-time multi-person pose estimation based on mmpose. *ArXiv*, abs/2303.07399, 2023. 4, 5

[14] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 6

[15] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. *arXiv preprint arXiv:2311.17005*, 2023. 1, 5

[16] Kevin Qinghong Lin, Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Z Xu, Difei Gao, Rong-Cheng Tu, Wenzhe Zhao, Weijie Kong, et al. Egocentric video-language pretraining. *Advances in Neural Information Processing Systems*, 35:7575–7586, 2022. 5

[17] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 6

[18] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6

[19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[20] 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. 3

[21] Shraman Pramanick, Yale Song, Sayan Nag, Kevin Qinghong Lin, Hardik Shah, Mike Zheng Shou, Rama Chellappa, and Pengchuan Zhang. Egovlpv2: Egocentric video-language pre-training with fusion in the backbone. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5285–5297, 2023. 6

[22] Santhosh Kumar Ramakrishnan, Ziad Al-Halah, and Kristen Grauman. Naq: Leveraging narrations as queries to supervise episodic memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6694–6703, 2023. 5

[23] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9 (86):2579–2605, 2008. 2

[24] 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. 6

[25] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575, 2014. 2, 5

[26] Bo Xiong and Kristen Grauman. Snap angle prediction for 360° panoramas. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 5, 6

[27] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 1, 4, 5