`EgoExoLearn`: A Dataset for Bridging Asynchronous Ego- and Exo-centric View of Procedural Activities in Real World

Supplementary Material

This supplementary material shows details about our benchmark including formal definition, implementation, and additional experiment results. Also, we show additional details about the collection and annotation of the dataset.

S1. Additional Benchmark Details

S1.1. Cross-view association

S1.1.1 Detailed task definition

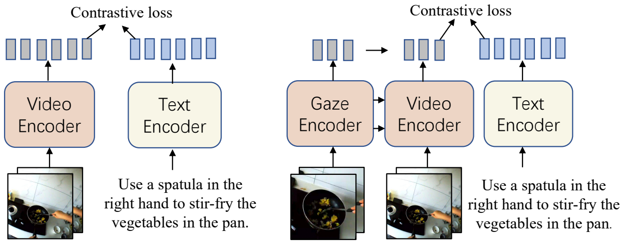
The training set consists of separate egocentric videos V^{ego} with associated narration T^{ego} and exocentric videos and narrations (V^{exo},T^{exo}). For each egocentric video, a sequence g with corresponding gaze is provided. Note that, we do not provide explicit pair information in the training set.

In the validation/test set, we introduce two evaluation settings, *i.e.,* Ego2Exo and Exo2Ego. We describe the formulation of Ego2Exo as follows. Each sample consists of an egocentric query video V^{ego} and K exocentric candidate videos $\{V_1^{\text{exo}}, ..., V_K^{\text{exo}}\}$, where only one candidate exocentric video corresponds to the query egocentric video, *i.e.,* the same action is being performed. In the Exo2Ego setting, the query is exocentric videos while egocentric videos form the candidate set. For both Ego2Exo and Exo2Ego settings, we consider $K = 20$ candidates.

S1.1.2 Implementation details

Training setting. As explicit pairing is not available in the training set, we propose a simple baseline approach to align egocentric videos and exocentric videos in the semantic space. In specific, we train a dual-encoder architecture consisting of a video encoder $f_v(\cdot)$ and a text encoder $f_t(\cdot)$ on both ego- and exo-videos and narrations using the contrastive loss, named as *co-training* in our experiments. Following [17, 24], we adopt a TimeSformer-B [3] as the video encoder and a clip [19] text encoder. We randomly sample 4 frames as input. The model is initialized with weights pre-trained on Ego4d video and text pairs [13, 17]. We train the dual encoder model for 5 epochs with a fixed learning rate 1e-5 and a batch size of 32. At the inference stage, the text encoder is discarded and only the video encoder is used. For each query, we compute its video representation with K features of the candidate videos and select the one with highest cosine similarity as the model prediction.

Network architecture. To leverage the gaze information in associating egocentric and exocentric videos, we further propose a multi-view branch for the video encoder [23]. One branch encodes the original video while the other branch encodes the gaze cropped video, as illustrated in Fig. S1. The feature of the original video cross-attends to the gazed video feature every at the 5th, 8th, and 11th transformer block, enabling multi-scale feature fusion for improved visual representation. For exocentric videos, we simply input the original video to the gaze branch.


Contrastive loss
Video Encoder
Text Encoder
Use a spatula in the right hand to stir-fry the vegetables in the pan.
Contrastive loss
Gaze Encoder
Video Encoder
Text Encoder
Use a spatula in the right hand to stir-fry the vegetables in the pan.

(a) Dual architecture

(b) Dual architecture w/ gaze branch

Figure S1. Cross-view association network with naive dual architecture (a) and improved architecture with additional gaze branch (b).

S1.1.3 Annotation details

The process of pair construction consists of five stages: (1) Scenario Matching. We gather all the egocentric and exocentric videos under the same scenario (e.g. cooking the same dish or conducting the same experiment) into each group. (2) Noun and Verb Matching. Based on the noun and verb vocabularies, for each group, we pair an egocentric caption with another if they contain exactly the same nouns and verbs. (3) Sentence Matching with LLM. We ask the LLM (e.g. ChatGPT) to determine whether each ego-exo caption pair obtained in stage 2 describes the same activity at sentence-level, reducing the linguistic ambiguity caused by word matching. (4) Negative Sampling. We randomly choose video clips from the same video as negative samples in the candidate set. (5) Two-round Manual Verification. We manually check the semantic meaning of each ego-exo pair and corresponding ego-exo video to make sure the exact match. This verification is performed in two rounds by two different individuals. In total, the size of the validation/test set is 868/2200. As stated in the main manuscript, we do not provide such pairs for the training set and leave the modeling of cross-view association on unpaired samples to be further explored for the community.

S1.2. Cross-view action anticipation & planning

S1.2.1 Detailed task definition

Task definitions of cross-view action anticipation and planning have followed the previous benchmarks of [7] and [13]. Our cross-view benchmark extends on the original task setting and focuses on mutual assistance between egocentric and exocentric video data.

Action anticipation. The action anticipation task focuses on forecasting the verb and noun categories of the subsequent fine-level action at $\tau = 1$ second into the future. Considering a fine-level action segment $a = (s, e, c)$, where s, e, and c represent the start time, end time, and category of a respectively, the model is restricted to observing video data only up to time $s - \tau$. The model's objective is to predict the forthcoming action, encompassing relevant verbs and nouns. The performance of the model in this benchmark is evaluated using class-mean Top-5 recall, as outlined in [7].

Action planning. The objective of the action planning task is to generate the next K steps of coarse-level actions. Considering N_a fine-level action segments $A = \{a_i = (s_i, c_i)\}_{i=1}^{N_a}$, where s_i (ensuring $s_i < s_{i+1}$) and c_i represent the start time and category of a_i respectively, the model is limited to observing video data up to time s_i and is tasked with forecasting the K actions $s_i, ..., s_{i+K-1}$ into the future. For evaluation purposes, we adopt ED@K as the metric, following the approach outlined in Ego4D LTA [13]. In our specific configuration, we set K to 8 and sample 5 predicted sequences for evaluation.

Cross-view benchmark. In our cross-view benchmark, we begin by assessing zero-shot cross-view action understanding. Following this, we employ various methods to leverage information in one view to assist the understanding in the other view. Thus, this benchmark is focused on designing approaches that utilize both ego and exo-view data to enhance the cross-view performance. Figure S2 shows the overall framework of our cross-view benchmark for action anticipation and planning. Figure S3 further illustrates our various cross-view settings.

S1.2.2 Implementation details

Network architecture. To adapt our cross-view training settings, we rely on the TA3N [5] code base, acknowledged for its clarity and comprehensibility, and widely adopted in recent research. We employ CLIP [19] as the feature extractor for generating frame-level video features. Both action anticipation and planning tasks entail leveraging historical information to forecast future actions. Thus, we input a 2-second context into the model. Within the specified temporal range, we uniformly sample 5 frames as the input. Utilizing the 3D feature map extracted by TA3N [5], we perform average pooling to condense the feature map into a vector $v \in \mathbb{R}^d$. We employ a projector $\mathbf{W}_{anti}$ to predict C_{anti} classes for the action anticipation task, where C_{anti} is the number of verb or noun categories. For the action planning task, we use a projector $\mathbf{W}_{plan}$ to predict $C_{plan} \times K$ classes, where C_{plan} is the number of coarse-level categories and K (set to 8) is defined in Sec S1.2.1.

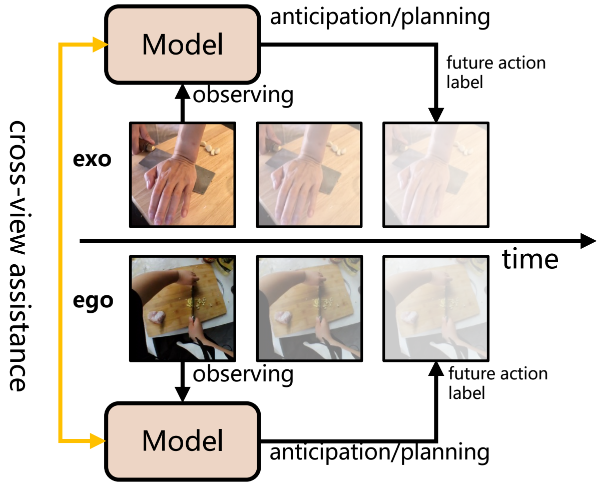


Figure S2. Overall framework of cross-view action anticipation and planning. The model observes the past video and tries to anticipate the next fine-level action (action anticipation) or the next K steps of the coarse-level actions (action planning). The model gets assistance from the knowledge in the other view.

Training. We first introduce the training settings of both tasks. Given the anticipation logits y_{anti} produced by the model and the corresponding ground truth $\hat{y}_{anti}$, we employ the standard cross-entropy loss for supervision:

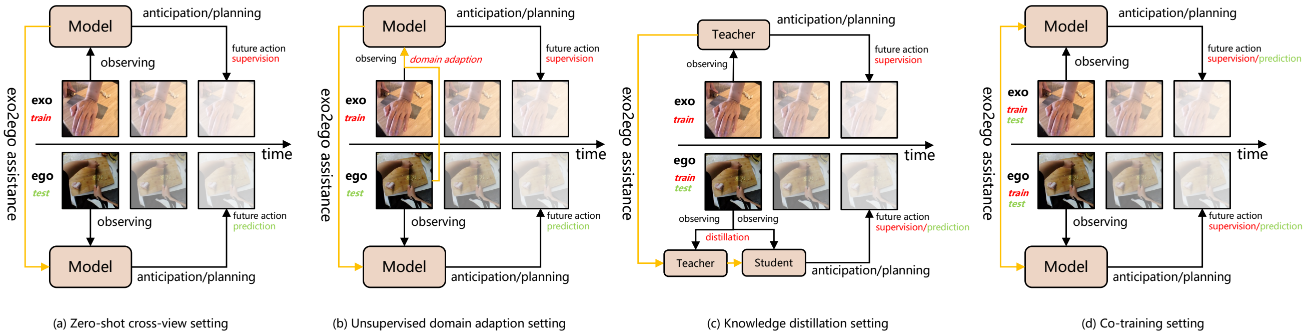
$$\mathcal{L}_{anti} = \mathcal{L}_{CE}(y_{anti}, \hat{y}_{anti}). \tag{1}$$

For an action sequence $y_{plan}^1, \dots, y_{plan}^K$ predicted by the action planning model, the loss function is defined as:

$$\mathcal{L}_{plan} = \frac{1}{K} \sum_{i=1}^{K} \mathcal{L}_{CE}(y_{plan}^i, \hat{y}_{plan}^i). \tag{2}$$

The model is trained using the SGD optimizer with a learning rate set to 1e-2 and the training process spans 40 epochs.

Zero-shot cross-view setting. In the zero-shot cross-view setting, the model is initially trained on data in one view and directly tested on data in the other view. This setting is crucial for understanding how well a model trained on data from one perspective can adapt to and accurately interpret data from another perspective, without any additional training specific to that new viewpoint. Figure S3(a) illustrates the procedure of the "exo2ego" cross-view setting, where



(a) Zero-shot cross-view setting

(b) Unsupervised domain adaption setting

(c) Knowledge distillation setting

(d) Co-training setting

Figure S3. Four settings for cross-view action anticipation and cross-view action planning. (a) The zero-shot setting directly evaluates the model trained on one view on the test data of the other view. (b) The unsupervised domain adaptation (UDA) setting involves leveraging data from another view, but without using the labels associated with this data. (c) In the knowledge distillation setting, for a model in one view, a teacher model trained on the other view is used to provide assistance. (d) The co-training setting directly uses the data and labels of both views. (a) to (d) represent 4 increasing degrees of cross-view information usage.

the model is first trained on exocentric data and then tested on egocentric data. The "ego2exo" setting works vice versa.

Unsupervised domain adaptation setting. In the unsupervised domain adaptation setting, the training process involves using data and labels from the source view, plus the video data from the target view. The annotations from the target view are not used. Figure S3(b) illustrates the "exo2ego" cross-view setting, where exocentric data serves as the source domain, and egocentric data serves as the target domain. In addition to task supervision, the overall loss function also contains a domain adaption loss derived from TA3N [5] for unsupervised domain adaptation settings:

$$\begin{aligned}\mathcal{L}_{DA} = &\frac{1}{N_S}\sum_{i=1}^{N_S}\mathcal{L}_y^i + \frac{1}{N_{S\cup T}}\sum_{i=1}^{N_{S\cup T}}\gamma\mathcal{L}_{ae}^i \\ &- \frac{1}{N_{S\cup T}}\sum_{i=1}^{N_{S\cup T}}(\gamma^s\mathcal{L}_{sd}^i + \gamma^r\mathcal{L}_{rd}^i + \gamma^t\mathcal{L}_{td}^i).\end{aligned} \tag{3}$$

Therefore, the overall loss function for both tasks under this setting is

$$\mathcal{L} = \mathcal{L}_{anti/plan} + \mathcal{L}_{DA}. \tag{4}$$

Knowledge distillation setting. In the knowledge distillation setting, the training process comprises two stages: (1) training the teacher model on the data in one view, and (2) training the student model on the data in the other view, meanwhile distilling knowledge from the teacher model. Figure S3(c) depicts the "exo2ego" cross-view setting, where the teacher model is trained on exocentric data, and the student model is trained on egocentric data. In addition to task supervision, the overall loss function for training the student model also contains a knowledge distillation loss for knowledge distillation settings. Specifically, we use L2 loss to minimize the feature y^S_{feat} and y^T_{feat} output by the student and teacher:

$$\mathcal{L}_{KD} = \mathcal{L}_{L2}(y^S_{feat}, y^T_{feat}). \tag{5}$$

Therefore, the overall loss function of the teacher and student for both tasks under this setting is

$$\mathcal{L}_{teacher} = \mathcal{L}_{anti/plan}, \tag{6}$$

$$\mathcal{L}_{student} = \mathcal{L}_{anti/plan} + \mathcal{L}_{KD}. \tag{7}$$

Co-training setting. In the co-training setting, exocentric and egocentric data are both used to train the model. The model is then evaluated on the test set of the egocentric and exocentric data. Figure S3(d) depicts the "exo & ego" co-training setting.

S1.2.3 Annotation details

Cross-view action anticipation. The annotation process for cross-view action anticipation involves three stages: (1) extracting verbs and nouns for each fine-level action clip, (2) aligning the closed categories of training, validation, and testing set across egocentric and exocentric videos, (3) restricting the closed set to the intersection of categories present in all egocentric and exocentric videos, and (4) managing the long-tail distribution of the data by filtering out categories that occur less than 1/100 of the highest occurrence category. We delete all video clips without any label. As a result, this task contains 19/31 verb/noun categories. The size of the egocentric train/validation/test set is 34.5k/7.7k/17.3k, and the size of the exocentric train/validation/test set is 6.1k/2.1k/4.8k.

Cross-view action planning. Cross-view action planning utilizes coarse-level annotations with a total of 27 classes for training, validation, and testing. We sort all action steps in

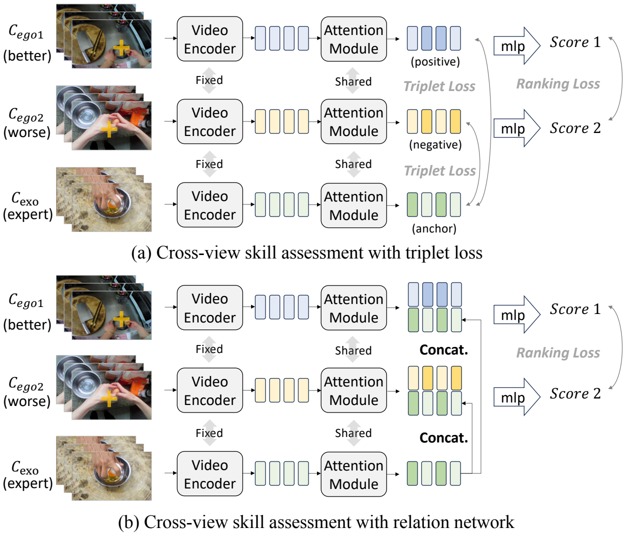
Method	Gaze	Anticipation↑				Planning↓	
		Ego-V	Ego-N	Exo-V	Exo-N	Ego	Exo
Exo-only	✗	30.7	23.5	**40.9**	**42.5**	83.5	**74.6**
Ego-only	✗	33.4	37.6	28.7	18.0	82.3	83.7
Ego-only	✓	**40.9**	**52.3**	37.5	37.6	**79.0**	81.8
Ego-only	Center	33.4	38.8	33.1	33.7	81.2	84.4
Unsupervised Domain Adaption							
Ego2Exo	✗	34.1	38.0	34.2	28.4	82.1	83.5
Ego2Exo	✓	**41.0**	**53.7**	37.2	37.3	**81.5**	83.8
Exo2Ego	✗	31.6	24.2	39.9	**42.4**	82.9	77.4
Exo2Ego	✓	34.1	31.5	**40.2**	42.3	81.8	**76.9**
Knowledge Distillation							
Ego2Exo	✗	30.7	25.1	**41.5**	**47.6**	83.0	75.1
Ego2Exo	✓	30.6	25.3	41.0	47.1	83.1	**74.6**
Exo2Ego	✗	34.6	38.3	30.1	18.9	81.9	84.9
Exo2Ego	✓	**41.2**	**55.9**	37.0	39.8	**79.0**	82.6
Co-training							
Ego & Exo	✗	33.9	37.3	**40.3**	46.7	82.0	74.8
Ego & Exo	✓	**41.6**	**52.9**	39.6	**47.9**	**78.3**	74.4

Table S1. Results of cross-view action anticipation and planning benchmarks on the validation set. For anticipation, the class-mean Top-5 recall is used as the evaluation metric (higher is better). For planning, the Edit distance is used as the evaluation metric (lower is better). Gray cells show the cross-view performance.

each video by their start time. Consequently, this task is oriented towards predicting potential sequences of future action starts. After filtering, we obtain 2.1k/0.8k/1.2k action steps in the egocentric train/validation/test set and 2.4k/0.3k/0.4k action steps in the egocentric train/validation/test set. Note that it is also possible to use the fine-level action annotations for this task, which will result in a much larger dataset split. We do not use this setting since we observe a large variation in the fine-level actions due to practical issues such as environmental constraints and unskilled performance. We believe the combination of our cross-view anticipation and cross-view planning can well evaluate the ability to bridge ego-exo procedural activities at both clip-level and task-level.

S1.2.4 Additional results

Table S1 presents the results of our baseline models on the validation set for the cross-view action anticipation and planning benchmarks. In the first block, zero-shot cross-view evaluation (*e.g., Exo-only* evaluated on ego view, and *Ego-only* evaluated on exo view) results in the lowest performance levels. This outcome underscores the challenge of applying learned representations from one perspective directly to another without any intermediary processing or adaptation. A significant improvement in zero-shot cross-view performance is observed with the introduction of gaze-cropped inputs. This enhancement suggests that **gaze can be an effective bridge for the ego and exo actions**. Further improvements in performance are noted when implementing methods such as Unsupervised Domain Adaptation (UDA), Knowledge Distillation (KD), and Co-Training (CT). The results also demonstrate that the extent of performance improvement varies across different cross-view settings. This variation highlights the complexity of bridging activities in ego and exo views and the importance of selecting the most appropriate method based on the specific requirements of each task.



(a) Cross-view skill assessment with triplet loss

(b) Cross-view skill assessment with relation network

Figure S4. Cross-view referenced skill assessment with triplet loss and relation network.

S1.3. Cross-view referenced skill assessment

S1.3.1 Detailed task definition

Our training dataset comprises the following components: (1) Egocentric Video Pairs: Denoted as P, each pair $(C_{ego1}, C_{ego2}) \in P$ is arranged such that video C_{ego1} displays better skill than C_{ego2}. (2) Accompanying Gaze Sequences: For every pair of egocentric videos $(C_{ego1}, C_{ego2}) \in P$, corresponding gaze sequences (g_1, g_2) are provided. (3) Exo-View Expert Demonstration: Each pair $(C_{ego1}, C_{ego2}) \in P$ is accompanied by an expert demonstration video C_{exo}, showcasing the same action as in C_{ego1} and C_{ego2} from an exo-view perspective. The objective is to develop a ranking function $f(\cdot)$ that adheres to the condition $f(C_{ego1}) > f(C_{ego2})$ given (g_1, g_2) and C_{exo} as the reference.

S1.3.2 Implementation details

Network architecture. As shown in Fig. S4, we assume C_{ego1} exhibits a higher skill level compared to C_{ego2}. Built upon a pairwise ranking skill assessment model RAAN [9], we employ different video encoders, including I3D [4] and VideoMAE [21], to extract video features from C_{ego1}, C_{ego2},

	#video clip	#valid pairs	Av. length	Corr. exo	Gaze
EPIC-Skills [8]	216	2592	85s	✗	✗
BEST [9]	500	16782	**180s**	✗	✗
Infant Grasp [16]	94	3318	5s	✗	✗
Ours	**3304**	**34239**	10s	✓	✓

Table S2. Comparison of skill assessment datasets based on human pairwise ranking annotation.

and C_{exo}. The features are processed by an attention module as described in [9] and resulting in refined features F_{ego1}, F_{ego2}, and F_{exo}. Then, we apply two different approaches to leverage the reference exo-view demonstration video: 1) Triplet loss (TL). We designate F_{exo} as the *anchor*, F_{ego1} as the *similar item (positive)*, and F_{ego2} as the *dissimilar item (negative)*. Then, we apply a triplet margin loss with margin = 1, to aid the model in understanding that the anchor is closer to the positive than the negative item. In our scenario, C_{ego1} demonstrates a skill level closer to the expert. 2) Relation network (RN). Inspired by [20], we implement a relation network that concatenates the features of the ego and exo clips. Precisely, we set $F_{ego1} = Concat(F_{ego1}, F_{exo})$ and $F_{ego2} = Concat(F_{ego2}, F_{exo})$. By combining ego and exo features, this network is designed to implicitly discern which of the two egocentric video clips bears a closer relation to the demonstration video in terms of skill level. Finally, the refined features F_{ego1} and F_{ego2} are processed by an MLP to regress skill scores for the two ego videos.

Training. For the ego branch of our network, we employ the training objectives from [8, 9]. 1) a margin ranking loss is applied on the finally generated scores to ensure $ego1$ is ranked higher than $ego2$. 2) a disparity loss is applied within the attention module to prevent the network from getting trapped in local minima during training 3) a rank-aware loss and a diversity loss are also applied following [9]. Besides the ego branch, to leverage the exo demonstration video, we propose to utilize a triplet loss to aid the model in comprehending that $ego1$ exhibits skills more akin to those of an expert.

S1.3.3 Annotation details

We include two types of annotations for skill level. The first type is self skill assessment. During data collection, subjects are asked to assess themselves on various aspects, including their familiarity with cooking environments, the number of times they have completed the task previously, the frequency of performing the task, the typical duration required to complete the task, and whether they've taught others how to perform the task. Based on the self-evaluation results, we have observed a considerable diversity in subjects' skill levels, which motivates us to craft the skill assessment benchmark. One related work is HoloAssist [22] where they show the distribution of the performers' familiarity with the tasks measured by a self-reported score (0-10) by the subjects. However, no related benchmarks is provided by HoloAssist.

Method	Gaze	Egg Cracking	Peeling	Stir-fry	Cutting	Avg
Ego pairs only						
Who's better* [8]	✗	79.08	74.52	82.87	78.35	78.71
RAAN* [9]	✗	83.09	77.30	86.25	82.86	82.23
Who's better* [8]	✓	79.95	75.67	82.94	79.21	79.44
RAAN* [9]	✓	**84.79**	78.97	86.14	82.96	**83.22**
Ego pairs + Exo						
RAAN* [9] + RN	✗	83.14	77.39	**86.47**	82.48	83.01
RAAN* [9] + TL	✗	81.99	77.48	86.16	82.54	82.04
RAAN* [9] + RN	✓	82.84	78.75	86.19	**83.33**	82.78
RAAN* [9] + TL	✓	83.64	**79.41**	86.14	83.07	83.07

Table S3. Ranking accuracy of cross-view referenced skill assessment. "*" means using VideoMAE [21] extracted video features. In the upper part of the table, only ego video pairs are used, while in the lower part, exo demonstrations are incorporated by "RN": relation network and "TL": triplet loss.

One drawback of self-evaluated skill level is that individuals may showcase varying skill levels in each video instance, even across multiple attempts [8]. As a more objective complement of the self-assessment, we adopt the pairwise comparison approach [8, 9, 16] for annotation. We provide annotators with four criteria: Fluency, Speed, Proficiency, and Skillfulness. These standards serve as the basis for their ranking assessment. From the annotation results, we find 40% of the rankings deviate from the rankings based on the self-evaluations of the two subjects in the video pair. This finding supports that relying solely on self-evaluation is inadequate for creating a robust skill assessment benchmark.

As shown in Table S2, our dataset stands out as the only skill assessment dataset featuring the gaze modality and corresponding exo-view demonstration videos. Notably, our dataset surpasses previous ones in both video clip quantity and valid pair numbers. We follow the setting in [8, 9] to employ 4 individuals to rank the same video pair to ensure credibility. We exclude annotations with fewer than 3 consistent opinions instead of 4 to ensure our dataset contains challenging pairs. Regarding action categories, our dataset comprises 6 actions: Egg cracking, Peeling, Stir-fry, Cutting into chunks, Slicing into strips, and Chopping into pieces. In the main paper, we merge the last three actions into a comprehensive category labeled "Cutting", encompassing various knife-using skills.

S1.3.4 Additional results

Results with I3D [4] feature are shown in Table 5 of the main manuscript. We show the results with VideoMAE [21] feature in Tab. S3. Comparing results from the two tables, we observe an overall increase in performance in all cases in Table S3 because of the stronger backbone model. While

Method	Gaze	Val						Test					
		Ego			Exo			Ego			Exo		
		Acc	Edit	F1@Avg	Acc	Edit	F1@Avg	Acc	Edit	F1@Avg	Acc	Edit	F1@Avg
Exo-only	✗	27.99	33.81	6.95	**38.64**	**40.28**	**23.64**	24.80	35.29	8.13	**42.65**	**37.32**	**20.14**
Ego-only	✗	65.35	44.25	40.91	19.81	21.18	7.31	62.50	44.29	39.43	25.09	22.45	6.89
Ego-only	✓	**66.01**	**46.60**	**41.78**	21.08	23.29	7.44	**65.99**	**48.83**	**42.95**	25.14	22.28	8.17
Ego-only	Center	62.14	45.92	36.60	19.13	23.02	7.37	60.42	46.29	36.52	22.83	22.55	7.2
Unsupervised Domain Adaption													
Ego2Exo	✗	65.52	44.15	40.78	20.78	22.02	7.55	63.41	44.35	40.15	25.67	23.12	7.45
Ego2Exo	✓	**66.12**	**46.42**	**42.11**	21.78	23.99	8.43	**65.91**	**48.81**	**42.78**	25.87	23.44	8.56
Exo2Ego	✗	28.76	33.76	7.56	38.44	39.98	23.61	25.34	35.75	8.67	**42.56**	**36.71**	20.03
Exo2Ego	✓	29.12	34.91	8.49	**38.47**	**40.01**	**23.69**	27.78	39.12	9.87	42.45	36.66	**21.12**
Knowledge Distillation													
Ego2Exo	✗	33.25	25.68	9.18	39.62	40.36	20.07	32.00	26.70	9.73	**43.03**	38.06	**20.44**
Ego2Exo	✓	31.16	28.62	8.60	**40.28**	**42.24**	**23.09**	29.17	28.49	8.45	41.68	**36.33**	20.12
Exo2Ego	✗	65.91	45.80	**42.37**	22.65	17.86	7.16	62.77	46.73	41.12	28.20	17.78	6.06
Exo2Ego	✓	**66.02**	**47.98**	41.71	23.47	24.24	8.12	**64.53**	**49.36**	**42.24**	28.34	23.49	7.80
Co-training													
Ego & Exo	✗	64.43	42.00	37.40	37.93	**42.18**	**23.20**	61.75	41.45	36.43	41.07	**38.73**	21.93
Ego & Exo	✓	**66.57**	**44.36**	**39.87**	**41.89**	39.13	22.70	**65.57**	**44.30**	**39.62**	**42.27**	35.10	**22.50**

Table S4. Results on cross-view temporal action segmentation benchmark. Gray cells show the cross-view performance.

we can still observe performance gain when adding Exo reference video, this improvement is less significant compared with the corresponding table in the main manuscript. We suspect that this variation is attributed to the varying degrees of influence that the intrinsic properties of the extracted features exert on the observed enhancements.

S1.4. Cross-view action segmentation

S1.4.1 Detailed task definition

The action segmentation task in our framework is focused on both categorizing each time step and delineating action steps within procedural videos. Given a lengthy video V comprising N_V frames at 25 FPS, the model is tasked with classifying the category of each frame in the video. The evaluation metric includes assessing frame-level classification accuracy. Additionally, sequence-level metrics such as edit distance and instance-level metric F1 are employed for further evaluation [11]. The extended cross-view action segmentation benchmark, similar to cross-view action anticipation and planning, aims to pursue performance improvement by receiving aid from other views.

S1.4.2 Implementation details

Network structure. We employ I3D [4] as the feature extractor to generate temporal features, following the methodology of previous work [11]. To implement our various training settings, we utilize the SSTDA [6] code base. For both training and testing, we downsample feature sequences and label sequences by a factor of 5 for efficiency.

Training. The loss function used to train the action segmentation task is derived from SSTDA [6]. The model consists of multiple stages. The overall loss function for a single stage is a combination of the classification loss and smoothing loss:

$$\mathcal{L}_{seg} = \mathcal{L}_{cls} + \gamma \mathcal{L}_{smooth}. \tag{8}$$

The model is trained using the Adam [14] optimizer with a learning rate set to 1e-3 and the training process spans 150 epochs.

Cross-view settings. Similar to cross-view action anticipation and planning, action segmentation also performs four cross-view settings. Though cross-view action segmentation shows different input and output, which yields dense prediction, the implementation of cross-view settings is consistent with Section S1.2.2.

S1.4.3 Annotation details

The annotation for the cross-view action segmentation task is derived from coarse-level annotations. To create non-overlapping segment annotations for temporal action segmentation, we establish the center point of the overlapping portion of two segments as their boundary. Subsequently, we introduce background segments labeled as “no action” in temporal regions not covered by action annotations. Finally, we obtain 173/57/85 videos in the egocentric train/validation/test set and 210/24/32 videos in the exocentric train/validation/test set.

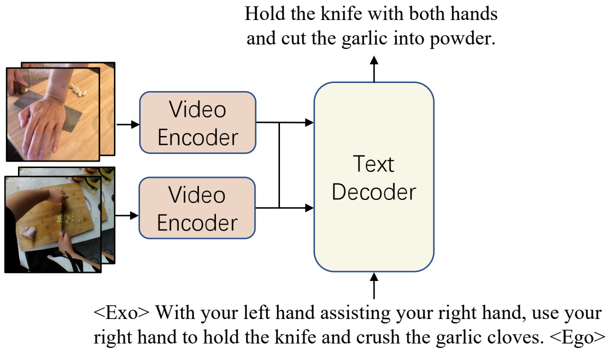

Hold the knife with both hands and cut the garlic into powder.
Video Encoder
Video Encoder
Text Decoder
<Exo> With your left hand assisting your right hand, use your right hand to hold the knife and crush the garlic cloves. <Ego>

Figure S5. Cross-view referenced captioning with a video encoder and a text decoder.

S1.4.4 Experimental results

Table S4 presents the results of our baseline models on the validation set and test set for the cross-view action segmentation benchmarks. These results mirror the trends observed in the action anticipation and planning benchmarks: without any assistance from another view, the models can only perform well on the test data in the same view. The inclusion of gaze data enhances model performance in both the ego-only setting and the cross-view setting. This suggests that focusing on areas of visual attention, as indicated by gaze data, is beneficial for better understanding and segmenting actions, regardless of the viewpoint. When information from another view is leveraged, all three methods – Unsupervised Domain Adaptation (UDA), Knowledge Distillation (KD), and Co-Training (CT) – contribute to performance improvements in the cross-view setting. Each method offers a different mechanism for integrating cross-view insights, thus aiding in the segmentation task. Reflecting the varying degrees of labeled data utilization, Co-Training (CT) tends to outperform Knowledge Distillation (KD), which in turn outperforms Unsupervised Domain Adaptation (UDA).

S1.5. Cross-view referenced video captioning

S1.5.1 Detailed task definition

Cross-view referenced video captioning evaluates the model's captioning ability to leverage cross-view information for caption generation. Our motivation is that egocentric videos require extensive efforts to collect, and are thus limited in scale and diversity. In contrast, large-scale exocentric videos can be easily sourced from the Internet. The question is, *how to leverage such exocentric videos to help the understanding of limited egocentric videos?*.

Formally, at the training stage, we have egocentric videos of limited size $\{(V_1^{\text{ego}}, T_1^{\text{ego}}), ..., (V_N^{\text{ego}}, T_N^{\text{ego}})\}$ with N samples, and exocentric videos $\{(V_1^{\text{exo}}, T_1^{\text{exo}}), ..., (V_M^{\text{exo}}, T_M^{\text{exo}})\}$, where $N \ll M$. Each video is paired with a fine-grained text description. The goal is to train a cross-view video captioning model $f(\cdot)$ using exocentric videos as references. At the inference stage, the model is required to generate the captions of the testing egocentric videos, given the other set of exocentric videos as references. Note that, $N \leq M$ only holds for the training set. In particular, we limit the number of the referenced exocentric videos by formulating the task as a K-shot captioning [1] problem, where K denotes the maximum number of exocentric videos that the model is allowed to use during inference. The inference process can be formulated as $f(V^{\text{ego}}|\{(V_1^{\text{exo}}, T_1^{\text{exo}}), ..., (V_K^{\text{exo}}, T_K^{\text{exo}})\}$ In practice, we consider three settings, 0-shot, 1-shot, and 2-shot.

S1.5.2 Annotation

We directly apply the fine-grained language annotations in our dataset. The referenced exocentric videos are randomly selected for training/validation/testing, respectively. The training set only contains 1000 egocentric videos with 6270 referenced exocentric videos. For the validation/testing set, there are 8181/2143, 18243/4930 egocentric videos and referenced exocentric videos, respectively.

S1.5.3 Implementation details

For the baseline model, we choose a Flamingo-style captioning model [1, 2, 15], an advanced vision-language model designed for few-shot vision-language tasks, as shown in Fig. S5. Please refer to [1] for the architectural details. We simply pre-pend the referenced video(s) before the input video, and add the referenced caption as prompts to the text decoder. We train the model for 3 epochs using the Adam optimizer, with an initial learning rate of 1e-4 and a batch size of 32. We adopt the cross-view association network (Fig S1(b)) to select referenced samples.

S1.5.4 Results

Table S5 lists the cross-view referenced captioning performance. We consider three baseline models: (i) **Single-view** models include *Ego-only* and *Exo-only*, where the former one merely adopts egocentric videos for training and inference without seeing exocentric videos. The *Exo-only* model uses all referenced exocentric videos for training, and it is then evaluated on egocentric videos. (ii) **Co-training model** is trained on both egocentric videos and referenced exocentric videos, and transferred to egocentric test videos. (iii) **Referenced-training** model refers to our model introduced in Fig. S5, where the model leverages one (1-shot) or two (2-shot) exocentric videos to make predictions. As shown in Table S5, both the co-training model and referenced-training models outperform single-view models. For co-training models, the performance gain is due to the increased number of

Method	Ref Train	Ref Infer	Validation				Test			
			BLEU-4	METEOR	ROUGE-L	CIDER	BLEU-4	METEOR	ROUGE-L	CIDER
Exo-only	✗	✗	0.024	0.126	0.212	0.122	0.023	0.124	0.208	0.112
Ego-only (0-shot)	✗	✗	0.049	0.116	0.270	0.332	0.048	0.112	0.266	0.314
Co-training										
Ego+Exo	✓	✗	**0.069**	**0.139**	**0.294**	**0.460**	**0.068**	**0.137**	**0.290**	**0.427**
Ref-training										
Ego+Exo (1-shot)	✓	✓	0.047	0.121	0.275	0.378	0.046	0.123	0.275	0.372
Ego+Exo (2-shot)	✓	✓	0.044	0.119	0.272	0.372	0.045	0.122	0.272	0.380

Table S5. Cross-view referenced captioning performance. "Ref Train/Ref Infer" refers to whether the model uses exocentric videos during training/inference.

training data (ego+exo), compared to ego-only and exo-only counterparts. In terms of referenced-training models, they generally outperform the ego-only counterpart by additionally incorporating exocentric videos in the model. Results of both the co-training model and referenced-training models indicate the effectiveness of utilizing exocentric videos in improving egocentric video captioning when the data is of limited scale.

S1.6. Zero-shot action recognition

We assess the zero-shot classification performance of verb and noun subsets. In cases where samples have multiple labels, we straightforwardly replicate the samples for testing. Our testing procedure follows CLIP [19], evaluating the vision-language models based on Top-1 and Top-5 accuracy.

S1.6.1 Annotation

In this task, our evaluation specifically addresses zero-shot transfer within the closed set and does not encompass cross-view settings. It is noteworthy that this annotation does not require ensuring consistent categories between egocentric and exocentric datasets across their respective validation and testing sets. The size of the resulting egocentric verb-validation/verb-test/noun-validation/noun-test set is 14.4k/32.6k/20.2k/44.8k, and the size of the exocentric verb-validation/verb-test/noun-validation/noun-test set is 4.2k/10.4k/5.7k/13.1k, respectively.

S1.6.2 Implementation details

We use the 16 prompts from the zero-shot classification on Kinetics [4] for verb and noun subsets. These prompts are listed in Table S7. We sample the center frame of each video clip, and use OpenAI CLIP [19] to extract the visual features and textual features.

S1.6.3 Experimental results

Table S6 shows the performance of zero-shot action recognition. *Oracle* is the upper bound of accuracy, given that this is a multi-class action recognition problem. On both the validation set and the test set, the zero-shot performance on egocentric videos is worse than that on exocentric videos, particularly in the top-1 accuracy. This result indicates the limitation in cross-view action understanding of the current method.

S1.7. Fine-tuned action recognition

S1.7.1 Detailed task definition

We formulate the conventional Fully-supervised setting to a multi-label classification task. In assessing the performance of fully supervised action recognition, we employ the class-wise multi-label mean Average Precision (Marco mAP) evaluation metric due to the presence of multiple labels per clip. This evaluation protocol is reasonable because it matches the long-tail attribution of actions in `EgoExoLearn`.

S1.7.2 Annotation

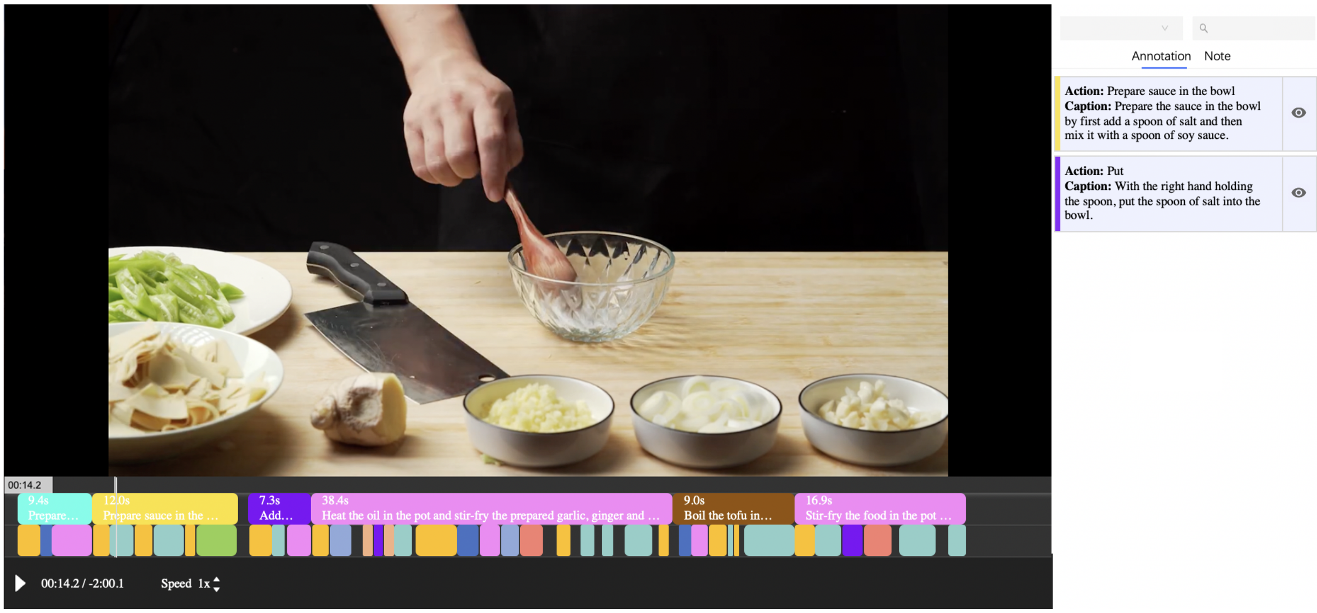
In this task, our evaluation focuses on the closed set and does not consider cross-view settings. Thus, our annotations ensure that egocentric and exocentric datasets maintain consistent categories across their respective training, validation, and testing sets. At last, this task contains 81/211 verb/noun categories in the egocentric set and 69/183 verb/noun categories in the exocentric set. The size of the egocentric train/validation/test set is 36k/8k/18k, and the size of the exocentric train/validation/test set is 6.2k/2.1k/4.8k.

S1.7.3 Implementation details

For evaluating this task, we utilize SlowFast-R50 [12] and MViT-Small [10] as the backbones. The weights pretrained on the Kinetics [4] dataset are employed for both backbones. Frames within each action clip are uniformly sampled and fed into the backbone. The multi-label classification task is supervised using the standard cross-entropy loss. Table S9 lists the training hyperparameters.

Model	Val								Test							
	Ego-Verb		Ego-Noun		Exo-Verb		Exo-Noun		Ego-Verb		Ego-Noun		Exo-Verb		Exo-Noun	
	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5
Oracle	56.14	99.79	39.72	99.29	50.06	99.74	36.74	97.70	55.41	99.66	46.55	99.69	45.77	99.72	37.00	97.59
CLIP [19]	7.89	22.71	7.08	19.26	9.49	22.62	7.70	20.45	6.96	21.95	6.39	18.19	9.02	20.99	7.09	19.93

Table S6. Results of zero-shot action recognition. *Oracle* denotes the upper bound of accuracy, because of the multi-label nature of clips in our dataset.



Action: Prepare sauce in the bowl
Caption: Prepare the sauce in the bowl by first add a spoon of salt and then mix it with a spoon of soy sauce.

Action: Put
Caption: With the right hand holding the spoon, put the spoon of salt into the bowl.

Figure S6. We use a web-based language annotation interface for the annotators. Annotators mark a segment of the video, select a category for this segment, and describe the segment based on the annotation requirement in their mother language.

S1.7.4 Experimental settings and results.

Table S8 shows the result of fine-tuned action recognition. MViT-S [10] (with 16 frames input) exhibits superior performance and generalization compared to the R50-based SlowFast [12] (with 4 frames for the slow branch and 32 frames for the fast branch as input). The results in Table S8 also reveal great potential improvement on more sophisticated model structures for this dataset.

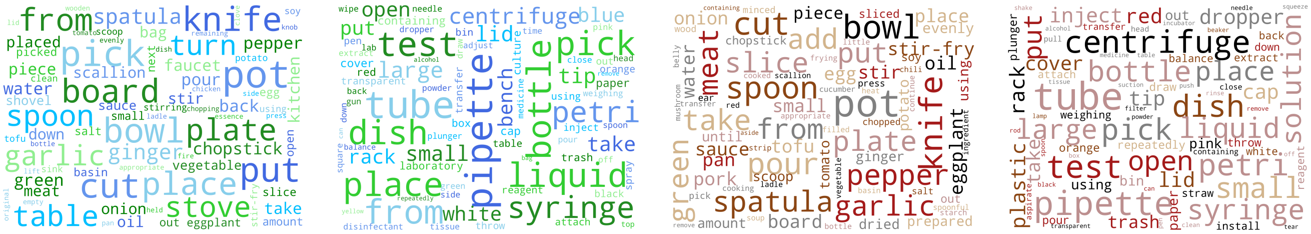
S2. Additional Dataset Details

S2.1. Language annotation

Different from previous datasets [7, 13, 22], our dataset includes two-level language annotations with manually annotated temporal boundaries. As described in Section 3.2 of the main manuscript, our annotation includes a coarse-level language annotation and a fine-level language annotation. We designed a web-based interface to facilitate the annotation. An example screenshot is shown in Figure S6.

For each video, the annotators are asked to quickly skim the video to grab the overall content, and then begin the annotation of each session. For the daily tasks, the annotators are instructed to describe each segment based on their own knowledge. For the tasks in specialized laboratories, we train the annotators showing them the process of the experiments, the technical terms of some tools/reagents (*e.g.*, pipette), and the purpose of each action step. To avoid describing objects that are impossible to determine visually (*e.g.*, the appearance of water and PBS reagent are exactly the same), we ask the annotators to describe their visual appearance instead (*e.g.*, pink reagent in a bottle with green cap). Figure S7 shows a word cloud of the language annotations separated by views and tasks. Figure S9 shows the distribution of lengths of the coarse and fine level language annotations. The average lengths of the coarse and fine level annotations are 21.5 seconds and 4.6 seconds, respectively.

Translation & Parsing. For all the non-English language annotations, we translate them into English using ChatGPT. We conduct a manual check on the translation quality and use



Ego-view, daily tasks

Ego-view, lab tasks

Exo-view, daily tasks

Exo-view, lab tasks

Figure S7. Word cloud of annotations separated by views and tasks.

#	Prompts
1	A photo of action {}.
2	A picture of action {}.
3	Human action of {}.
4	{}, an action.
5	{} this is an action.
6	{}, a video of action.
7	Playing action of {}.
8	{}
9	Playing a kind of action, {}.
10	Doing a kind of action, {}.
11	Look, the human is {}.
12	Can you recognize the action of {}?
13	Video classification of {}.
14	A video of {}.
15	The man is {}.
16	The woman is {}.

Table S7. Prompt templates used in the zero-shot action recognition task.

View	Model	Val		Test	
		Verb	Noun	Verb	Noun
Ego	Slowfast-R50 [12] 4×16	27.03	34.77	25.58	33.25
	MViT-S [10]	**29.83**	**39.45**	**28.16**	**36.46**
Exo	Slowfast-R50 [12] 4×16	15.79	22.08	11.71	16.65
	MViT-S [10]	**18.59**	**22.81**	**13.53**	**19.36**

Table S8. Results of fine-tuned action recognition. We utilize the multi-label mean Average Precision (mAP) evaluation metric because of the existence of multiple labels per clip. This choice is consistent with the methodology described in [18]. Specifically, we adopt macro mAP as the class-mean metric.

Google Translation API to translate again for unsatisfactory translations.

To effectively parse and analyze the annotations in our dataset, we employ a rule-based framework designed to extract verbs and nouns associated with specific actions of the left and/or right hand. The process is methodical and iterative to ensure the annotation quality. The overview of the parsing is as follows:

config	Egocentric	Exocentric
optimizer	AdamW [14]	
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$	
weight decay	1e-4 (Slowfast), 0.05 (MViT)	
learning rate scheduler	warmup constant	
learning rate	1e-4	
batch size	32	32
total epochs	20	30
flip augmentation	✓	
crop size	224	
randomresizedcrop	scale=(0.08, 1)	

Table S9. Training hyperparameters used for fine-tuned action recognition benchmark.

- Sentence Splitting: We begin by splitting the annotations into individual sentences using separators like commas. This step helps in isolating distinct actions or descriptions for more focused analysis.
- Keyword Identification and Extraction: For each split sentence, we use NLTK to identify keywords that indicate actions related to the left hand, right hand, both hands, etc. This involves analyzing the sentence structure and content to pinpoint relevant verbs and nouns. One challenge we encounter is the word "left" itself, which can be a verb in certain contexts. To address this, we temporarily mask the mentions of left and right hands in each sentence and then re-extract the verbs and nouns. This masking helps in distinguishing between the directional use of "left" and its use as a verb.
- Manual Review and Iteration: After the initial extraction, we conduct a manual review of the results to identify and correct any errors. This step is crucial for ensuring the accuracy and relevance of the extracted terms. If errors are found, we revisit the first and second steps, making necessary adjustments. This iterative process continues until the manual review yields satisfactory results.

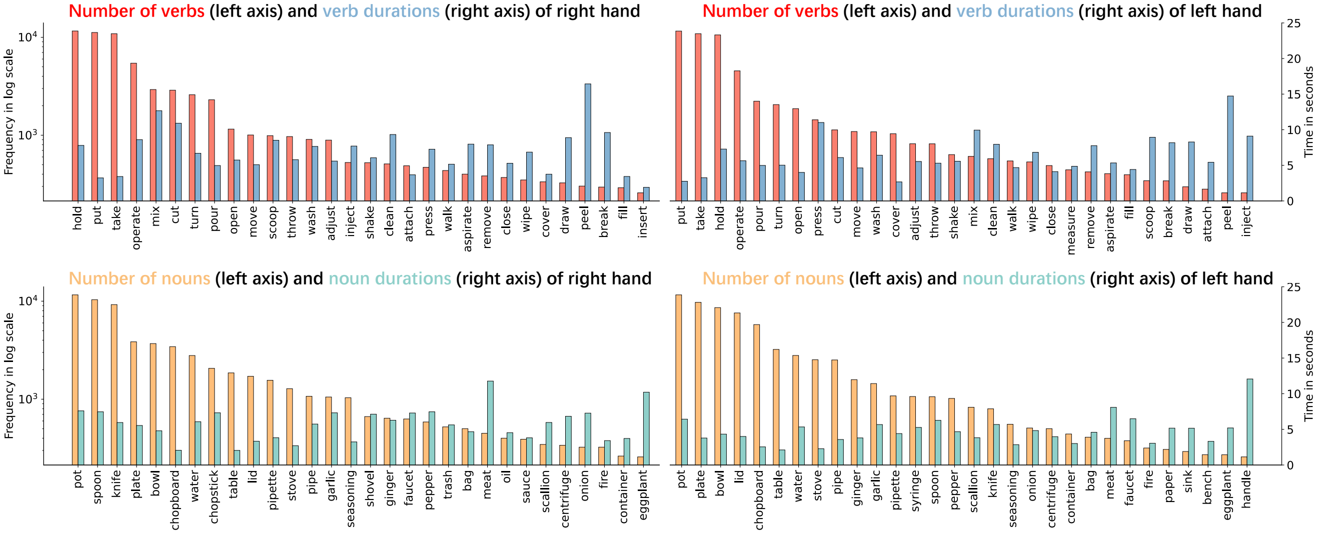


Figure S8. Occurrence and duration distribution of the annotated fine-level verbs and nouns associated with the left and right hands.

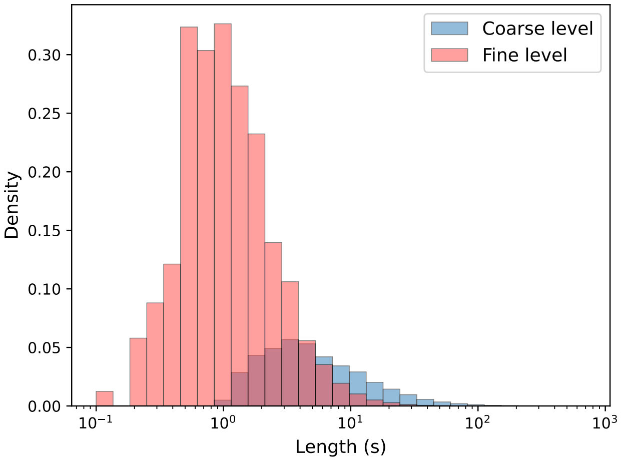


Figure S9. Distribution of the lengths of the coarse-level and fine-level language annotations.

Figure S8 shows the verbs and nouns extracted after associating with the left and right hands. We only show the top 30 categories due to the size limit.

S2.2. Post-processing.

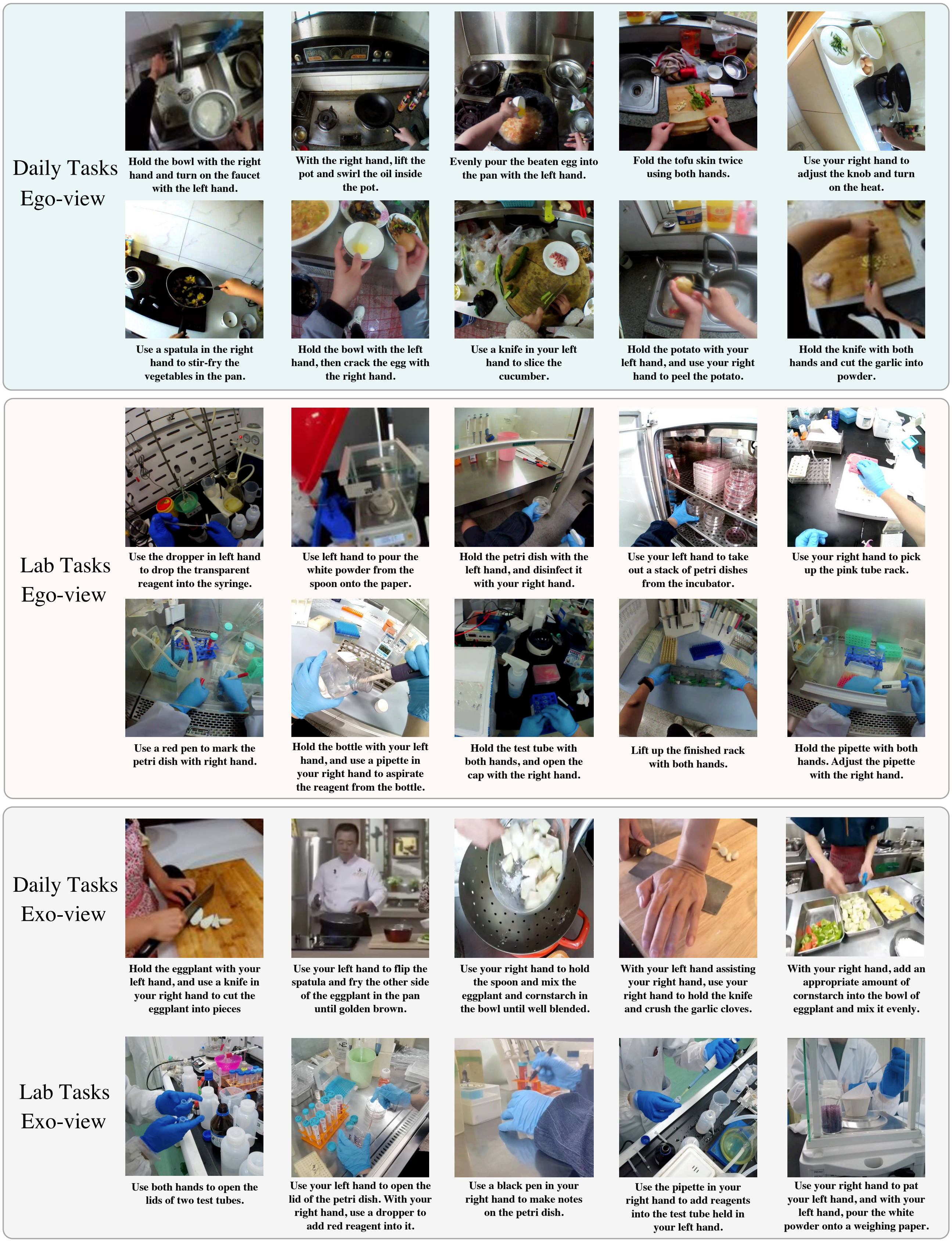
In dealing with the practical challenges of recording egocentric videos, particularly with Pupil Invisible devices that sometimes capture footage at variable frame rates due to issues like overheating, we employ post-processing for standardization. All videos are converted to a constant frame rate of 25fps to ensure uniformity and consistency in our dataset.

Additionally, our gaze data, which is recorded at a high frequency of 120Hz, provides detailed insights into the viewer's point of focus during the demonstration following process of the video. To ease the use of this gaze data, we align the timestamps of the egocentric camera with the eye-tracker. Once the alignment process is complete, we register each gaze data point to the temporally closest frame in the video. We then take the average of all gaze data points within one frame and use this as the final gaze data.

In Figure S10, we visualize the video frames along with the annotated fine-level language annotations. EgoExoLearn features a new demonstration following setting that is a complement to existing egocentric and ego-exo datasets. Meanwhile, as can be seen in Figure S10, compared with existing egocentric datasets, our language annotations contain much longer sentences, enabling our dataset to be used in the captioning benchmarks.

S2.3. IRB approval

We receive IRB approval before the data collection, adhering to the ethical standards and guidelines for research involving human participants. Participants involved in the study were provided with detailed consent forms and information sheets. These documents thoroughly explained the data capture process, the purpose of the study, and how the data would be used in the future. The consent forms, along with the information sheets, were reviewed and approved by the IRB to ensure they met all ethical standards and adequately informed participants. We maintain these documents and can provide them upon request for verification or further inquiry into our ethical and procedural practices during the data collection.



Hold the bowl with the right hand and turn on the faucet with the left hand.

With the right hand, lift the pot and swirl the oil inside the pot.

Evenly pour the beaten egg into the pan with the left hand.

Fold the tofu skin twice using both hands.

Use your right hand to adjust the knob and turn on the heat.

Use a spatula in the right hand to stir-fry the vegetables in the pan.

Hold the bowl with the left hand, then crack the egg with the right hand.

Use a knife in your left hand to slice the cucumber.

Hold the potato with your left hand, and use your right hand to peel the potato.

Hold the knife with both hands and cut the garlic into powder.

Use the dropper in left hand to drop the transparent reagent into the syringe.

Use left hand to pour the white powder from the spoon onto the paper.

Hold the petri dish with the left hand, and disinfect it with your right hand.

Use your left hand to take out a stack of petri dishes from the incubator.

Use your right hand to pick up the pink tube rack.

Use a red pen to mark the petri dish with right hand.

Hold the bottle with your left hand, and use a pipette in your right hand to aspirate the reagent from the bottle.

Hold the test tube with both hands, and open the cap with the right hand.

Lift up the finished rack with both hands.

Hold the pipette with both hands. Adjust the pipette with the right hand.

Hold the eggplant with your left hand, and use a knife in your right hand to cut the eggplant into pieces

Use your left hand to flip the spatula and fry the other side of the eggplant in the pan until golden brown.

Use your right hand to hold the spoon and mix the eggplant and cornstarch in the bowl until well blended.

With your left hand assisting your right hand, use your right hand to hold the knife and crush the garlic cloves.

With your right hand, add an appropriate amount of cornstarch into the bowl of eggplant and mix it evenly.

Use both hands to open the lids of two test tubes.

Use your left hand to open the lid of the petri dish. With your right hand, use a dropper to add red reagent into it.

Use a black pen in your right hand to make notes on the petri dish.

Use the pipette in your right hand to add reagents into the test tube held in your left hand.

Use your right hand to pat your left hand, and with your left hand, pour the white powder onto a weighing paper.

Figure S10. Examples of video frames and corresponding fine-level language annotations in our dataset.

S2.4. Tasks

Our dataset is collected for 5 types of daily tasks and three types of specialized laboratory tasks. The collection is performed in four different kitchens and three different specialized laboratories. The participants' ages range from 18 to 40 years with diverse occupations such as athletes, housekeepers, security guards, university students, and researchers. We carefully choose the daily tasks and specialized lab tasks such that a long series of procedures is needed before finishing. This can reflect the complexity of real-life activities meanwhile enabling our new benchmarks for ego-exo procedural activity bridging. Table S10 shows the names of the 8 tasks with an example procedure. In real recordings, the procedures are usually more complicated due to repetition and other practical issues related to the environment. Note that the scientific name of the specialized reagents are not included in the language annotations but are described using their visual appearance.

Acknowledgements. Hosted by Shanghai AI Laboratory, Shenzhen Institutes of Advanced Technology, and Nanjing University, this work is jointly completed by a team of researchers and students from multiple institutes. Authors can be contacted via email: Yifei Huang, Mingfang Zhang, and Lijin Yang at [hyf, mfzhang, yang-lj]@iis.u-tokyo.ac.jp, Guo Chen at 602022330002@smail.nju.edu.cn. Jilan Xu can be contacted via 18210240039@fudan.edu.cn, Baoqi Pei via 12321251@zju.edu.cn, Hongjie Zhang via nju.zhanghongjie@gmail.com, and Lu Dong via dl1111@mail.ustc.edu.cn. We give special appreciations to Yi Liu (yi.liu1@siat.ac.cn) for providing insightful ideas in the annotation phase, and Ruijie Zhang (zhangruijie@pjlab.org.cn) for the help in the arrangement of annotators.

Task name	Scenario	Example procedure
Task1: Twice-cooked Pork	Daily	1. Prepare spices: Take out and cut some scallion, ginger, and garlic for later use. 2. Prepare the pork: boil the pork together with scallion and ginger to remove impurities. Drain and set aside. Wash the pot if necessary. 3. Prepare vegetables: Take some pepper and onion, wash, and discard unused parts. 4. Cut vegetables: Cut the prepared vegetable into slices, and put the slices into a plate for later use. 5. Cut the pork: Remove the water on the pork. Use a knife to cut the pork into slices, and set aside for later use. 6. Stir-fry the pork: Add oil into heated pot. Then add the prepared spices into the pot. After stir-frying for about 10 seconds, add the pork into the pot. 7. Stir-fry the vegetables: Heat the pot and add oil, then add vegetables into the pot. Stir-fry until the vegetable is well-cooked, then add the pork into the pot. 8. Add seasoning: Add salt, soy sauce, sugar into the pot. Stir-fry a few times to evenly distribute the flavors. 9. Transfer: Transfer the cooked Twice-cooked Pork from the pot to a plate. Wash the pot if necessary.
Task2: Tofu Skin with Hot Pepper	Daily	1. Prepare tofu skin: Take out the tofu skin, fold them and cut the tofu skin into slices. 2. Prepare hot pepper: Take out some hot pepper, squeeze by hand and then cut into pieces. 3. Prepare spice: Take out and cut some scallion, ginger, and garlic for later use. 4. Boil tofu skin: Put some water into the pot, add baking soda. Boil the tofu skin until the water becomes cloudy. 5. Wash tofu skin: Take out the tofu skin and put them into cold water. Wash the tofu skin such that the smell of soda diminishes. Take out and drain water. 6. Prepare sauce in the pot: Heat up the pot and add some oil. Put the spices into the pot. Use a spoon to put some water, soy sauce, and salt into the pot and heat up until the water boils. 7. Cook tofu skin: Put the tofu skin into the pot, and continue to boil until the pot becomes dry. 8. Cook hot pepper: Add hot pepper into the pot, stir-fry for several times. 9. Transfer: Add some oil into the pot, then transfer the cooked dish into a plate.
Task3: Stir-fried potato, eggplant and green pepper	Daily	1. Prepare potato: Take out some potatoes, peel and clean them. 2. Prepare eggplant: Take out some eggplants, remove the stems, and clean them. 3. Prepare green pepper: Take out some green pepper, remove the stems, and clean them. 4. Cut green pepper: Squeeze the green pepper using the side of the knife, then cut them into pieces. 5. Cut eggplant: Rolling cut the eggplant into pieces. Use hand to squeeze water out of the eggplant pieces. Put some cornstarch onto the eggplant pieces and mix well. 6. Cut potato: Cut the potatoes into pieces. 7. Prepare spices: Take out and cut some scallion, ginger, and garlic for later use. 8. Prepare sauce: Take out a bowl. Add water, soy sauce, cornstarch, salt, sugar, vinegar, cooking wine into the bowl and mix them. 9. Boil potatoes: Boil some water and put the potatoes in. Take the potatoes out when the edges become transparent. 10. Fry vegetables: Add oil into the pot, heat the oil up and fry the peppers first and then the eggplants and then the potatoes. 11. Stir-fry: Add some oil into the pot, heat up and put the spices into the pot. Stir-fry a few times. Add the prepared sauce into the pot and then add all the vegetables. Stir-fry until the vegetables and the sauce are well mixed. 12. Transfer: Transfer the cooked dish from the pot into a plate.

Task4: Moo Shu Pork	Daily	1. Cut pork: Take out a piece of pork and cut into small slices. 2. Prepare pork: Put the pork into a bowl. Add some water and wash. Squeeze the pork and pour the water. Put the pork back and add oil, salt, and cooking wine. Mix well. 3. Prepare vegetables: Wash the necessary vegetables, use the pot to boil the vegetables. Take out for later use. 4. Prepare egg: Crack some eggs into a bowl, mix the eggs. 5. Boil vegetables: Boil some water in the pot. Add vegetables and continue to boil for a minute. 6. Fry eggs: Add oil into the pot and then fry the mixed egg. Put the fried egg scramble into a bowl. 7. Stir-fry vegetables: Heat the pot and add oil. After the oil gets heated, first add the prepared spices and then add the vegetables. Stir-fry the vegetables. 8. Stir-fry pork: Without taking the vegetables out of the pot, add pork into the pot, stir-fry all ingredients together. 9. Stir-fry egg: Without taking the ingredients out of the pot, add scrambled egg into the pot, stir-fry all ingredients together. 10. Transfer: Transfer the cooked dish from the pot into a plate.
Task5: Tomato dough drop soup	Daily	1. Prepare spices: Take out and cut some scallion and cumin for later use. 2. Prepare tomatoes: Take out tomatoes, wash and peel. 3. Cut tomatoes: Use a knife to cut the tomatoes first into slices and then into small pieces. 4. Prepare eggs: Crack eggs into a bowl, then stir until evenly mixed. 5. Fry eggs: Heat oil in a pan, add the evenly mixed egg mixture, and stir-fry, finally transfer the cooked scrambled eggs to a plate. 6. Stir-fry tomatoes: Put the chopped tomatoes into the pan, stir-fry them, and then add the scrambled eggs. 7. Soup-making: Add a large amount of clear water to the pot and bring it to a boil. 8. Prepare dough: Gradually add water to the flour while stirring until the flour forms dough. 9. Soup-making: Drop the flour dough into the boiling soup, while adding them, stir continuously. 10. Add seasoning: Add salt, pepper, and MSG (if desired) to the soup. 11. Transfer: Transfer the soup into a large bowl.
Task6: Solid Phase Peptide Synthesis	Chemical lab	1. Weighing: Use a balance to weigh the desired amount of amino acid powder (white powder). Put the powder into a test tube. 2. Reaction: Use a pipette to aspirate some SPPS resin (Transparent liquid) into the test tube. Shake the test tube and put the test tube onto the shaker machine. 3. Deprotection: Add the needed reagent into the tube to separate resin and peptide. 4. Suction Filteration: Take the test tube from the shaker machine, wash the peptide inside the tube, and suck the liquid into the vacuum tube. 5. Checking: Manual check and take necessary notes.
Task7: Total Protein Extraction	Medical lab	1. Preparation: Take out several test tubes, add the necessary amount of PBS reagent (transparent liquid). Take out the cells from the fridge, disinfect, and warm the cells. 2. Wash cells: Use a pipette to transfer the cells into test tubes. 3. Centrifuge: Balance the test tubes in the centrifuge and then start the centrifugation. 4. Reagent making: Prepare some petri dishes, mark each dish, and add the complete medium (pink liquid) into each dish. 5. Transfer cells: Take the cells out of the centrifuge, and check the cell state. Transfer the cells into the prepared petri dish. 6. Quantification: Use an electron microscope to check the cells and record the required information. 7. Other necessary steps: Repeat necessary steps, make necessary reagents, etc.

Task8: Cell subculture	Biology lab	1. Preparation: Prepare the cell, test tubes, reagents, and petri dishes. Mark accordingly. 2. Wash Cells: Use a pipette to aspirate PBS reagent, use PBS to wash the cells. 3. Digestion: Use a separate pipette to add pancreatic enzymes (pink liquid) into the cells, put the cells into an incubator and wait for 3 minutes. 4. Quantification: Use an electron microscope to check the cells and record the required information. 5. reagent making: Prepare some petri dishes, mark each dish, and add the complete medium (pink liquid) into each dish. Use the complete medium to wash the petri dish. 6. Centrifuge: Balance the test tubes in the centrifuge and then start the centrifugation. 7. Transfer cells: Take the cells out of the centrifuge, and check the cell state. Transfer the cells into the prepared petri dish. 8. Incubation: Put the cells with the petri dish into the incubator.

Table S10. The tasks in our `EgoExoLearn` with example procedures.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 7

[2] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023. 7

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021. 1

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4, 5, 6, 8

[5] Min-Hung Chen, Zsolt Kira, Ghassan AlRegib, Jaekwon Yoo, Ruxin Chen, and Jian Zheng. Temporal attentive alignment for large-scale video domain adaptation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019. 2, 3

[6] Min-Hung Chen, Baopu Li, Yingze Bao, Ghassan AlRegib, and Zsolt Kira. Action segmentation with joint self-supervised temporal domain adaptation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6

[7] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, 130:33–55, 2022. 2, 9

[8] Hazel Doughty, Dima Damen, and Walterio Mayol-Cuevas. Who's better? who's best? pairwise deep ranking for skill determination. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5

[9] Hazel Doughty, Walterio Mayol-Cuevas, and Dima Damen. The pros and cons: Rank-aware temporal attention for skill determination in long videos. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 4, 5

[10] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 6824–6835, 2021. 8, 9, 10

[11] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 6

[12] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 6202–6211, 2019. 8, 9, 10

[13] Kristen Grauman, Andrew Westbury, and Eugene Byrne et al. Ego4d: Around the World in 3,000 Hours of Egocentric Video. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 9

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6, 10

[15] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*, 2023. 7

[16] Zhenqiang Li, Yifei Huang, Minjie Cai, and Yoichi Sato. Manipulation-skill assessment from videos with spatial attention network. In *Proceedings of the International Conference on Computer Vision Workshops (ICCVW)*, 2019. 5

[17] 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. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1

[18] Mathew Monfort, Bowen Pan, Kandan Ramakrishnan, Alex Andonian, Barry A McNamara, Alex Lascelles, Quanfu Fan, Dan Gutfreund, Rogério Schmidt Feris, and Aude Oliva. Multi-moments in time: Learning and interpreting models for multi-action video understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):9434–9445, 2021. 10

[19] 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 *Proceedings of the International Conference on Machine Learning (ICML)*, 2021. 1, 2, 8, 9

[20] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5

[21] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 4, 5

[22] Xin Wang, Taein Kwon, Mahdi Rad, Bowen Pan, Ishani Chakraborty, Sean Andrist, Dan Bohus, Ashley Feniello, Bugra Tekin, Felipe Vieira Frujeri, Neel Joshi, and Marc Pollefeys. Holoassist: an egocentric human interaction dataset for interactive ai assistants in the real world. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 5, 9

[23] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3333–3343, 2022. 1

[24] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1