Supplementary Material

This section contains supplementary material to support the main paper text. The contents include:

- (§S1) Implementation details for three pre-trained egocentric models M^{τ} from Sec. 3.3.
- (§S2) Implementation details for Kinetics pre-training presented in Sec. 3.1.
- (§S3) Implementation details for fine-tuning on downstream egocentric datasets.
- (§S4) Additional results on EPIC-Kitchens, Charades-Ego and Something-Something v2 datasets.
- (§S5) Additional ablation studies, including ablations of the Interaction-map model, using M^{τ} as pre-trained models, appending features from M^{τ}, the impact of egocentric dataset scale on model performance, implicit pairing information in Ego-Scores and effect of using different backbones for Ego-Exo.
- (§S6) Additional qualitative results, including distribution of Ego-Score over Kinetics, additional qualitative examples and class-wise breakdown of improvements for auxiliary tasks M^{τ}.
- **Supplementary video.** A demonstration video shows animated version of video clips for the qualitative examples in §S6.

S1. Details: Pre-trained egocentric models M^{τ}

We provide additional implementation details for the task models used in Auxiliary egocentric tasks from Sec. 3.3.

Ego-Classifier M^{ego} for Ego-Score. We use a Slow-only model [19] with a ResNet-50 backbone as the ego-classifier M^{ego}. Then, we train M^{ego} on the Charades-Ego dataset [62] in which each instance is assigned with a binary label indicating if it is egocentric or exocentric. We take a Kinetics-pretrained model as initialization and train with 8 GPUs in 100 epochs. We adopt a cosine schedule for learning rate decaying with a base learning rate as 0.01 and the mini-batch size is 8 clips per GPU. To generate pseudo-labels for Kinetic videos, we sample $N = 2$ clips for each video and generate our Ego-Score using Eqn 1.

Object recognition model M^{obj} for Object-Score. We directly use an off-the-shelf object recognition model trained on ImageNet as M^{obj}. Specifically, we use a standard ResNet-152 network from Pytorch Hub[3]. For each Kinetics video, we sample $T = 64$ frames and generate Object-Score following Eqn 3.

[3]https://pytorch.org/hub/pytorch_vision_resnet/

Hand-object detector M^{int} for Interaction-Map. We adopt a pre-trained hand-object detector [60] to discover hand interaction regions. For the detected bounding-box for hands and interactive objects from Kinetics videos, we keep only high-scoring predictions and eliminate bounding boxes with confidence scores less than 0.5.

Note that all the three pre-trained egocentric models are either easy to access (off-the-shelf models M^{obj} and M^{int}) or easy to train (M^{ego}). Meanwhile, our auxiliary losses do not require the modification of the network, thus our model can be directly used as a drop-in replacement for downstream egocentric video tasks after pre-training.

S2. Details: Pre-training on Kinetics

We follow the training recipe in [19] when training on Kinetics, and use the same strategy for both Slow-only and SlowFast backbones and different implemented methods.

All the models are trained from scratch for 200 epochs. We adopt a synchronized SGD optimizer and train with 64 GPUs (8 8-GPU machines). The mini-batch size is 8 clips per GPU. The baseline learning rate is set as 0.8 with a cosine schedule for learning rate decaying. We use a scale jitter range of [256, 320] pixels for input training clips. We use momentum of 0.9 and weight decay of 10^{-4}.

S3. Details: Fine-tuning on Ego-datasets

Charades-Ego. During fine-tuning, we train methods using one machine with 8 GPUs on Charades-Ego. The initial base learning rate is set as 0.25 with a cosine schedule for learning rate decaying. We train the models for 60 epochs in total. Following common practice in [19], we uniformly sample 10 clips for inference. For each clip, we take 3 crops to cover the spatial dimensions. The final prediction scores are temporally max-pooled. All other settings are the same as those in Kinetics training.

EPIC-Kitchens. We use a multi-task model to jointly train verb and noun classification with 8 GPUs on EPIC-Kitchens [12]. The models are trained for 30 epochs with the base learning rate as 0.01. We use a step-wise decay of the learning rate by a factor of $10\times$ at epoch 20 and 25. During testing, we uniformly sample 10 clips from each video with 3 spatial crops per clip, and then average their predictions. All other settings are the same as those in Kinetics training.

For EPIC-Kitchens-100 [11], we take the same optimization strategy as EPIC-Kitchens [12], except training with 16 GPUs with a 0.02 base learning rate.

S4. Additional results

EPIC-Kitchens. We report results of an Ensemble of four Ego-Exo models on EPIC-Kitchens in Table S1. Specifi-

S1 (seen)		verbs		nouns		actions	
	Methods	top-1	top-5	top-1	top-5	top-1	top-5
w/ audio	Epic-Fusion [35]	64.75	90.70	46.03	71.34	34.80	56.65
	Epic-Fusion [35] (Ensemble)	**66.10**	**91.28**	**47.80**	**72.80**	**36.66**	**58.62**
w/o audio	SlowFast [19]	64.57	89.67	45.89	69.50	34.67	54.47
	Ego-Exo (Single)	65.97	90.32	47.99	70.72	37.09	56.32
	Ego-Exo (Ensemble)	**67.84**	**90.87**	**49.61**	**71.77**	**38.93**	**58.08**
S2 (unseen)							
w/ audio	Epic-Fusion [35]	52.69	79.93	27.86	53.87	19.06	36.54
	Epic-Fusion [35] (Ensemble)	**54.46**	**81.23**	**30.39**	**55.69**	**20.97**	**39.40**
w/o audio	SlowFast [19]	53.91	80.81	30.15	55.48	21.58	37.56
	Ego-Exo (Single)	55.34	81.46	31.72	58.25	22.81	40.18
	Ego-Exo (Ensemble)	**56.03**	**81.15**	**32.54**	**60.29**	**23.22**	**40.97**

Table S1: **Ego-Exo Ensemble results on EPIC-Kitchens (test set).** Our method outperforms all methods in both seen and unseen settings.

	Overall						Unseen Participants			Tail Classes		
	top-1			top-5			top-1			top-1		
Methods	verbs	noun	action	verb	noun	action	verb	noun	action	verb	noun	action
Leaderboard1†	66.63	48.98	38.59	89.94	73.84	58.62	60.56	43.58	31.63	29.80	15.02	12.97
Leaderboard2†	65.32	47.80	37.39	89.16	73.95	57.89	59.68	42.51	30.61	30.03	16.96	13.45
SlowFast [19]	63.89	49.66	37.42	88.71	74.99	58.17	57.37	44.31	29.71	33.57	22.57	16.55
Ego-Exo (Single)	66.07	51.51	39.98	89.39	76.31	60.68	59.83	45.50	32.63	33.92	22.91	16.96
Ego-Exo (Ensemble)	**67.13**	**53.94**	**42.08**	**90.07**	**77.83**	**62.69**	**61.05**	**49.15**	**35.18**	**34.73**	**24.92**	**18.19**

Table S2: **Ego-Exo Ensemble results on EPIC-Kitchens-100 action recognition test set.** Leaderboard1† and Leaderboard2† are the top two methods on the leaderboard at the time of submission. Our method is best across all categories.

cally, the Ensemble model includes Ego-Exo and Ego-Exo* with ResNet-50 and ResNet-101 backbones. As shown in Table S1, the Ensemble Ego-Exo further improves the performance in all categories, and consistently outperforms the Ensemble model of Epic-Fusion.

Table S2 shows the Ensemble results on EPIC-Kitchens-100. The Ensemble model of Ego-Exo outperforms the current best model on the leaderboard[4] in all categories at the time of submission, especially on noun and action classes with +5% and +3.5% improvements on Overall Top-1 metric. Note that even without Ensemble, our single model already ranks the first on leaderboard and achieves better results than the best leaderboard model in most categories.

Charades-Ego. We only train SlowFast and Ego-Exo methods on the egocentric videos from Charades-Ego in Table 4 of Sec 4. As Charades-Ego also provides third-person videos, we further jointly train first-person and third-person video classification during fine-tuning on Charades-Ego. In this setting, SlowFast and Ego-Exo achieve 25.06 and 28.32 mAP, respectively, with a ResNet-50 backbone. Hence our model further improves over that multi-task setting.

Something-Something V2 (SSv2). SSv2 [26] is a non-ego dataset with videos containing human object interactions. We further apply our Ego-Exo method on this dataset and find our *Ego-Exo* method improves over baseline *Third-only* (59.49% → 60.41% in accuracy). Though our goal remains to address egocentric video, it does seem our method can even have impact beyond it and works for general interaction scenario.

S5. Additional Ablation studies

Effect of hand-map and object-map in Interaction-Map. We ablate the Interaction-Map task M^{int} by only using the Hand-Map and Object-Map scores in Eqn 6. As shown in Table S3, using *Hand-Map* or *Object-Map* alone consistently improves over the baseline (*Third-only*) while combining them (*Interaction-Map*) achieves the best results overall.

[4] https://competitions.codalab.org/competitions/25923#results

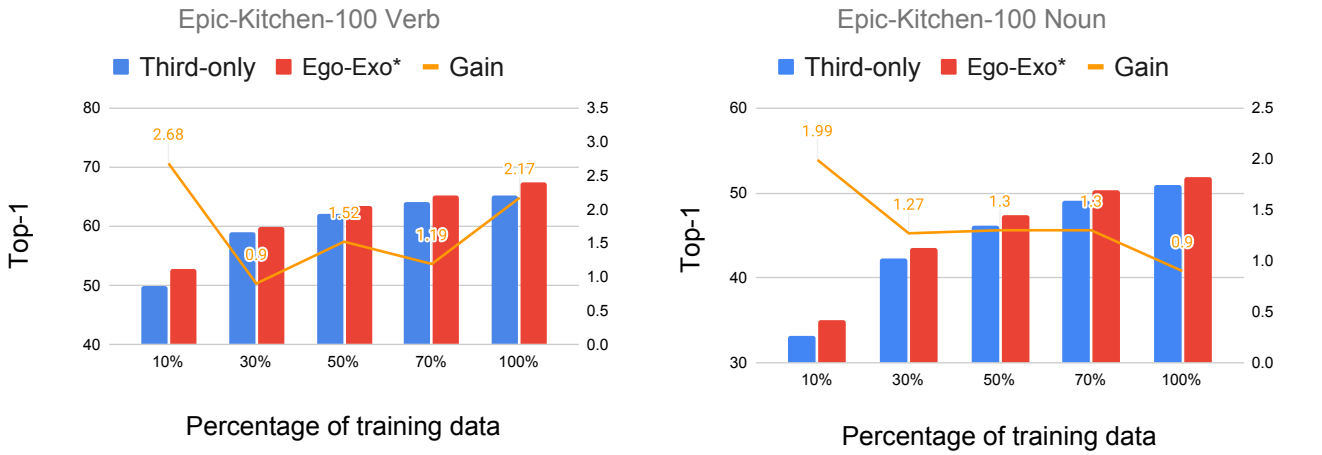

Figure S1: **Performance on EPIC-Kitchens-100 using different percentages of training videos.** Our method consistently improves over the baseline *Third-only* when using different percentages of training videos, with large improvements in very limited data settings (10% of data). Orange curve shows absolute improvement.

Methods	C-Ego mAP	EPIC verbs top-1	EPIC verbs top-5	EPIC nouns top-1	EPIC nouns top-5
Third-only	24.69	61.19	87.49	46.18	69.72
Hand-Map	25.28	61.35	88.02	47.33	**70.03**
Object-Map	**26.15**	61.32	87.66	46.65	69.56
Interaction-Map	25.91	**62.55**	**88.50**	**47.71**	69.62

Table S3: **Ablation study on Interaction-Map task.** Combining Hand-Map and Object-map (Interaction-Map) achieves better results overall. Values are averaged over 3 runs.

Methods		C-Ego mAP	EPIC verbs top-1	EPIC verbs top-5	EPIC nouns top-1	EPIC nouns top-5
Third-only		24.69	61.19	87.49	46.18	69.72
pre-trained	M^{ego}	23.29	61.95	87.07	46.09	68.88
	M^{obj}	22.17	57.34	86.63	45.44	68.28
append	M^{ego}	24.92	60.87	87.38	46.40	69.97
	M^{obj}	24.73	61.08	87.35	45.80	68.97
	M^{int}	24.68	61.85	87.39	46.89	69.98
	3 aux	24.75	61.05	87.45	46.41	70.02
distilled	M^{ego}	25.01	62.22	87.78	46.26	68.76
	M^{obj}	25.49	61.65	87.57	46.27	69.52
	M^{int}	25.91	62.55	**88.50**	47.71	69.62
	Ego-Exo	**26.23**	**62.83**	87.63	**48.15**	**70.28**

Table S4: **Comparison with fine-tuning or appending features from auxiliary task models.** Our distillation methods outperform the other two schemes.

Effect of taking M^{τ} as pre-trained model. In section 3.3, we introduce several auxiliary egocentric tasks M^{τ} and distill information from them into the video model using auxiliary losses in our Ego-Exo pre-training framework. An alternative way to exploit these signals is to directly use these models (M^{τ}) from auxiliary egocentric tasks M^{τ} as our pre-trained models, then fine-tune them on the egocentric datasets. Specifically, we take the ego-classifier M^{ego} and object recognition model M^{obj} as pre-trained models. We do not include hand-object detector M^{int} here as the detection backbone is not compatible with the video backbone.

As shown in Table S4, though auxiliary task models capture specific egocentric properties, directly use them as pre-trained models is still insufficient. Our methods successfully embed the egocentric information from these auxiliary tasks into the video model through distillation losses, and still enjoy the strong representations learned from the large-scale third-person dataset.

Effect of appending embeddings from M^{τ}. Another alternative way to exploit these information from auxiliary tasks (M^{τ}) is to directly use the extracted features on egocentric datasets using M^{τ}. Specifically, we extract the embeddings after the global pooling layer of the three auxiliary models and concatenate them with the ego models during fine-tuning. The results are shown in the 'append' rows of Table S4. It indicates these baselines are less effective than the proposed distillation scheme in our Ego-Exo method.

Impact of the scale of egocentric datasets. We study our model performance under varying scales of egocentric video supervision by using different percentages of videos in EPIC-Kitchens-100 [11]. Fig S1 shows that our model consistently outperforms the baseline *Third-only* at all dataset scales, though both models perform worse with less egocentric videos during fine-tuning. When using only

Methods	C-Ego	EPIC verbs		EPIC nouns	
	mAP	top-1	top-5	top-1	top-5
Third-only	24.69	61.19	87.49	46.18	69.72
Ego-Score (no-pair)	24.88	62.22	87.36	46.16	68.10
Ego-Score	25.01	62.22	87.78	46.26	68.76
Ego-Exo (no-pair)	26.29	62.72	87.61	48.07	70.31
Ego-Exo	26.23	62.83	87.63	48.15	70.28

Table S5: **Ablation study on the implicit pairing information**. Methods achieves similar results with or without implicit pairing information. Note that all the methods do not use explicit pairing information from Charades-Ego.

Backbones	C-Ego	EPIC verbs		EPIC nouns	
	mAP	top-1	top-5	top-1	top-5
Third-only I3D	25.07	59.42	**87.82**	47.16	69.39
Ego-Exo I3D	**26.61**	**61.51**	87.28	**47.87**	**69.60**
Third-only TSM	25.66	60.17	**87.75**	46.58	70.13
Ego-Exo TSM	**26.22**	**61.80**	87.60	**47.71**	**70.30**

Table S6: **Results of using I3D and TSM backbones.** *Ego-Exo* consistently outperforms *Third-only*.

10% of training data, our method improves over the *Third-only* by +2.68% and 1.99% on verb and noun tasks.

Implicit pairing information in Ego-Scores. We train the Ego-Classifier M^{ego} for Ego-Scores on Charades-Ego [62]. During training, we only use the binary label to indicate the instance is egocentric or not and don't utilize any pairing information. However, Ego-Score might still contains some implicit pairing information. Here, we conduct a ablation study by only taking one view (either ego or non-ego instance) for each pair in Charades-Ego when training M^{Ego}. Table S5 shows that methods without any implicit pairing information achieves similar performance. This demonstrates that the implicit pairing information is not critical for our Ego-Exo method.

Ego-Exo with different backbone networks. Besides using Slow and SlowFast backbones in Sec 4, Table S6 further compares the results using I3D and TSM as the backbone structure. *Ego-Exo* achieves better results over the baseline *Third-only* on both Charades-Ego and Epic-Kitchen datasets. It indicates the versatility of our idea wrt the chosen backbone.

S6. Additional qualitative results

Distribution of Ego-Score over Kinetics. As mentioned in Sec. 3.3, though Kinetics videos are predominantly captured in the third-person perspective, the Ego-Score generated by the pretrained classifier M^{ego} is not trivially low for all video instances. Fig S2 plots the distribution of values this score takes. While a majority of instances have very low scores (not *ego-like*), a large number of instances prominently feature egocentric signals (image inset, right) and have higher scores.

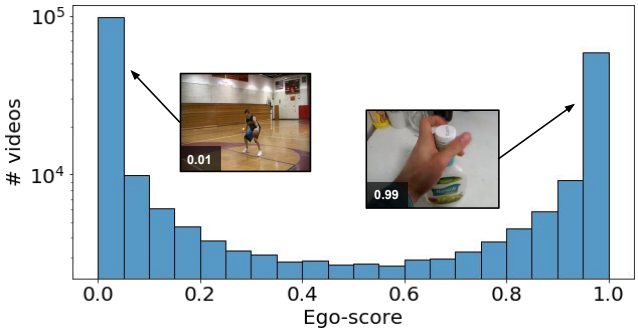

Figure S2: **Distribution of Ego-Scores across Kinetics instances.** Despite being from the third-person perspective, videos in Kinetics display egocentric properties captured by the Ego-Score.

Figure S3: **Charades-Ego classes improved by each egocentric signal.** Each circle contains the classes improved over *Third-only* by a particular ablated model from Table 2.

Class-wise breakdown of improvements from each egocentric task M^{τ}. We present a qualitative result corresponding to the ablation experiment in Table 2 in the main paper. Fig. S3 shows a venn diagram where each circle contains classes from Charades-Ego that a particular ablated model in Table 2 improves upon, over the baseline model. For example, the red circle is a model with only Ego-score

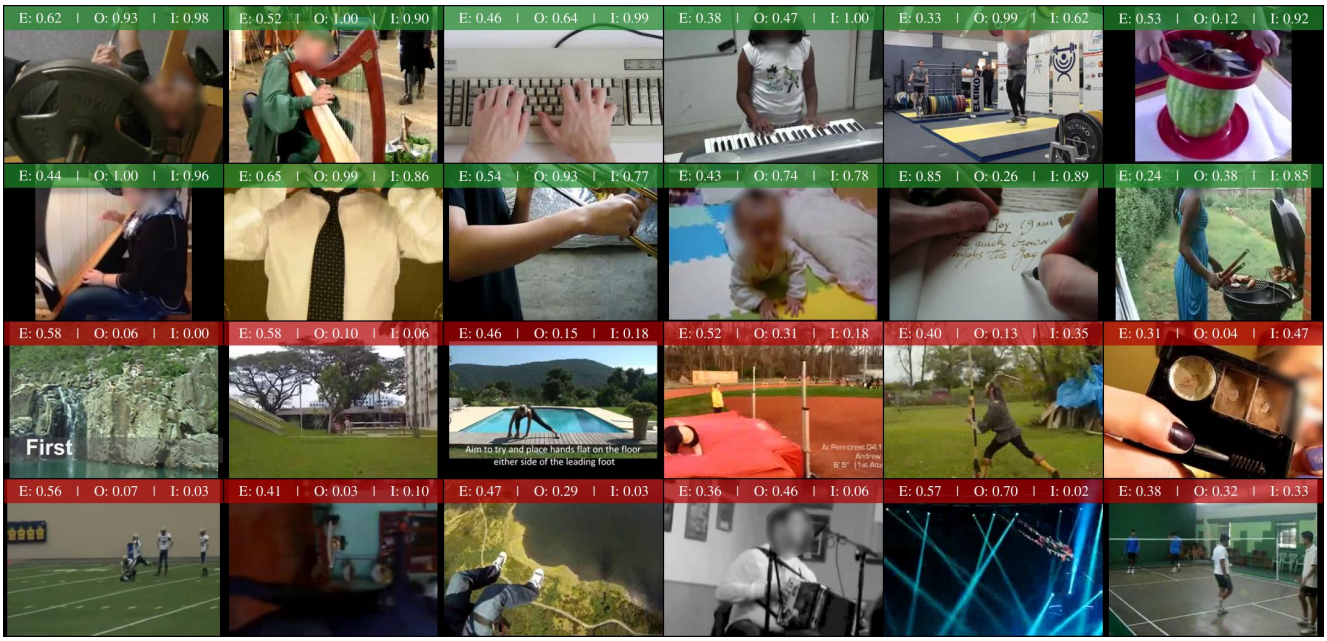

Figure S4: **Additional Kinetics instances sorted by Ego-Exo scores.** Top two rows: Instances that prominently feature hands/objects/egocentric-like motion patterns. Bottom two rows: Instances that feature static scenes devoid of egocentric-like activity.

(row 2, Table 2). The overlapping regions between two circles contain classes that are improved by both corresponding ablated models. Note that the three ablated models all contains some classes which are only improved by one particular ablated model, which suggests that three auxiliary tasks capture different egocentric properties.

Additional qualitative examples In Fig. S4, we show additional examples of instances from Kinetics, sorted by the scores generated by our pre-trained egocentric models M^{τ} to supplement Fig. 5 in the main paper. The top two rows contain instances with high scores (more ego-like, more prominently features objects, and more hand-object interactions), while the bottom two rows feature instances with low scores. Note that the instances shown are frames from the corresponding video clips. Typically, video clips with more ego-like viewpoint and motions usually have higher Ego-Score. Please see the animated version of this figure in the supplementary video.