HiERO: understanding the hierarchy of human behavior enhances reasoning on egocentric videos

Supplementary Material

Sec. A provides further details on the datasets and tasks used in this work. Sec. B presents additional implementation details and a discussion of some key design choices behind HiERO. Sec. C evaluates different clustering algorithms for HiERO on the Step Grounding and Procedure Learning tasks. Sec. D discusses the unsupervised emergence of procedural steps in HiERO's features space. Sec. E analyzes the impact of using a more informative backbone in the previous SOTA on the EgoProceL benchmark. Finally, Sec. F presents additional qualitative results on the Step Localization task.

A. Dataset and task details

A.1. Ego4D

Ego4D is a large scale egocentric vision dataset with 3670 hours of daily-life activities captured from 931 subjects around the world. Videos are annotated with fine-grained textual descriptions of the activities performed by the camera wearer or other participants in the scene, *e.g.*, *"#C C stirs food in a frying pan with a spoon in his right hand"*, and with task-specific annotations on a subset of the videos for a wide range of tasks, including episodic memory, spatial and temporal grounding of the interactions, forecasting, etc. We focus our analysis on two benchmark, namely EgoMCQ and EgoNLQ.

EgoMCQ. EgoMCQ is a development benchmark introduced with EgoVLP [29] to validate the quality of video-language pretraining models. It features $39k$ multiple-choice questions generated from Ego4D annotations. Given a textual query and five candidate video clips, the task is to identify the correct clip. Candidates may belong to the same video (*intra-video*) or from different videos (*inter-video*). Performance is evaluated in terms of accuracy.

EgoNLQ. EgoNLQ is a temporal grounding task that requires multi-modal video and language reasoning. Given a textual query from a set of predefined templates, the goal is to identify the temporal boundaries (start and end timestamps) of the video segment that answers the query. The benchmark includes $13.6k$ / $4.5k$ / $4.4k$ queries in the train, validation and test splits respectively. We follow previous works in video-language pre-training [29, 39, 57] and evaluate HiERO on this task using VSLNet [56] as grounding head, using the same hyper-parameter tuning recipe as EgoVLP [29] and reporting results on the validation set. As for EgoNLQ, performance is evaluated in terms of Top-1 and Top-5 Recall at different Intersection over Union (0.3 and 0.5) between the predicted and the ground truth segments.

A.2. Goal-Step

Goal-Step [48] extends the Ego4D dataset with annotations of hierarchical activity labels, identifying goals, steps and substeps in procedural activities. It provides dense annotations for $48k$ procedural step segments (480 hours), from a taxonomy of 501 labels. We evaluate HiERO on the Step Grounding and Step Localization tasks.

Step Grounding. Step Grounding is a temporal grounding task, in which the goal is to recognize the temporal boundaries of a procedural step given its description in natural language. For supervised experiments we use the same architecture of the baseline (VSLNet [56]) with the same hyper-parameters and report performance as the average of 8 runs. When using EgoVLP features we extend the number of samples in the input sequence from 128 to 256. Performance is evaluated in terms of Top-1 and Top-5 Recall at different Intersection over Union (0.3 and 0.5) between the predicted and the ground truth segments.

Step Localization. Step Localization is more closely related to action segmentation. Given a long video, the goal is to find all the procedure steps in the video with their corresponding start/end time and label according to the Goal-Step taxonomy. Models are trained and evaluated on steps and substeps without distinctions. The supervised models use ActionFormer [55] as localization head, with base learning rate of 2e-4 and training for 32 epochs with linear warm-up for 16 epochs. Performance is evaluated in terms of mAP at different Intersection over Union (IoU) thresholds between the predicted and the ground truth segments.

A.3. EgoProceL

EgoProceL [4] collects multiple egocentric vision datasets focusing on procedural tasks that require multiple steps, *e.g.*, *Preparing a salad* or *Assemblying a PC*: MECCANO [41], Epic-Tents [21], CMU-MMAC [10], EGTEA [28] and PC Assembly/Disassembly [4]. Table 6 reports the number of videos and key-steps in each task of the dataset. Annotations assign each video frame to a specific key-step of the corresponding task. We evaluate HiERO on the **Procedure Learning** task, following

Task	Videos Count	Key-steps Count
PC Assembly [4]	14	9
PC Disassembly [4]	15	9
MECCANO [41]	17	17
Epic-Tents [21]	29	12
CMU-MMAC [10]		
Brownie	34	9
Eggs	33	8
Pepperoni Pizza	33	5
Salad	34	9
Sandwich	31	4
EGTEA+ [28]		
Bacon and Eggs	16	11
Cheese Burger	10	10
Continental Breakfast	12	10
Greek Salad	10	4
Pasta Salad	19	8
Hot Box Pizza	6	8
Turkey Sandwich	13	6

Table 6. **Number of videos and key-steps in EgoProceL [4].**

the same evaluation protocol of previous works [4, 5, 8]. Specifically, we compute framewise step assignment and evaluate the F1-score and Intersection over Union (IoU) between the predicted steps and the ground truth labels for each step separately. The F1-score is computed as the harmonic mean of precision and recall. Precision is the proportion of correctly identified key-step frames out of all frames predicted to be key-steps, while recall is the proportion of correctly identified key-step frames out of the total number of actual key-step frames. Predictions and ground truth labels are matched using the Hungarian algorithm, following previous works [4, 8].

B. Additional implementation details

HiERO follows an encoder-decoder architecture with three stages, each comprising three layers of TDGC [38], with hidden feature size 768 and the threshold for temporal graph connectivity d is set to 1. Input features are first projected to size 768 using a linear layer. For $\mathcal{L}_{vna}$ and $\mathcal{L}_{ft}$, we set the temperature parameter to $\tau = 0.05$. When evaluating HiERO on EgoMCQ, we assume that only a single functional thread is present in the input video, given the short duration of the clip, and disable the functional threads clustering of the decoder.

Strategy	Trainable Params	EgoMCQ Inter	EgoMCQ Intra
Frozen	20.10 M	84.2	46.0
LoRa [18]	20.99 M	88.2	49.7
Full Fine-Tuning	86.47 M	**90.3**	**53.3**

Table 7. **Comparison of different fine-tuning strategies for the text-encoder of HiERO**, using Omnivore features and measuring performance on EgoMCQ. Full fine-tuning significantly improves accuracy.

Text-encoder fine-tuning. EgoVLP [29] and LaViLa [57] were trained for video-text alignment. Therefore, when building HiERO on these backbones we reuse their respective text encoders, with no additional training. Instead, Omnivore was not trained for video-text alignment and does not have a text encoder. In this case, we bootstrap the text encoder of HiERO from a pretrained DistillBERT [43] and fine-tune it during the the training process. We experiment different strategies to fine-tune the text encoder, using LoRa [18] to reduce the number of trainable parameters or fully updating the text encoder, as shown in Table 7. While LoRa provides a significant improvement compared to the frozen text encoder, the gap with the full fine-tuning is consistent. Remarkably, with little computational overhead (training lasts less than 20 GPU hours), HiERO reaches performance close to that of EgoVLP, despite not being trained end-to-end on Ego4D.

Δ	EgoVLP [29] Inter	EgoVLP [29] Intra	LaViLa [57] Inter	LaViLa [57] Intra	HiERO (EgoVLP) Inter	HiERO (EgoVLP) Intra
N/A (paper)	90.6	57.2	94.5	63.1	–	–
0	90.7	**53.4**	93.9	**57.9**	89.0	52.4
1	**91.0**	52.5	**94.1**	56.7	90.9	57.4
2	90.8	48.7	93.6	52.5	91.3	58.8
4	89.9	42.2	93.1	44.8	**91.8**	**59.5**

Table 8. **Impact of the additional context window on EgoMCQ Accuracy (%).** The first row refers to the original results, as reported in their respective papers.

Impact of the context window in EgoMCQ. HiERO is built on dense pre-extracted features from fixed size segments (16 frames) of the video, using a pre-trained backbone, *e.g.*, EgoVLP [29] or LaViLa [57]. Each segment is mapped to a node of the input graph $\mathcal{G}$. We adapt the evaluation process for HiERO to work with pre-extracted features. Specifically, when evaluating HiERO on benchmarks that require a fixed size input, *e.g.*, EgoMCQ, the nodes correspond to all video segments that fall between the start t_s and end timestamps t_e of the input. Since clips in EgoMCQ are very short ($0.84s$ on average), we slightly extend the clip segment by a context window Δ to provide additional temporal context and ensure and the resulting graph has a reasonable number of nodes for processing. We adapt EgoVLP and LaViLa to our setting, *i.e.*, using dense features extracted from video segments with additional temporal context, and evaluate the impact of this additional temporal context on EgoVLP and LaViLa in Table 8, showing that this additional context does not trivially translate to better performance on this benchmark. In contrast, HiERO is trained to exploit such additional temporal context and achieves best performance when used in combination with a larger input window ($\Delta = 4$). At the same time, HiERO is quite robust even to shorter context windows.

α	β	EgoMCQ	
		Inter	Intra
1	*all*	91.8	**59.5**
1	4	91.8	57.4
1	16	**92.0**	58.5
2	*all*	91.5	**59.5**
2	4	91.5	56.5
2	16	91.9	58.2

Table 9. **Ablation on the size of the video-narrations alignment window.** For β, *all* means that all narrations from the same video that are not part of the positives set are considered as negatives.

Video-Narrations alignment window. We evaluate in Table 9, different choices for the α and β parameters that control the size of the alignment window in $\mathcal{L}_{vna}$. α controls the window size for positive samples, with higher values resulting in narrower windows. β controls the window for sampling negatives narrations from the same video. Higher values indicate larger windows, with *all* meaning that all narrations from the videos are taken as negative, except the ones that fall inside the positives window. The α parameter has little impact on both *inter* and *intra* accuracy. The β parameter has a more noticeable impact on performance, with best results when all intra-video narrations are used as negatives.

B.1. Additional details on the Cut&Match module

The Cut&Match module updates the connectivity of a video graph $\mathcal{G}$ in the HiERO architecture to connect regions, *i.e.*, video segments, that may be temporally distant but encode functionally related actions. This is achieved by grouping the graph nodes into K different partitions based on features cosine similarity using spectral clustering. As a result, the input graph $\mathcal{G}$ is partitioned into K sub-graphs $\{\tilde{\mathcal{G}}^{l+1}_{d,1}, \dots, \tilde{\mathcal{G}}^{l+1}_{d,K}\}$. Temporal reasoning is implemented on each sub-graph separately and nodes are then mapped back to the original graph.

Approximated graph partitioning. To efficiently implement the graph partitioning step on a batch of graphs, we approximate node partitioning by uniformly sub-sampling each graph to a fixed number of nodes based on the node timestamps. This allows to effectively batch all the operations involved in the graph partitioning step, *i.e.*, eigendecomposition of the Laplacian matrix and clustering, on all the graphs in the batch, regardless of their number of nodes. Spectral clustering is applied on the sub-sampled graphs and the cluster assignments are propagated to the original graph: each node in the original graph is assigned the label of the temporally closest node in the subsampled graph.

B.2. Zero-shot procedural tasks implementation

HiERO can address several procedural tasks in zero-shot by framing them as a graph clustering problem. We take graphs from different depths of the architecture depending on the task. For tasks that require video-language matching, such as step grounding or localization, we take the output of the last layer as the other layers are not language aligned. For tasks where this constraint is not present, *e.g.*, procedure learning on EgoProceL, we use features from deeper layers. Clustering is computed using the Spectral Clustering implementation from `scikit-learn`.

Features	Algorithm	mIoU@0.3		mIoU@0.5	
		R@1	R@5	R@1	R@5
EgoVLP	KMeans (L2)	10.37	24.65	6.85	16.46
EgoVLP	KMeans (Cos.)	8.97	23.21	5.91	15.15
EgoVLP	Spectral	10.73	24.70	7.38	16.53
Ours (EgoVLP)	KMeans (L2)	9.87	24.21	6.46	15.71
Ours (EgoVLP)	KMeans (Cos.)	10.35	24.85	6.93	16.27
Ours (EgoVLP)	Spectral	**11.57**	**27.41**	**7.87**	**18.70**

Table 10. **Impact of different clustering algorithms on the Step-Grounding task on Ego4D Goal-Step [48].** We evaluate the baselines and HiERO using KMeans and Spectral Clustering.

B.3. Features extraction with HiERO

On the Ego4D [17] dataset, we utilize the official `omnivore_video_swinl` features and extract dense features from 16-frame windows with a stride of 16 frames using the EgoVLP [29] and LAVILA [57] backbones. We follow the same procedure to extract features for the datasets in the EgoProceL [4] benchmark. When using HiERO as a features extractor, *e.g.*, to train VSLNet [56] for the Step Grounding task, we take features from the output layer of the decoder. HiERO's features have size 768 and maintain the same temporal granularity of the input features.

C. Comparison between clustering algorithms

Our approach builds a similarity graph from the video segments and discovers functional threads as strongly connected regions of the graph. In this context, spectral clustering groups segments and actions that may not be close in terms of euclidean or cosine distance but are linked through similar actions, forming a strongly connected region of the graph. We show the effectiveness of this design choice in Table 10 on the Step Grounding task from Goal-Step [48], comparing Spectral Clustering with KMeans using euclidean and cosine distances between the node embeddings. On the EgoVLP baseline, the two algorithms have similar performance. Similarly, we evaluate different clustering algorithms on EgoProceL in Table 11.

D. Procedure step emergence in HiERO

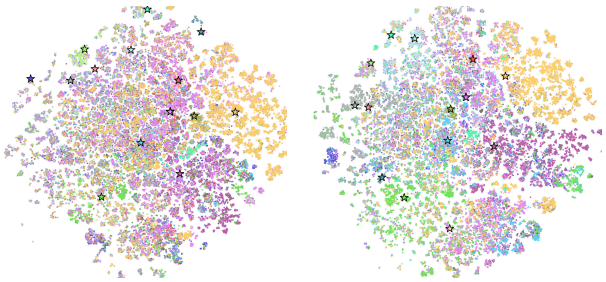
We evaluate the emergence of high-level *functional threads* in HiERO by analyzing the distribution of the textual embeddings for narrations and key-step labels from Goal-Step. For each ground truth (Fig. 5a) or zero-shot step predic-

Method	Algorithm	Average		CMU-MMAC [10]		EGTEA [28]		MECCANO [41]		EPIC-Tents [21]		PC Ass. [4]		PC Disass. [4]	
		F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU
Omnivore	K-Means	38.4	20.8	38.9	22.1	36.1	17.0	38.4	20.2	42.0	22.8	34.9	20.2	39.9	22.7
Omnivore	Spectral	39.1	22.0	44.7	26.8	37.1	19.2	36.0	19.0	40.8	21.9	35.7	21.5	40.3	23.5
EgoVLP	KMeans	40.6	22.0	46.6	28.2	37.3	17.3	32.9	16.1	40.1	20.9	39.0	21.5	47.3	28.1
EgoVLP	Spectral	40.0	21.9	49.2	31.0	36.6	18.3	33.1	16.1	37.4	19.2	38.2	20.8	45.4	25.6
Ours (Omnivore)	K-Means	43.7	24.2	46.9	27.3	38.6	18.4	**43.9**	**24.4**	45.2	**25.1**	43.4	23.7	44.0	26.1
Ours (Omnivore)	Spectral	44.0	24.5	47.2	27.7	39.7	**19.9**	41.6	22.1	**45.3**	24.3	43.7	25.1	46.3	27.9
Ours (EgoVLP)	K-Means	44.2	24.7	50.2	30.5	**40.4**	19.8	39.5	20.4	41.8	22.2	44.3	24.9	48.9	30.3
Ours (EgoVLP)	Spectral	**44.5**	**25.3**	**53.5**	**34.0**	39.7	19.6	39.8	20.3	39.0	20.3	**44.9**	**25.6**	**49.9**	**32.1**

Table 11. **Comparison of different clustering strategies on Omnivore and EgoVLP features [4].**

Method	Zero-Shot		Linear Probing	
	Top-1	Top-5	Top-1	Top-5
EgoVLP	10.11	29.47	25.22	53.08
Ours (EgoVLP)	**12.03**	**32.28**	**30.22**	**58.96**

Table 12. **Key-step classification accuracy on Goal-Step [48]**, using an oracle for *step* and *substep* detection. Steps and substeps are more easily recognizable in the HiERO feature space, despite no specific supervision.



(a) Ground truth steps (b) Predicted candidate steps

Figure 5. **Features distribution of narrations and procedural steps in Goal-Step [48].** Dots and stars represent the textual embeddings of the narrations and key-step labels, respectively, while the colors indicate the step to which the narrations are assigned.

tion (Fig. 5b), we collect all the narrations within the corresponding temporal window. Our results show that HiERO generates candidate steps where narrations are more tightly associated with the predicted key-step and form more distinct clusters, suggesting that narrations within the same step are semantically closer, irrespective of the granularity of the steps defined in the taxonomy. To show that HiERO features are more aligned with the key-step taxonomy despite no specific supervision, we train a linear probe on its features to predict the key-step label given the corresponding trimmed video segment (Table 12). Compared to EgoVLP, HiERO improves noticeably the alignment between the visual features and the key-steps taxonomy (+7.02% top-1 accuracy), showing the steps and substeps are more easily recognizable in the HiERO's feature space.

	Context (Stride)	CMU		MEC.		PC Ass.		PC Dis.		Avg.	
		F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU
–	2 (15)	36.5	18.8	39.2	20.2	33.7	17.9	32.2	16.9	35.4	18.5
OV	2 (15)	35.4	18.7	35.1	17.5	22.8	12.0	32.8	18.2	31.5	16.6
OV	4 (1)	31.6	20.1	36.9	18.3	33.0	18.8	31.0	16.4	33.1	18.4
OV†	4 (1)	31.6	17.5	33.3	17.8	32.0	17.4	34.9	19.0	32.9	17.9

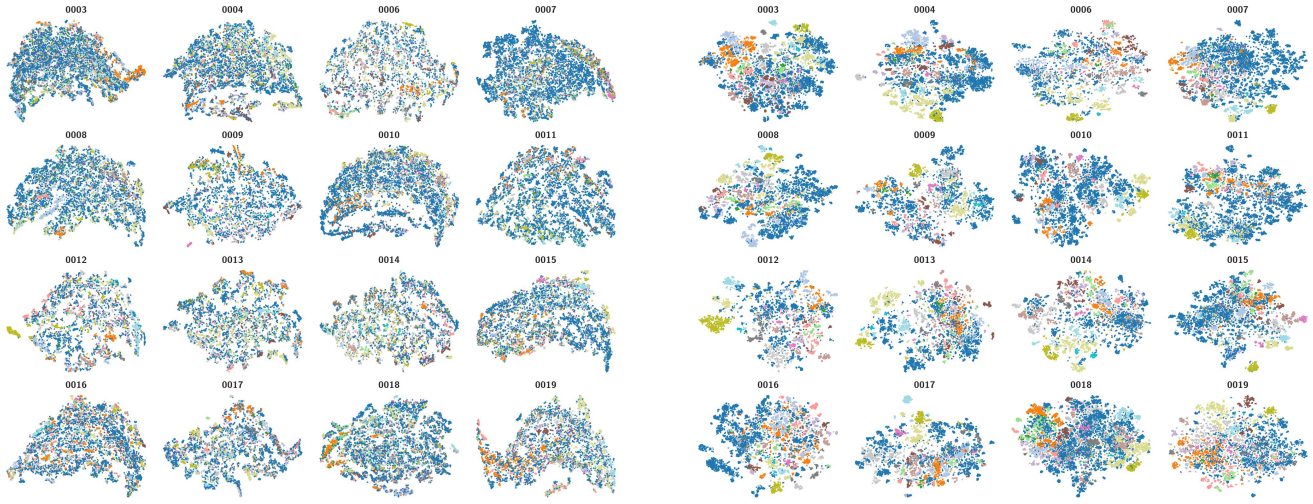
Table 13. **OPEL [8] with Omnivore backbone**, comparing different temporal context windows. OV: Omnivore backbone. OV†: frozen Omnivore backbone.

E. OPEL with Omnivore backbone

The Omnivore baseline significantly outperforms the previous SOTA on EgoProceL. We suggest that two main factors could explain the performance gap: (i) the different backbone and pre-training strategies used by OPEL (ResNet-50) and Omnivore, and (ii) different temporal contexts used for feature extraction. We replace the ResNet-50 backbone in OPEL with Omnivore, varying the temporal context and stride used for features extraction (Table 13). The two backbones show comparable performance, with an improvement observed as the temporal context increases. We were unable to evaluate larger context windows due to memory overflows in the training process. In addition, we show in Fig. 6 the features distribution of Omnivore against OPEL. Despite not being trained on MECCANO, Omnivore features exhibit quite clear clusters corresponding to the ground truth step labels. We argue that this behavior is the result of Omnivore being trained for action recognition on Kinetics-400.

F. Additional visualizations

Fig. 7 shows additional qualitative results on the Step Localization task, comparing our approach with EgoVLP [29]. We observe that most failure cases are associated to mismatches between the temporal granularity of the predictions and the ground truth, or to confusion between semantically similar steps or sub-steps.



(a) OPEL (Res-Net 50) Features

(b) Omnivore Features

Figure 6. **Features distribution of Omnivore and OPEL on MECCANO [41]**, with dots representing different video segments, and colors encoding the ground truth step labels. Despite not being trained on MECCANO, Omnivore features show a quite distinct separation between segments of the same action (same color).

References

[1] Triantafyllos Afouras, Effrosyni Mavroudi, Tushar Nagarajan, Huiyu Wang, and Lorenzo Torresani. Ht-step: Aligning instructional articles with how-to videos. In *NeurIPS*, 2023. 2

[2] Kumar Ashutosh, Rohit Girdhar, Lorenzo Torresani, and Kristen Grauman. Hiervl: Learning hierarchical video-language embeddings. In *CVPR*, 2023. 1, 2, 6

[3] Kumar Ashutosh, Santhosh Kumar Ramakrishnan, Triantafyllos Afouras, and Kristen Grauman. Video-mined task graphs for keystep recognition in instructional videos. *NeurIPS*, 2023. 2

[4] Siddhant Bansal, Chetan Arora, and CV Jawahar. My view is the best view: Procedure learning from egocentric videos. In *ECCV*, 2022. 1, 2, 6, 7, 8, 10, 11, 12, 13

[5] Siddhant Bansal, Chetan Arora, and CV Jawahar. United we stand, divided we fall: Unitygraph for unsupervised procedure learning from videos. In *WACV*, 2024. 2, 7, 11

[6] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *ICCV*, 2017. 8

[7] Matthew M Botvinick, Yael Niv, and Andew G Barto. Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective. *Cognition*, 2009. 1

[8] Sayeed Shafayet Chowdhury, Soumyadeep Chandra, and Kaushik Roy. Opel: Optimal transport guided procedure learning. In *NeurIPS*, 2024. 1, 2, 7, 11, 13

[9] Richard P Cooper and Tim Shallice. Hierarchical schemas and goals in the control of sequential behavior. *Psychological Review*, 2006. 1

[10] Fernando De la Torre, Jessica Hodgins, J Montano, S Valcarcel, R Forcada, and J Macey. Carnegie mellon university multimodal activity (cmu-mmac) database, 2008. 7, 10, 11, 13

[11] Nikita Dvornik, Isma Hadji, Ran Zhang, Konstantinos G Derpanis, Richard P Wildes, and Allan D Jepson. Stepformer: Self-supervised step discovery and localization in instructional videos. In *CVPR*, 2023. 2

[12] Ehsan Elhamifar and Dat Huynh. Self-supervised multi-task procedure learning from instructional videos. In *ECCV*, 2020. 2

[13] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 6

[14] Hongyang Gao and Shuiwang Ji. Graph u-nets. In *ICML*, 2019. 2, 4

[15] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A single model for many visual modalities. In *CVPR*, 2022. 1, 6, 8

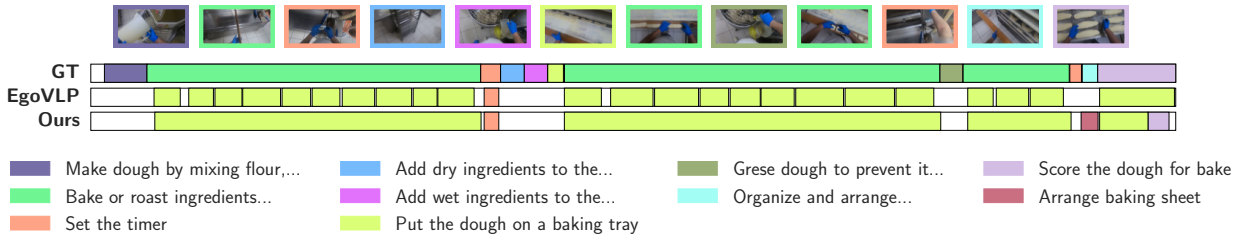
[16] Gabriele Goletto, Tushar Nagarajan, Giuseppe Averta, and Dima Damen. Amego: Active memory from long egocentric videos. In *ECCV*, 2024. 2

[17] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022. 3, 6, 12

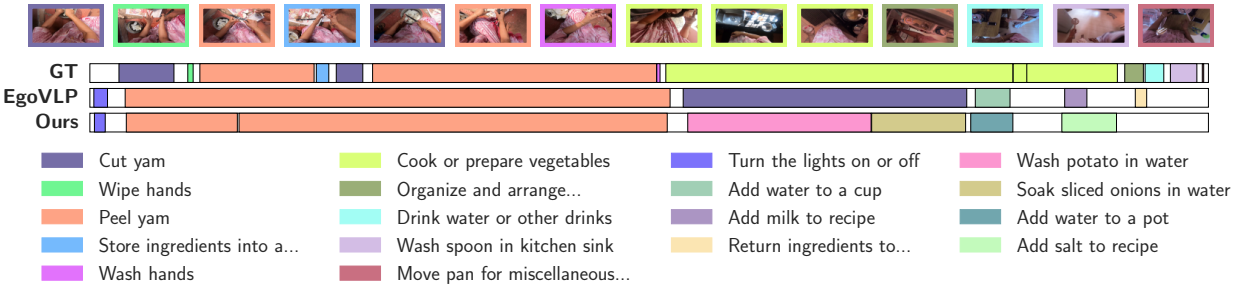
[18] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. 11

[19] Yifei Huang, Yusuke Sugano, and Yoichi Sato. Improving action segmentation via graph-based temporal reasoning. In *CVPR*, 2020. 4

[20] Md Mohaiminul Islam, Ngan Ho, Xitong Yang, Tushar Nagarajan, Lorenzo Torresani, and Gedas Bertasius. Video recap: Recursive captioning of hour-long videos. In *CVPR*, 2024. 2

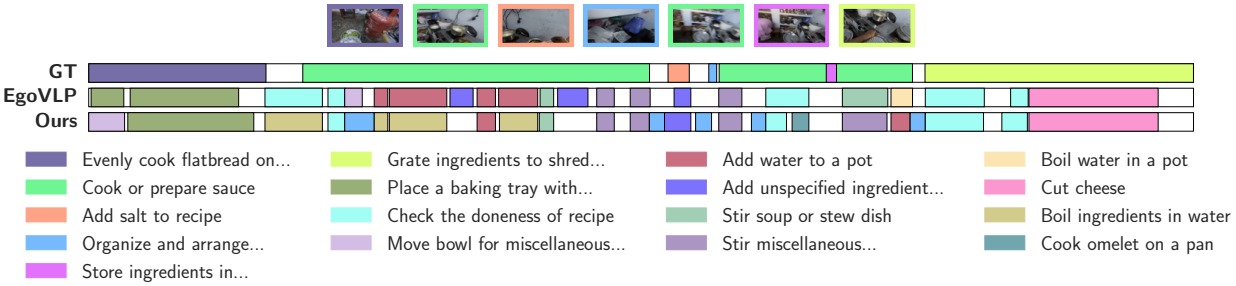


(a) Video 1 (`2dbb7845-1de0-4e26-877a-035c051d12a5`)



(b) Video 2 (`f4cc5fdc-f64f-4dd7-9b95-61db9bbf33d5`)

(c) Video 3 (`c546c508-8352-4c5c-8770-e8f30fb4562a`)



(d) Video 4 (`acc6839e-9d6d-46db-921b-51812834d3b2`)

Figure 7. **Failure cases on the Zero-Shot Localization task on Goal-Step [48]**, showing the ground truth steps, the predictions obtained by clustering the EgoVLP and HiERO features and the middle frame of each step from the ground truth. We find that most cases of failure are related to a mismatch between the granularity of ground truth steps and predictions. In **Video 1** (Fig. 7a), both EgoVLP and HiERO detect the most occurring step (■ "*Bake or roast ingredients in oven*"), but EgoVLP is breaking the segment into more clusters and both methods confuse it with a similar step (■ "*Put the dough on the baking tray*"). In **Video 2** (Fig. 7b), both EgoVLP and HiERO group the initial part of the video in a single long step (■ "*Peel yam*"). In the second half of the video, HiERO predicts more fine-grained steps than the ground truth, *e.g.*, (■ "*Wash potato in water*") rather than (■ "*Cook or prepare vegetable*"). A similar issue appears in **Video 3** (Fig. 7c), in which there is a mismatch between the step ground truth, *e.g.*, (■ "*Boil noodles in water*") and (■ "*Drain noodle*") and the predicted finer steps. **Video 4** (Fig. 7d) shows a more significant failure case where both methods predict many more steps than in the ground truth.

[21] Youngkyoon Jang, Brian Sullivan, Casimir Ludwig, Iain Gilchrist, Dima Damen, and Walterio Mayol-Cuevas. Epic-tent: An egocentric video dataset for camping tent assembly. In *ICCVW*, 2019. 7, 10, 11, 13

[22] Baoxiong Jia, Ting Lei, Song-Chun Zhu, and Siyuan Huang. Egotaskqa: Understanding human tasks in egocentric videos. In *NeurIPS*, 2022. 2

[23] Matthew Kowal, Mennatullah Siam, Md Amirul Islam, Neil DB Bruce, Richard P Wildes, and Konstantinos G Derpanis. Quantifying and learning static vs. dynamic information in deep spatiotemporal networks. *IEEE TPAMI*, 2024. 2

[24] H. Kuehne, A. B. Arslan, and T. Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *CVPR*, 2014. 3

[25] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 1955. 7

[26] Sateesh Kumar, Sanjay Haresh, Awais Ahmed, Andrey Konin, M Zeeshan Zia, and Quoc-Huy Tran. Unsupervised action segmentation by joint representation learning and online clustering. In *CVPR*, 2022. 2

[27] Jun Li and Sinisa Todorovic. Set-constrained viterbi for set-supervised action segmentation. In *CVPR*, 2020. 2

[28] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *ECCV*, 2018. 7, 10, 11, 13

[29] 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 *NeurIPS*, 2022. 2, 4, 5, 6, 8, 10, 11, 12, 13

[30] Xudong Lin, Fabio Petroni, Gedas Bertasius, Marcus Rohrbach, Shih-Fu Chang, and Lorenzo Torresani. Learning to recognize procedural activities with distant supervision. In *CVPR*, 2022. 2

[31] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. In *NeurIPS*, 2023. 2

[32] Effrosyni Mavroudi, Triantafyllos Afouras, and Lorenzo Torresani. Learning to ground instructional articles in videos through narrations. In *ICCV*, 2023. 2

[33] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *CVPR*, 2022. 2

[34] Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. Grounded human-object interaction hotspots from video. In *ICCV*, 2019. 2

[35] Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. Ego-topo: Environment affordances from egocentric video. In *CVPR*, 2020. 2

[36] Zwe Naing and Ehsan Elhamifar. Procedure completion by learning from partial summaries. In *BMVC*, 2020. 2

[37] Simone Alberto Peirone, Francesca Pistilli, Antonio Alliegro, and Giuseppe Averta. A backpack full of skills: Egocentric video understanding with diverse task perspectives. In *CVPR*, 2024. 2

[38] Simone Alberto Peirone, Francesca Pistilli, Antonio Alliegro, Tatiana Tommasi, and Giuseppe Averta. Hier-egopack: Hierarchical egocentric video understanding with diverse task perspectives. *arXiv preprint arXiv:2502.02487*, 2025. 4, 11

[39] 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 *ICCV*, 2023. 1, 2, 5, 6, 10

[40] Will Price, Carl Vondrick, and Dima Damen. Unweavenet: Unweaving activity stories. In *CVPR*, 2022. 1, 2

[41] Francesco Ragusa, Antonino Furnari, and Giovanni Maria Farinella. Meccano: A multimodal egocentric dataset for humans behavior understanding in the industrial-like domain. *CVIU*, 2023. 7, 10, 11, 13, 14

[42] Alexander Richard, Hilde Kuehne, Ahsan Iqbal, and Juergen Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. In *CVPR*, 2018. 2

[43] V Sanh. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. 6, 11

[44] Saquib Sarfraz, Naila Murray, Vivek Sharma, Ali Diba, Luc Van Gool, and Rainer Stiefelhagen. Temporally-weighted hierarchical clustering for unsupervised action segmentation. In *CVPR*, 2021. 1, 2, 7

[45] Luigi Seminara, Giovanni Maria Farinella, and Antonino Furnari. Differentiable task graph learning: Procedural activity representation and online mistake detection from egocentric videos. In *NeurIPS*, 2024. 2, 3

[46] Yuhan Shen and Ehsan Elhamifar. Progress-aware online action segmentation for egocentric procedural task videos. In *CVPR*, 2024. 2

[47] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE TPAMI*, 2000. 2

[48] Yale Song, Eugene Byrne, Tushar Nagarajan, Huiyu Wang, Miguel Martin, and Lorenzo Torresani. Ego4d goal-step: Toward hierarchical understanding of procedural activities. In *NeurIPS*, 2024. 1, 2, 6, 7, 8, 10, 12, 13, 15

[49] Sebastian Stein and Stephen J McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, 2013. 3

[50] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *NeurIPS*, 2022. 1

[51] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2007. 3

[52] Huiyu Wang, Mitesh Kumar Singh, and Lorenzo Torresani. Ego-only: Egocentric action detection without exocentric transferring. In *ICCV*, 2023. 8

[53] Xudong Wang, Ishan Misra, Ziyun Zeng, Rohit Girdhar, and Trevor Darrell. Videocutler: Surprisingly simple unsupervised video instance segmentation. In *CVPR*, 2024. 2

[54] Yangtao Wang, Xi Shen, Yuan Yuan, Yuming Du, Maomao Li, Shell Xu Hu, James L Crowley, and Dominique Vaufreydaz. Tokencut: Segmenting objects in images and videos

with self-supervised transformer and normalized cut. *IEEE TPAMI*, 2023. 2

[55] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *ECCV*, 2022. 7, 8, 10

[56] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020. 6, 7, 10, 12

[57] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *CVPR*, 2023. 1, 2, 5, 6, 10, 11, 12

[58] Yiwu Zhong, Licheng Yu, Yang Bai, Shangwen Li, Xueting Yan, and Yin Li. Learning procedure-aware video representation from instructional videos and their narrations. In *CVPR*, 2023. 2

[59] Honglu Zhou, Roberto Martín-Martín, Mubbasir Kapadia, Silvio Savarese, and Juan Carlos Niebles. Procedure-aware pretraining for instructional video understanding. In *CVPR*, 2023. 2

[60] Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 2

[61] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *CVPR*, 2019. 2