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

Simone Alberto Peirone Francesca Pistilli Giuseppe Averta
Politecnico di Torino
simone.peirone@polito.it

Abstract

Human activities are particularly complex and variable, and this makes challenging for deep learning models to reason about them. However, we note that such variability does have an underlying structure, composed of a hierarchy of patterns of related actions. We argue that such structure can emerge naturally from unscripted videos of human activities, and can be leveraged to better reason about their content. We present HiERO, a weakly-supervised method to enrich video segments features with the corresponding hierarchical activity threads. We prove the potential of our enriched features with multiple video-text alignment benchmarks (EgoMCQ, EgoNLQ) with minimal additional training, and in zero-shot for procedure learning tasks (Ego4D Goal-Step). Our results prove the relevance of using knowledge of the hierarchy of human activities for multiple reasoning tasks in egocentric vision.
Project page: github.com/sapeirone/HiERO.

1. Introduction

Think about a typical home routine. You enter the kitchen, grab onions and carrots, chop them, and put them in a pan on the stove with oil. At the same time, you fill a pot with water and put it on the stove. While you wait the water to boil to cook the pasta, you pour some tomatoes in the pan. Zooming out, these actions fall into interleaved threads like preparing vegetables and cooking pasta, both part of a broader routine like preparing a meal, which may overlap with others like washing dishes. Egocentric video understanding has traditionally focused on isolated actions [1, 2], often neglecting the hierarchical structure of human activity [11]. Procedure Learning (PL) addresses this to some extent, but typically only models a single level of aggregation with supervised training on scripted examples. Conversely, we claim that there is significant value in learning from the hierarchy of human behavior at multiple levels of abstraction. Indeed, the richness of human activities lies not only in single actions execution, but more prominently in how these are interconnected at different levels of abstractions. These structures can emerge without supervision, but their quality depends on the feature extractor. Video models may cluster actions by visual similarity [7], semantic similarity (*e.g.*, dicing carrots or slicing an onion) [5], or functional similarity—grouping steps that contribute to a shared goal, like meal preparation. We introduce HiERO, a hierarchical architecture with a Temporal Encoder that aggregates local temporal context and a Function-Aware Decoder that discovers functionally coherent clusters using spectral graph clustering. In this context, activity patterns emerge as strongly connected regions capturing actions that are functionally and temporally related, allowing the model to reason on higher-level activities (Fig. 1). HiERO can perform a wide set of reasoning tasks, including natural language queries, step grounding, and others, mostly in zero-shot.

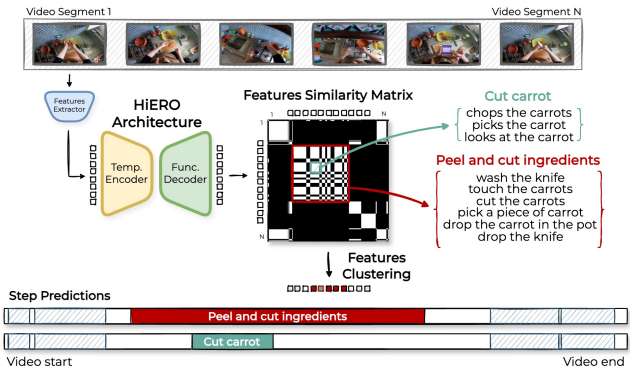


Figure 1. **Zero-Shot procedure step localization with HiERO**. Given a long egocentric video, HiERO computes segment-level features that encode the functional dependencies between the actions in the video at different scales.

2. Related works

Long-form understanding. Long-form video understanding in egocentric vision requires diverse reasoning abilities to grasp the broader context of human activities [4, 6], interpret interactions between objects, people, and locations [7], and model the procedural nature of human activities [10].

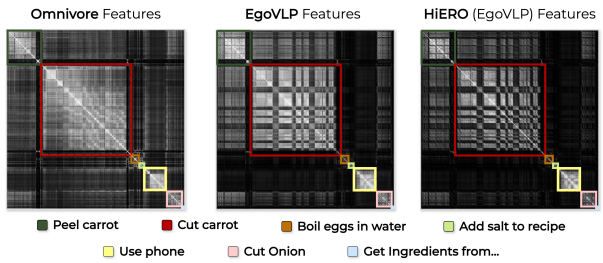


Figure 2. **Step clusters in the features similarity matrix of a video from Ego4D [3].** Colored rectangles are the GT steps.

Several approaches learn transferable representations for downstream video understanding tasks by aligning short video clips and their corresponding textual narrations [9]. HierVL [1] extends this approach by incorporating video-level alignment through summaries. Conversely, HiERO captures long-range functional dependencies between human actions without requiring explicit supervision or instructional video datasets.

3. Method

We design HiERO based on the intuition that, given a sufficiently large collection of videos capturing human activities in-the-wild, *functional dependencies* between actions naturally emerge as frequently co-occurring patterns directly from observations [10]. With HiERO, we learn a feature space that captures these functional dependencies between actions, *i.e.*, those that frequently co-occur together are close to each other and distant from the others. As a result, such space allows related actions to be easily grouped into high-level patterns with a simple clustering operation. Our approach represents the video as a graph, in which nodes correspond to short temporal segments, ideally representing one or a few actions, and detects functional threads as regions of this graph whose nodes encode similar actions based on their feature similarity.

Functional threads discovery. In our setting, we define a strongly connected region as a group of graph nodes with high *functional similarity*. The concept of similarity strongly depends on the backbone used for node embeddings. If the backbone maps semantically similar actions—like *cutting an onion* and *peeling a carrot*—close in feature space, these regions reflect high-level *functional threads* (e.g., *preparing vegetables*). Figure 2 illustrates how different backbones affect feature similarity. Omnivore, trained for supervised visual classification, emphasizes appearance-based similarity. In contrast, EgoVLP, trained with narration supervision, reveals more coherent functional regions, even without explicit step-level labels. These regions often align with procedural steps or sub-steps. Our method leverages this structure to cluster nodes into high-level functional threads, ultimately partitioning a graph $\mathcal{G}$ into n subgraphs $\mathcal{G}_1, \ldots, \mathcal{G}_n$, each representing a distinct step in the video.

3.1. The HiERO architecture

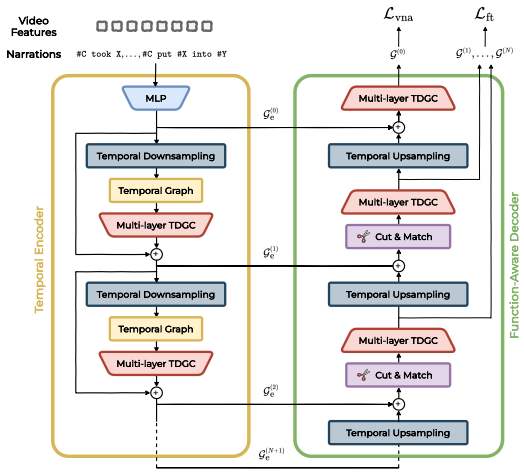


Figure 3. **Architecture of HiERO.**

Inspired by previous works in video understanding [8], we encode an input video $\mathcal{V}$ as a *video graph* with N nodes $\mathcal{G} = (\mathbf{X}, \mathcal{E}, \mathbf{p})$, where $\mathbf{X} \in \mathbb{R}^{N \times D}$ is the node embeddings matrix, edge $e_{ij} \in \mathcal{E}$ connects nodes i and j if their temporal distance is smaller than a threshold τ and the attribute $\mathbf{p} \in \mathbb{R}^N$ encodes the temporal position of each node. Each node represents a fixed-length segment of the video and the node embeddings are computed using a video features extractor from the segment frames. At training time, each video is also associated with a set of narrations, denoted as $\mathcal{T}_\mathcal{V} = \{(n_i, t_i)\}_i$, where n_i and t_i are the textual narration and its corresponding timestamp. HiERO is built as an encoder-decoder architecture (Fig. 3).

Temporal encoder. The *Temporal Encoder* $\mathcal{E}$ is implemented as a stack of N_l GNN-based blocks with temporal subsampling operations to map the input video graph $\mathcal{G}^{(0)}$ to a set of temporally coarsened representations. Each stage is composed of multiple TDGC [8] layers that implement temporal reasoning on the graph by combining the embedding of node i with a learnable projection of its temporal neighbors $\mathcal{N}(i)$, *i.e.* nodes within a certain temporal distance d. Then, the nodes are subsampled to halve the temporal resolution of the graph and obtain $\mathcal{G}^{(l+1)}$, which is fed to the next layer of the encoder. Therefore, the encoder progressively extends the temporal context of the nodes, regardless of whether the actions performed are related or not.

Function-Aware decoder. The *Function-Aware decoder* $\mathcal{D}$ shares the same architecture of the encoder with one significant difference: instead of implementing message passing on the local temporal neighborhood of the nodes, each decoder stage first groups the graph nodes based on their func-

tional similarity, *i.e.*, whether they represent functionally similar actions, and then implements temporal reasoning on each group separately. This procedure connects nodes that may be temporally distant but encode similar actions (*functional threads*), allowing the model to reason about long-term patterns not necessarily connected in time. At each stage l the decoder takes the sum of the graph $\mathcal{G}_e^l$ from the corresponding temporal encoder stage and the interpolated output of the previous layer of the decoder and feed it to the *Cut & Match* module, which partitions via spectral clustering [12] the graph into a set of K smaller graphs each corresponding to a group of functionally similar nodes. After this process, nodes that correspond to far apart segments of the video may be clustered together. We then use TDGC to perform temporal reasoning into each partition separately and map the nodes back to the original graph.

3.1.1. Training HiERO

We train HiERO to map video segments representing co-occurring actions close in the feature space $\mathcal{L}_{vna}$ and to detect functional threads not necessarily close in time $\mathcal{L}_{ft}$. HiERO is trained with a combination of the two losses.

Video-narrations alignment. The *video-narrations alignment loss* $\mathcal{L}_{vna}$ encourages temporally co-occurring actions to be closer in the embedding space. Inspired by prior video-language models [5, 9], it uses a contrastive loss that brings node embeddings closer to narrations within a temporal window (*positives*) and pushes away others (*negatives*). Unlike prior work [5], which aligns each node to a single narration, our method accounts for multiple co-occurring narrations, producing more context-aware embeddings that better capture high-level action patterns.

Functional threads loss. Aligning the visual embeddings from larger temporal windows to their corresponding textual descriptions is more difficult. Using narrations is impractical as they are too fine-grained and the number of positive and negatives samples would grow rapidly with the depth of the network and the size of the alignment window. Other forms of *high-level* supervision, *e.g.*, video summaries, require huge annotation efforts. Instead, we apply video-narrations alignment only on the output of the decoder and introduce a contrastive regularization objective to make features at deeper layers belonging to the same functional thread more similar to each other. The *functional threads loss* $\mathcal{L}_{ft}$ leverages the graph partition assignments from the *Cut & Match* modules in the decoder.

4. Experiments

We train HiERO on EgoClip [5], a curated set of 3.8M clip-text pairs obtained from Ego4D textual narrations, using pre-extracted features from several backbones, *i.e.*, Omnivore [2], EgoVLP [5] and LaViLa [14], showing that HiERO can be easily applied to different backbones.

Method	EgoMCQ		EgoNLQ			
	Accuracy (%)		mIOU@0.3		mIOU@0.5	
	Inter	Intra	R@1	R@5	R@1	R@5
Omnivore [2]† (CVPR'22)	–	–	6.56	12.55	3.59	7.90
EgoVLP [5] (NIPS'22)	90.6	57.2	10.84	18.84	6.81	13.45
HierVL-Avg [1] (CVPR'23)	90.3	53.1	–	–	–	–
LaViLa [14] (CVPR'23)	94.5	63.1	12.05	22.38	7.43	15.44
EgoVLPv2 [9] (ICCV'23)	91.0	60.9	12.95	**23.80**	7.91	**16.11**
Ours (Omnivore)	90.1	53.4	10.27	18.20	6.01	12.52
Ours (EgoVLP)	91.6	59.6	11.41	19.67	7.05	13.91
Ours (LaViLa)	**94.6**	**64.4**	**13.35**	21.12	**8.08**	15.31

Table 1. **Results on EgoMCQ and EgoNLQ's validation set**, using VSLNet [13] as grounding head for the latter. †Reproduced.

Evaluation benchmarks. We evaluate our approach on several egocentric vision benchmarks to validate its effectiveness in different scenarios. We validate the video-text alignment components of HiERO on **EgoMCQ** [5], a set of 39K *text-to-video* multiple-choice questions derived from Ego4D narrations, and **EgoNLQ**, a natural language queries benchmark that aims to localize the segment of a video (start and end timestamps) answering a given textual query. For Procedure Learning, we evaluate HiERO on the Step Grounding and Step Localization tasks from **GoalStep** [11]. The design of HiERO allows to address these tasks in a completely *zero-shot* setting.

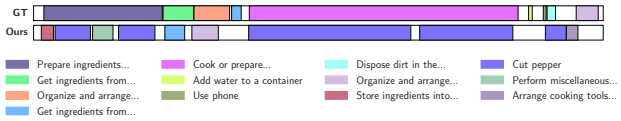
4.1. Quantitative Results

4.1.1. Video-Text Alignment on EgoMCQ

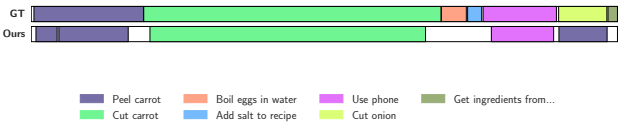
We evaluate HiERO on EgoMCQ [5] and EgoNLQ [3] to validate its video-text alignment capabilities and to show that reasoning on functional threads at different scales can support various video understanding tasks (Table 1). Our window-based alignment loss encourages HiERO to learn functional dependencies between actions, while clustering groups together similar actions at different scales and over a long temporal horizon. Together, these objectives are effective to discriminate between similar short-term actions, which is critical for EgoMCQ, as well as to capture long-range causal and temporal dependencies in the video, which is essential for EgoNLQ. Unlike other backbones that extract features from a short temporal window and rely entirely on the grounding head for high-level reasoning, our features inherently capture a broader semantic understanding of the video. In both benchmarks, HiERO significantly improves the SOTA, regardless of the features extraction backbone (+1.3% on intra accuracy on EgoMCQ and Top-1 Recall at IoU = 0.3 on EgoNLQ). Remarkably, HiERO achieves good results even with Omnivore features, despite not being trained end-to-end on Ego4D.

4.1.2. Step Grounding

This task aims to localize a procedure step given its description in natural language. Performance is measured with Recall at different IoU thresholds. The supervised baseline



(a) Video 1 (`0f07958c-04e3-4be9-9118-f3313c4e183e`)



(b) Video 4 (`64b355f3-ef49-4990-8622-9e9eef68b495`)

Figure 4. **Zero-Shot Localization qualitative results on [11].**

Method	Approach	mIoU@0.3		mIoU@0.5	
		R@1	R@5	R@1	R@5
Omnivore [11]	Supervised	12.02	19.99	7.71	14.17
EgoVLP	Zero-Shot	10.73	24.70	7.38	16.53
Ours (Omnivore)	Zero-Shot	9.29	22.89	6.24	15.05
Ours (EgoVLP)	Zero-Shot	**11.57**	**27.41**	**7.87**	**18.70**

Table 2. **Step-Grounding on Ego4D Goal-Step [11].**

proposed in [11] leverages VSLNet [13] as grounding head on top of the Omnivore pre-extracted features. Instead, we adapt HiERO to this task by clustering the video segments and selecting as prediction candidates the segment whose average visual features are most similar to the textual features of the query step. This allows to address the grounding task in *zero-shot* without any additional training. Table 2 shows that HiERO consistently outperforms the Omnivore and EgoVLP baselines in the supervised setting. In *zero-shot*, HiERO beats the supervised counterpart on Top-5 Recall and achieves results close to SOTA on the other metrics.

4.2. Qualitative results on Step Localization

We show in Fig. 4 some success and failure cases in the *zero-shot* Step Localization task on Goal-Step. This task aims to predict triplets `(start time, end time, label)` for all the procedure steps and substeps in the video. We adapt HiERO to this task in zero shot by clustering the output features to localize the steps and use the similarity between the visual and the textual features of the steps taxonomy to predict their labels. We observe that many failure cases of our approach are related to the ambiguous granularity of the step labels in the ground truth, which leads to confusion between steps that could be either steps or sub-steps, *e.g.*, *Cook or prepare the vegetables* and *Cut the pepper* in Fig. 4a.

5. Limitations

The actions hierarchy learned by our approach does not adhere to a predefined taxonomy, which makes our approach flexible but also prone to ambiguities when evaluated on benchmarks that assume a fixed set of possible steps. Also, the imbalanced scenarios distribution in Ego4d may impact the learned actions hierarchy, leading to low quality of the learned representation on the least represented scenarios.

6. Conclusions

In this paper, we discuss the relevance of learning about the hierarchical structure of human behavior collected in egocentric videos. We propose HiERO, a weakly-supervised method able to fully exploit functional threads to enhance reasoning capabilities. HiERO features proved their suitability for video-text alignment tasks, and in zero-shot for procedural learning tasks, proving the effectiveness and importance of using functional reasoning at multiple levels.

References

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

[2] 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.

[3] Kristen Grauman et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022.

[4] Md Mohaiminul Islam et al. Video recap: Recursive captioning of hour-long videos. In *CVPR*, 2024.

[5] Kevin Qinghong Lin, Jinpeng Wang, et al. Egocentric video-language pretraining. In *NeurIPS*, 2022.

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

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

[8] 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.

[9] Shraman Pramanick et al. Egovlpv2: Egocentric video-language pre-training with fusion in the backbone. In *ICCV*, 2023.

[10] 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.

[11] 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.

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

[13] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization. In *ACL*, 2020.

[14] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *CVPR*, 2023.