Supplementary Material for HierVL: Learning Hierarchical Video-Language Embeddings

Kumar Ashutosh[1], Rohit Girdhar[2], Lorenzo Torresani[2], Kristen Grauman[1,2]
[1]UT Austin, [2]FAIR, Meta AI

List of Contents

This supplementary material contains the following additional information

1. A **video** showing
    - Our idea and key challenges
    - Network architecture
    - Pretraining setup and evaluation
    - Downstream tasks overview and results
    - Visualization of learned representation
2. Setup details of T-SNE visualization (Fig. 3).
3. Setup details of downstream tasks
4. Scaling effect of HierVL-SA and EgoVLP

1. Demo Video

Please see our project page for the video showing our idea, network architecture, experimental setup, downstream tasks and visualizations: https://vision.cs.utexas.edu/projects/hiervl/

2. Setup details of T-SNE visualization

We visualize and compare the learned representations between HierVL-SA and EgoVLP. We take 500 summary texts and their child-level narrations and extract features using our text encoder f_n. Next, we run a T-SNE optimization [3] on these hierarchical features and plot 30 summary representations (bigger circle with transparency) and their closest $k\%$ child features (smaller bold circles). We set $k = 75$. We can see our HierVL-SA is able to bring together summary and narration representations. Even though one might expect that summaries and narrations are similar; their representations are quite far if they are not trained hierarchically, as seen in EgoVLP (right). This reinforces how summary-level descriptions capture a different granularity of understanding for the video, beyond the literal steps of the actions. Our HierVL links the two levels to inform clip representations with broader context about the general goal of the activity and its sequence.

3. Setup details of downstream tasks

In this section, we detail our training hyperparameters for all the downstream tasks discussed in Sec. 4.2.

3.1. Ego4D Long-Term Anticipation

For the Ego4D long-term anticipation, we use our pretrained video representation f_v and the aggregator Agg for 150 epochs. We use the codebase released by Ego4D [2]. We replace the default SlowFast [1] architecture of the baseline with our video representation f_v—which is kept frozen. Next, we use our aggregator Agg, followed by a multi-head decoder. Every multi-head decoder d_i predicts the next i^{th} verb. Same with noun. Finally, predicted action is a combination of verb and noun. We did a hyperparameter search and selected the best performing configuration based on validation performance.

Verb prediction. There are 115 verb classes in the dataset. We use a batch size of 128 and use two nodes (eight 32GB GPUs each). The learning rate of multi-head decoder and aggregator Agg is set to 3×10^{-3} and 1×10^{-4}. The dropout is set to 0.5.

Noun prediction. There are 478 noun classes in the dataset. We again use a batch size of 128 and use two nodes (eight 32GB GPUs each). The learning rate in this case is 5×10^{-3} for multi-head decoder and 5×10^{-4} for aggregator Agg. To avoid overfitting, the dropout is set to 0.6.

3.2. CharadesEgo Action Recognition

For the zero-shot setting, f_V is non-trainable. For the fine-tuned setting, f_v is trainable. We choose a batch size of 32 on eight 32GB GPUs (single node). The learning rate is 3×10^{-5}. The model is trained for five epochs.

3.3. EPIC-KITCHENS Multi-Instance Retrieval

For the zero-shot setting, none of f_v, f_n is trainable. In the fine-tuned setting, both f_v and f_n are trainable. We choose a batch size of 32 on eight 32GB GPUs (single node). The model is trained for 200 epochs. The learning rate is updated with 3×10^{-5} for 100 epochs and 1×10^{-6} thereafter.

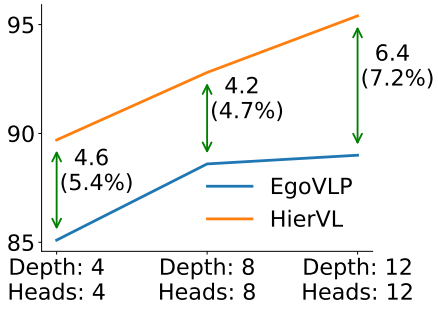

Figure 1. Accuracy on SummaryMCQ task for different architecture sizes.

4. Scaling effect of HierVL-SA and EgoVLP

We investigate if the need for hierarchical annotations would diminish if a bigger model is used. While it is infeasible to increase the network size with the same hyperparameters (batch size, #gpus, etc.) we can check the trend by decreasing the model size. See Fig 1. We experiment with three TimeSformer backbone sizes — depth and num-heads as 4, 8 and 12 on the SummaryMCQ task (Main Table 1). We see that as the network size is made larger the performance of HierVL keeps on increasing. Meanwhile, the performance of EgoVLP plateaus. This suggests that our idea's advantage does not get subsumed by a larger network.

References

[1] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 1

[2] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. 1

[3] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 1