AdaVid: Adaptive Video-Language Pretraining
———— Supplementary Material ————

Chaitanya Patel Juan Carlos Niebles Ehsan Adeli
Stanford University
https://chaitanya100100.github.io/AdaVid/

A. FLOPs computation

A.1. Multi-Head Self-Attention

We follow [3] for calculating the total FLOPs of Multi-head self attention as $8ND^2 + 4N^2D$, where N represents the number of tokens and D denotes the token dimension.

This computation is detailed as follows: The FLOPs required for matrix multiplication of sizes $(M \times N)$ and $(N \times P)$ is $2MNP$. For each attention head, each projection matrix (query, key, value) involves $2 \times N \times D \times \frac{D}{H}$ FLOPs. Thus, the total FLOPs for this operation are $6ND^2$. Each head computes the dot product between the query and key, involving $2 \times N \times \frac{D}{H} \times N$ FLOPs. The total FLOPs for this step amount to $2N^2D$ Each head also computes the weighted sum of values, requiring $2 \times N \times N \times \frac{D}{H}$ FLOPs. Thus, the total FLOPs for this computation are $2N^2D$. Finally, the output projection layer involves $2 \times N \times D \times D = 2ND^2$ FLOPs.

A.2. Feed-forward network

The token-wise feed-forward (MLP) layer consists of two matrix multiplications: The first transformation converts from dimension D to dimension F, while the second converts from dimension F back to dimension D. Each matrix multiplication operation involves $2NDF$ FLOPs. Therefore, the total number of FLOPs for the feed-forward network (FFN) is $4NDF$. Typically, the value of F is set to $4D$, resulting in a total of $16ND^2$ FLOPs.

A.3. Transformer Layers for Video Encoder

As mentioned in the paper, input video with T frames are patchified where each frame has N tokens, giving total TN number of tokens. These tokens can be processed by multiple transformer layers, each consisting of an attention module and a FFN module. If each layer uses **Dense Attention**, then its total complexity would be $24TND^2 + 4T^2N^2D$.

Transformer layer with **Space-Time Attention** contains 3 modules: space attention, time attention and FNN [2]. Space attention requires $T(8ND^2 + 4N^2D)$ and time attention requires $N(8TD^2 + 4T^2D)$. Thus, the total complexity of a space-time layer is $32TND^2 + 4TND(N + T)$.

For **hierarchical modeling**, the input video is divided into S segments of T/S frame each [1], and processed by a video encoder with space-time attention. This gives effective complexity of each transformer layer to be $32TND^2 + 4TND(N + T/S)$.

B. Qualitative Results

In Figure 1 and Figure 2, we show visualizations of our model's predictions under various evaluation configurations for EgoMCQ and EgoSchema benchmarks, respectively.

C. AdaVid-Agg Architecture

Figure 3 shows the architecture of the lightweight hierarchical AdaVid-Agg model. AdaVid-Agg is designed to encode a sequence of AdaVid-EgoVLP features into the feature representation for long videos. During the training of AdaVid-Agg, we extract features from AdaVid-EgoVLP using various evaluation configurations. This approach enables the entire pipeline to be compute adaptive during inference.

References

[1] Kumar Ashutosh, Rohit Girdhar, Lorenzo Torresani, and Kristen Grauman. Hiervl: Learning hierarchical video-language embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23066–23078, 2023. 1

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, page 4, 2021. 1

[3] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. 1

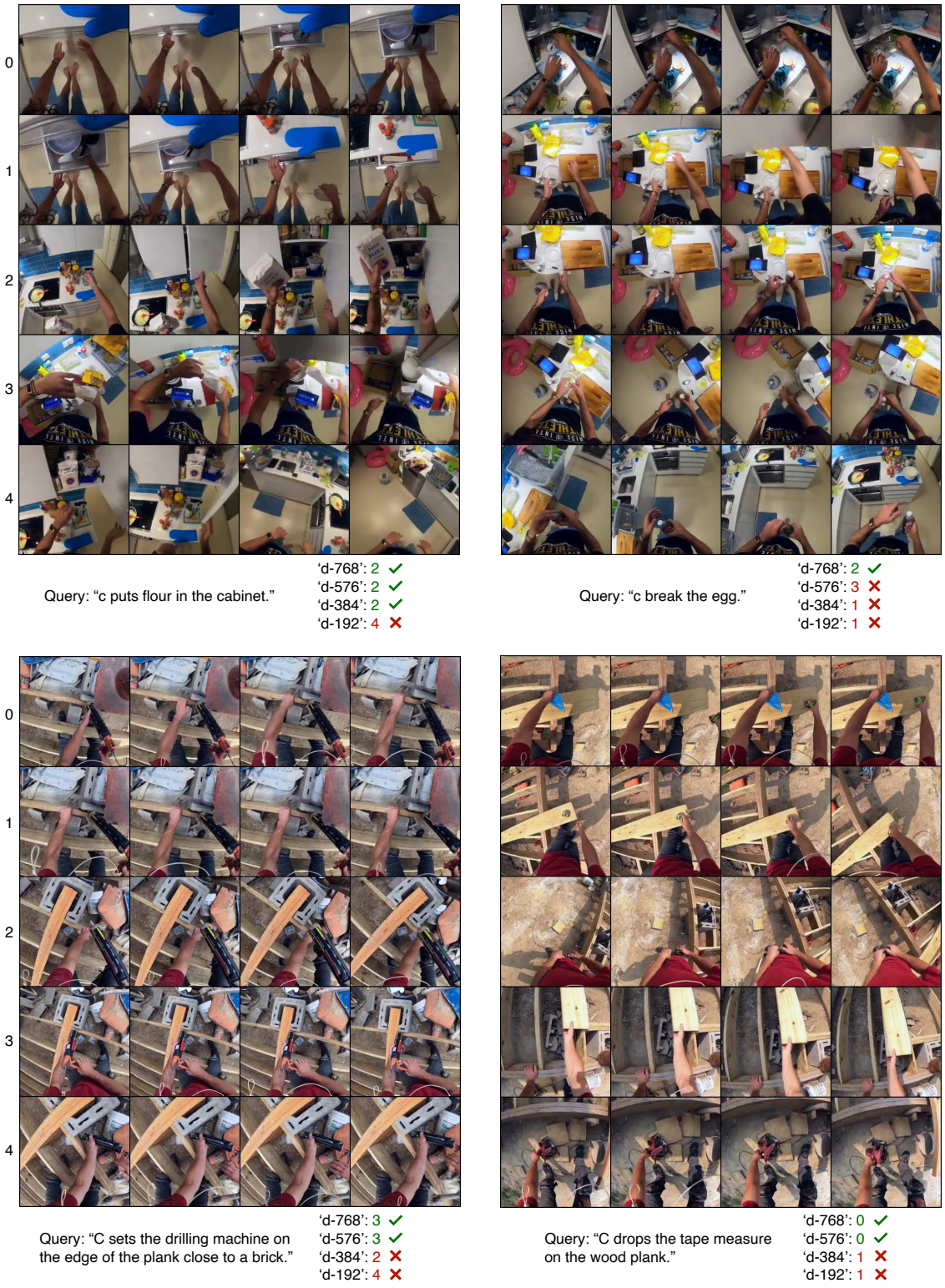


Figure 1. We show four challenging examples from the EgoMCQ(intra) benchmark, each consisting of a text query and five candidate video clips. We also show the predictions made by our AdaVid-EgoVLP model under four different evaluation configurations. The results indicate that the model can do accurate fine-grained video analysis by adaptively increasing its compute.

What is the primary focus of activity in the video and how does interaction between c and the child contribute to this?

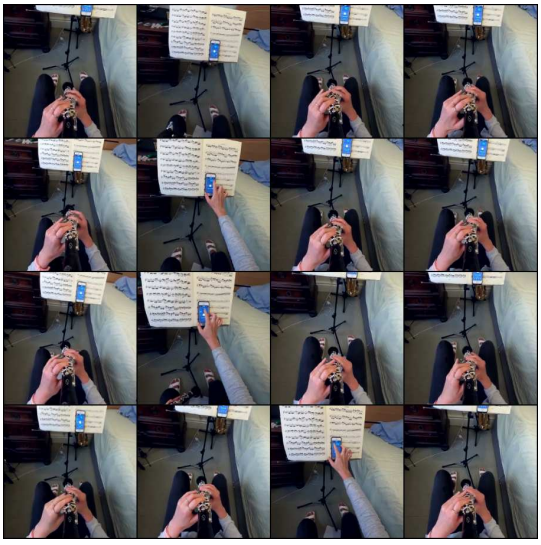
0: C is making a hat. the child interacts with c by talking to her and occasionally touching the hat.
1: C is making a sweater. the child interacts with c by talking to her and occasionally touching the sweater.
2: C is making a pair of gloves. the child interacts with c by talking to her and occasionally touching the gloves.
3: C is making a blanket. the child interacts with c by talking to her and occasionally touching the blanket.
4: C is knitting a scarf. the child interacts with c by talking to her and occasionally touching the scarf.

'd-768': 4 ✓
'd-576': 4 ✓
'd-384': 4 ✓
'd-192': 4 ✓

What is the primary process being undertaken in this video, and how does it consist of both repetitive actions and brief moments of interpersonal interaction?

0: Currently, c is in the process of making a delightful cake.
1: C is making kaliche ladoo balls.
2: Currently, c is in the process of making a delicious pie.
3: Currently, c is in the process of making a delicious pizza.
4: C is making a sandwich.

'd-768': 1 ✓
'd-576': 1 ✓
'd-384': 1 ✓
'd-192': 0 ✗



What are the similarities and differences in c's handling of the saxophone throughout the video?

0: C holds the saxophone with both hands when playing it, and with her left hand when not playing it.
1: C holds the saxophone with her right hand when playing it, and with her left hand when not playing it.
2: C holds the saxophone with both hands at all times.
3: C holds the saxophone with her left hand at all times.
4: C holds the saxophone with her right hand when playing it, and with both hands when not playing it.

'd-768': 0 ✓
'd-576': 0 ✓
'd-384': 4 ✗
'd-192': 4 ✗

What is the primary objective of c's actions throughout the video, and how can you concisely describe the two main methods he employs?

0: Currently, individual c is engaged in cutting down several trees.
1: C is clearing brush.
2: Currently, c is meticulously trimming the overgrown hedges outside.
3: In the garden, c is diligently removing dead branches from trees.
4: C is pruning the fence.

'd-768': 4 ✓
'd-576': 4 ✓
'd-384': 4 ✓
'd-192': 3 ✗

Figure 2. We show four examples from the EgoSchema VideoQA benchmark, each consisting of a video and a question with 5 candidate answers. We also show the predictions made by our AdaVid-Agg model under four different evaluation configurations. The results indicate that the model can do long-form video analysis efficiently by adaptively increasing its compute.

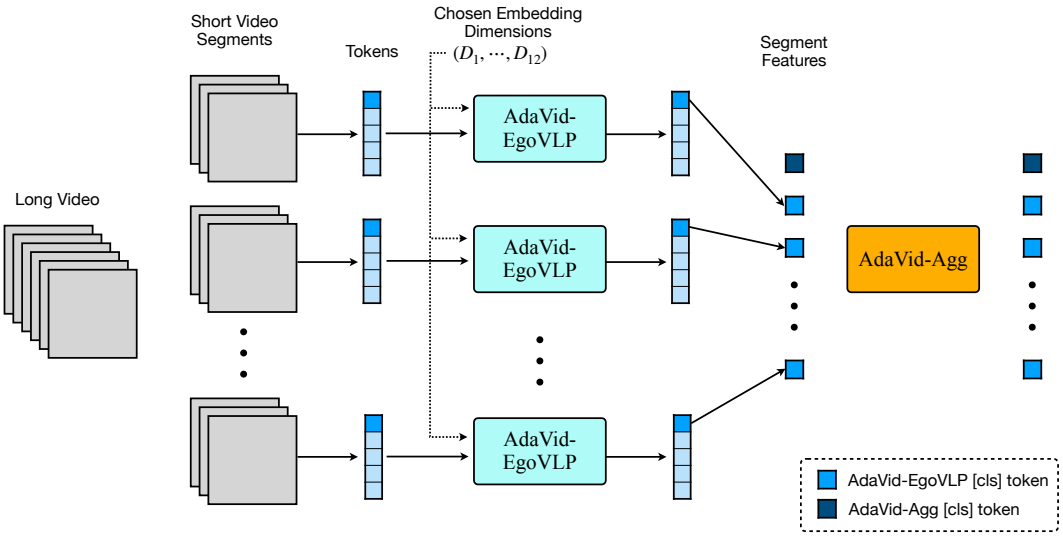


Figure 3. **AdaVid-Agg**: A long video is divided into $S = 16$ shorter segments and encoded using the pretrained AdaVid-EgoVLP model. The sequence of segment features is then processed by the lightweight AdaVid-Agg transformer model, which predicts a single feature representation for the entire video. It is important to note that AdaVid-EgoVLP is trained independently and remains frozen during the training of AdaVid-Agg.