Koala: Key frame-conditioned long video-LLM

Reuben Tan[1] Ximeng Sun[1] Ping Hu[2] Jui-hsien Wang[3] Hanieh Deilamsalehy[3]
Bryan A. Plummer[1] Bryan Russell[3] Kate Saenko[1]
[1]Boston University, [2]University of Electronic Science and Technology of China, [3]Adobe Research
{rxtan, sunxm, pinghu, bplum, saenko}@bu.edu, {juiwang, deilamsa, brussell}@adobe.com

In this supplemental, we provide the following additional material to the main paper:

A. Instruction templates

As mentioned in the main paper, we train our Koala approach on instructional videos from the HowTo100M dataset [7]. The videos are sourced from YouTube using a list of high-level activities obtained from WikiHow[1]. As such, each instructional video has a corresponding high-level task label such as "replace a car tire" and "make a bacon lettuce and tomato sandwich." Given the instruction-tuned nature of the base video-LLM, we manually craft question and response templates as shown in Table 1. In Table 1, we use <VISUAL> as a placeholder for the expression "[INST] <Video><ImageHere></Video>." During finetuning and downstream evaluations, we substitute the "<ImageHere>" token with the final contextualized video tokens and substitute "{task label}" with the corresponding high-level task label. For training, we create the question prompt P and response R by randomly sampling a pair from Table 1.

[1]https://www.wikihow.com/

B. CLIP filtering of training data

We observe instances where the high-level task labels are not visually relevant to the video content. An example of the aforementioned instances is a video of a person simply describing an action without showing it. Given the demonstrated importance of clean data [3] in training instruction-tuned foundation models, we perform video filtering using the pretrained CLIP ViT-L14 [8] model variant.

Specifically, we use CLIP's visual and text encoders $\text{CLIP}_{\text{visual}}$ and $\text{CLIP}_{\text{text}}$ to measure the similarity between N encoded extracted frames for each video $V = \{V_i\}_{i=1}^{N}$ and its corresponding task label L. We uniformly sample 128 frames from each video and keep the video if it satisfies the following constraint:

$$\max_{V_i \in V}(\text{CLIP}_{\text{visual}}(V_i)^T \text{CLIP}_{\text{text}}(L)) \geq \tau, \quad (1)$$

where τ denotes the cosine similarity threshold.

We show examples of filtered videos using the maximum CLIP scores in Figure 1. In the filtering process, we generally observe that selecting videos based on the maximum relevance score of any frame with respect to the high-level task labels yields videos with increased visual diversity across its frames, as compared to using the mean score across all sampled frames. We set τ to be 0.26 in practice after manually inspecting the visual relevance of about 500 videos and their corresponding similarity scores between the video frames and the corresponding task label.

Prompt template	Response template
<VISUAL> What is the most likely objective in the video? [/INST]	The most likely objective in the video is to {task label}.
<VISUAL> What is the most likely goal in the video? [/INST]	The most likely goal is to {task label}.
<VISUAL> What is the person trying to do in the video? [/INST]	The person is trying to {task label}.
<VISUAL> What is happening in the video? [/INST]	This video demonstrates the steps to {task label}.
<VISUAL> Describe the most likely objective in the video. [/INST]	The most likely objective in the video is to {task label}.
<VISUAL> Describe the most likely goal in the video. [/INST]	The most likely goal is to {task label}.
<VISUAL> Describe what the person is trying to do in the video. [/INST]	The person is trying to {task label}.
<VISUAL> Describe what is happening in the video. [/INST]	This video demonstrates the steps to {task label}.

Table 1. **Instruction and sample response templates.** We use these templates to transform high-level goal labels of the finetuning dataset into the instruction tuning format during our finetuning stage. We use <VISUAL> as a placeholder for the expression [INST] <Video><ImageHere></Video>. Note that we substitute the <ImageHere> token with the final contextualized video tokens in practice during finetuning and downstream evaluations.

C. Implementation details

Training. We optimize the learnable weights of our introduced Conditioned Segment (CS) and Conditioned Video (CV) functions using the AdamW [5] optimizer for two epochs. We also adopt a linear warmup schedule over 10% of training steps with a maximum learning rate of $1e^{-5}$ and gradually anneal it based on a cosine schedule. Our final filtered training set consists of approximately 250K videos in total. In this work, we build our approach off the state-of-the-art Video-LLama [12] model. We train our model on 4 RTX 6000 GPUs. We also define the dimensionality of the outputs of key frames, contextualized segment and inter-segment tokens. For a set of T key frames V_{key}, we define the output of the key frames tokenizer function $\mathcal{F}_{\text{key}}$ as: $z_{\text{key}} \in \mathbb{R}^{N \times D}$, where N and D denote the number and dimensionality of the frozen video queries Q_{video}, respectively. The outputs of our Conditioned Segment and Conditioned Video tokenizer functions z_{segs} and z_{inter} also have similar dimensionality of $\mathbb{R}^{N \times D}$.

Similarly, our segment and inter-segment queries have the same dimensionality of $\mathbb{R}^{N \times D}$. The LLM linear projection functions ϕ project the dimensionality of the key frames tokens z_{key} and contextualized inter-segment tokens z_{inter} from D to D^f where D^f denotes the dimensionality of the textual tokens as input into the frozen LLM. Similar to prior work [12, 13], we set N, D and D^f to be 32, 768 and 4096, respectively. The final value of w in Equation 6 (main) is 0.0203.

Downstream evaluations. We adopt the same evaluation method of calculating log-likelihood for each candidate answer and selecting the highest-scoring option for fair comparisons with prior work [1, 4]. Note that we include the soft video tokens (Section 3 main) in all question-answer prompts. Given the instruction-tuned and generative nature of our final vLLM, we formulate an input text prompt for the zero-shot evaluations on the downstream multiple-choice question answering benchmarks. Specifically, for each question Q and the set of answer options $A = \{a_1, \cdots, a_{||A||}\}$, we experiment with the following manually-crafted text prompt for the j-th candidate answer a_j: "Given the question <Q>, the answer is <a_j>." We compute the final prediction for each question by selecting the answer option that returns the highest logit score for the question and candidate answer pair. For all models and evaluation datasets, we report the best results obtained across varying number of input frames.

D. Evaluation datasets

Zero-shot evaluation benchmarks. Our main goal is to introduce an approach for long-form video understanding. Consequently, we evaluate our proposed Koala approach on several zero-shot long video question answering tasks with the multiple choice format including EgoSchema [6] and procedure-understanding in Seed-Bench [4]. Additionally, we also evaluate on the task of short-term action recognition [4] to analyze if the introduced CS and CV functions are detrimental to understanding short videos.

1. EgoSchema [6] - EgoSchema is a challenging long video question-answering benchmark that contains 5031 3-minutes long videos and each question contains 5 possible options.

2. Seed-Bench Procedure Understanding [4] - The procedure understanding task contains 1170 questions with 4 answer options and the goal is to select the option that specifies the correct sequence of actions.

3. Seed-Bench Action Recognition [4] - To determine the effectiveness of Koala on short-term temporal understanding, we also evaluate on the action recognition task, which contains 1740 questions.

4. NExT-QA [11] - The NExT-QA dataset evaluates a video model's capability to describe and explain temporal actions in videos. NExT-QA contains approx-

Figure 1. **Examples of videos filtered using maximum CLIP scores of video frames with respect to their task labels.** We use the CLIP [48] model to compute a similarity score between each extracted frame and the corresponding task label of the video. We generally observe that filtering videos based on the maximum CLIP score of any frame with respect to the task label results in videos with more visual diversity.

imately 52K question-answer pairs for 5,440 videos. Additionally, these questions are split into several categories such as temporal or descriptive.

E. Additional evaluations

We report the results of our zero-shot evaluation on the test split of the NExT-QA [A] benchmark in Table 2. NExT-QA divides its questions into three categories: **(1) Causal (C)**, **(2) Temporal (T)**, **(3) Description (D)**. Compared to prior work, our approach achieves higher accuracy across the **Causal (C)** and **Temporal (T)** categories, demonstrat-

Video-LLM	Acc_C	Acc_T	Acc_D	Acc_AVG
Video-Llama (finetuned)	27.43	32.14	32.38	29.71
VideoLlama	31.32	35.49	**42.64**	34.47
MovieChat	31.12	35.80	42.49	34.43
Koala (ours)	**32.83**	**38.13**	41.21	**35.85**

Table 2. **Zero-shot evaluation on NExT-QA test split.** We observe that our Koala model performs better than other approaches across most of the different video understanding tasks.

Approach	Aggregate pre-LLM	EgoSchema	GFLOPs
Base	No	33.25	12K
Average	Yes	33.39	12K
Memory module [9]	Yes	34.62	12K
Concatenation	No	35.72	15K
Koala (ours)	Yes	**40.33**	13K

Table 3. **Comparison of performance and efficiency trade-offs between different video aggregation baselines.** We observe that our Koala approach improves the ability of the base vLLM for long-term temporal understanding significantly while only increasing the computational cost marginally.

ing its effectiveness at understanding long temporal context. However, our approach under-performs on **Description (D)** questions that involve counting the ordinality of objects. This result suggests that using curated descriptive annotations for the final finetuning stage, as done in prior work [32, 46, 64], may be beneficial for understanding such concepts.

F. Ablation model baselines and efficiency metrics

We provide additional implementation details on the baseline models in Section 4.2 of the main paper here before describing their performance and efficiency trade-offs. Recall that our goal is to compare Koala to these baselines to better understand how to integrate long-term temporal visual context with vLLMs.

Average. In contrast to existing vLLMs which often just extract a small and fixed number of key frames for each video regardless of its temporal duration, we subsample S segments of T key frames. We encode each segment separately with the key frames tokenizer $\mathcal{F}_{\text{key}}$ and average-pool the key frames tokens over the segments to compute the final visual input z_{final} into the base LLM. Specifically, we compute the final input as:

$$z_{\text{final}} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{F}_{\text{key}}(S_i), \tag{2}$$

where S_i denotes the frames for the i-th segment.

Memory module. A common approach to model long-term temporal context for long videos is to use a feature memory module to store representations of past video segments for lookup. Inspired by [2, 9], we also adopt a simple baseline using a short-term memory module as well as a long-term memory module to mitigate the issue of forgetting information from the distant past. At a high level, we pass in $\mathcal{F}_{\text{key}}(S_i)$ across all video segments into the short- term memory and use the long-term memory tokens as input into the LLM.

The key frames tokenizer function in pretrained vLLMs is often limited by the maximum number of key frames that can be used as input due to the length of the sequence of learnt temporal positional embeddings. To extend the original sequence of positional embeddings, we adopt an approach [10] to hierarchically decompose the learnt positional embeddings such that we can extend them from its initial length n to n^2. We refer interested readers to Song *et al*. [9] for more details.

Concatenation. Last but not least, we also introduce the concatenation ablation to study the importance of aggregating temporal context over the input frames and encoding the information in the soft video tokens *before* projecting them into the feature space of the base LLM. The concatenation baseline differs from the other baselines since it is relying on the self-attention layers in the pretrained LLM to aggregate temporal context over multiple segments of key frames. For this ablation, we encode each segment separately with $\mathcal{F}_{\text{key}}$ and concatenate the visual tokens from all segments as input into the LLM instead of average-pooling them. Mathematically, we formalize this operation as such:

$$z_{\text{final}} = \text{concat}\{\mathcal{F}_{\text{key}}(S_1), \cdots, \mathcal{F}_{\text{key}}(S_N)\}, \tag{3}$$

where concat$\{\}$ denotes the concatenation operation.

Trade-off between performance and efficiency. In addition to the performance on the EgoSchema benchmark, we also compare the performance and efficiency trade-offs between the different baselines in Table 3. We observe that the concatenation baseline not only performs worse at understanding long videos but is also the most computationally expensive variant with 15K GFLOPS. This is reasonable since we are computing the full self-attention operation over the extended sequence of video tokens in each layer of the base LLM. In contrast, while our Koala approach uses $\sim$1K GFLOPS more than the base, average and memory module baselines, it outperforms them by a significant margin of $\sim$6%.

Ablation over number of segments and frames per segment. In Figure 2, we study the effect of varying the number of video segments and frames within each segment during training. In general, we observe that increasing the number of frames per segment (Figure 2a and c) while reducing the number of segments (Figure 2b and d) is generally beneficial for long video understanding, as exemplified by the $\sim$1.5% increase in accuracy on procedure understanding when the number of frames per segment increases from 8 to 16 with 4 segments. The drop in accuracy with increasing segments may be due to redundant information factored into the temporal context aggregation.

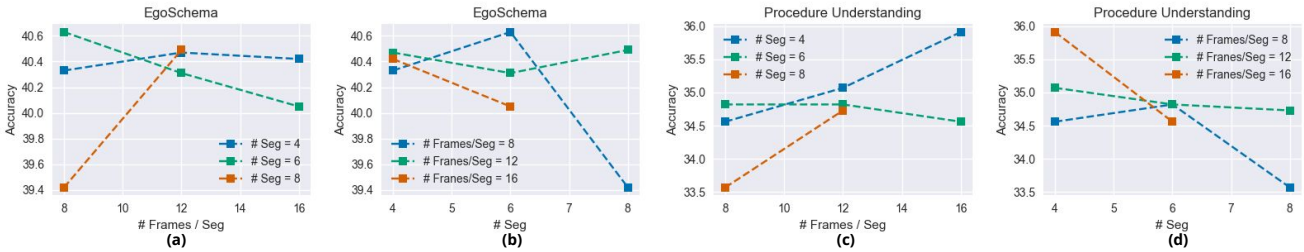


Figure 2. **Ablation over number of segments and frames.** Increasing the number of frames per segment while using a smaller number of segments during training is generally beneficial for long video understanding. We note that we run into an out-of-memory error with 8 segments of 16 frames each.

G. Additional qualitative visualizations

Visual examples of EgoSchema predictions. To gain insights into how our introduced spatiotemporal queries have helped improve the long-term temporal understanding capability of the frozen base vLLM, we provide several examples of correct predictions on the very challenging EgoSchema benchmark in Figure 4. Note that while EgoSchema is meant as a zero-shot evaluation benchmark, we use the subset of evaluation samples for which the correct answers are provided in these visualizations.

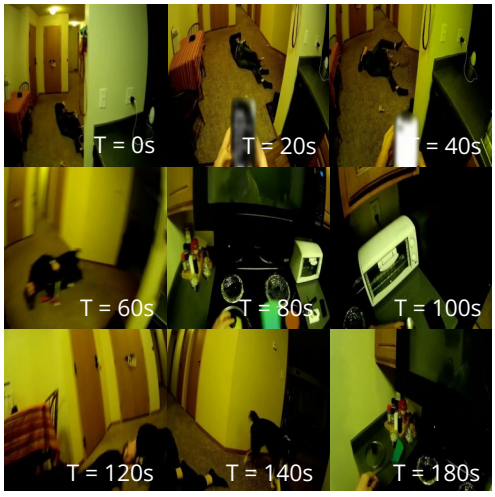
In Figures 4a and 4b, we see that the model often makes its predictions based on the first few input video frames and does not incorporate visual information from the entire videos, resulting in limited temporal context. In contrast, our approach is able to incorporate information over a larger time window, allowing it to summarize videos more accurately. Additionally, we also see using the spatiotemporal queries also encourage the base vLLM to hallucinate less visual details (Figures 4c and 4d), resulting in more accurate summarizations. Since it may be a little difficult to understand minutes-long videos from just a few select key frames, we have also attached the videos as part of the supplemental submission for reference.

Sample conversational generations. Using our final pretrained Koala model, we also provide qualitative visualizations of sample conversations with videos that are randomly downloaded from YouTube. In Figure 5, we observe that our Koala model is capable of reasoning about the contextual relationships between multiple short actions to infer reasonable summaries of long videos. For instance, we see that Koala is also able to explain the reasoning behind its predictions of making a nightstand and constructing a raised garden bed in Figure 5a and 5b, respectively. Additionally, we also provide examples of questioning our Koala vLLM about important details in long videos in Figure 6. We see that our vLLM is generally able to structure its responses using the correct temporal ordering of the observed actions.

? How would you summarize the primary objective the camera wearer is trying to achieve throughout the video in one sentence, considering the recurring drilling and removing actions they performed?

1. Currently, the camera wearer is attempting to carefully drill holes into the tile backer board's surface.
2. The camera wearer is attempting to carefully take off tile backer washers that are located on the tile backer board.
3. The camera wearer is trying to attach tile backer washers to the tile backer board.
4. The camera wearer is trying to clean the tile backer board.
5. Currently, the camera wearer is attempting to carefully paint the tile backer board surface.

(a) Example prediction 1



? What are the primary activities the camera wearer and the man are engaged in, and how do they differ in focus or involvement with the dog?

1. The camera wearer is making breakfast, while the man is playing with the dog.
2. Currently, the camera wearer is actively playing with the dog outdoors, while the man is occupied making breakfast inside.
3. Currently, the camera wearer and the man are both actively engaged in preparing breakfast together.
4. The camera wearer and the man are both playing with the dog.
5. Currently, the camera wearer is working diligently, while the man is happily playing with the dog nearby.

(b) Example prediction 2

? Can you summarize the overall purpose of the video and describe how the main character the camera wearer achieved their goal?

1 Currently, the camera wearer is attempting to construct a lawn mower from scratch.

2 Currently, the camera wearer is attempting to skillfully paint a lawn mower with precision.

3 The camera wearer is trying to mow the lawn.

4 Currently, the camera wearer is actively attempting to successfully sell a lawn mower to someone.

5 The camera wearer is trying to fix a lawn mower.

(c) Example prediction 3

? How would you describe the overall purpose of the camera wearer's actions in this video, and what are the key steps involved in achieving this purpose?

1 The camera wearer is preparing a meal.

2 Currently, the camera wearer is actively cleaning and tidying up the kitchen area.

3 Currently, the camera wearer is actively engaged in washing the dishes.

4 The camera wearer is taking a shower.

5 Currently, the camera wearer is in the process of getting dressed and ready.

(d) Example prediction 4

Figure 4. **Sample predictions on EgoSchema.** We provide some qualitative examples of predictions made by our proposed Koala approach and the base Video-Llama model on the very challenging long-term video understanding EgoSchema benchmark.

Ground-truth title: IKEA furniture assembly time lapse

1 minutes 45 seconds

Summarize what is happening in the video in a short sentence.

The people in the video are demonstrating the steps to make a simple nightstand from an ikea kit.

Can you describe the video in detail and explain why you think they are making a nightstand?

This video demonstrates the steps to make a simple nightstand from an ikea kit. The video begins with a man and woman standing on the floor next to a couch with the kit in their hands. They then assemble the nightstand with steps that include cutting pieces of wood, assembling and screwing them together, and attaching the shelf. The video then proceeds to show the final product and all steps of the assembly. It is likely that they are making a nightstand because they are holding the kit in their hands, and the pieces of the nightstand are made from wood.

(a) Video link: https://www.youtube.com/watch?v=TvmFKsmatbI

Ground-truth title: Pallet compost bins | Building our homestead

9 minutes 29 seconds

Summarize the overarching task the person is trying to achieve in the video in one sentence.

The overarching task is to build a raised garden bed.

Can you explain why you think the task is to build a raised garden bed and describe what actions the person performed to achieve this task in detail? Be as specific as possible.

Yes, the task is to build a raised garden bed. The person performed several actions to achieve this task. First, they dug a hole for the raised garden bed. Then, they built the frame of the raised garden bed using 2x4s and 2x6s. Next, they filled the base of the raised garden bed with soil. The person also used various tools such as a shovel, hammer, and drill to complete the task. They also used wood to build the frame and sides of the raised garden bed.

(b) Video link: https://www.youtube.com/watch?v=62hBgU8XzKo

Figure 5. Sample generations.

Ground-truth title: Easy vegan poutine with mushroom gravy

5 minutes 43 seconds

Describe in detail the actions that the person in the video took to cook poutine. Be as specific as possible.

The woman in the video made poutine using a recipe from the website allrecipes.com. The recipe calls for french fries, cheese curds, and gravy. The woman followed a recipe and made the poutine in a pot on the stovetop. She started by making the gravy and then added the cheese curds and french fries. She cooked the poutine until the cheese curds were melted.'

(a) Video link: https://www.youtube.com/watch?v=T33BkvAkctY

Ground-truth title: Small bird feeder / Using plastic water bottle / Bird feeder technology

6 minutes 08 seconds

Explain the key actions that the person takes to make the bird feeder. Be as specific as possible.

The person makes a bird feeder using a bird feeder mold, bird seed, scissors, a plastic bottle, and a twist tie. The person first cuts a plastic bottle into a bird feeder shape. Then, the person uses the plastic bottle as the base of the bird feeder and fills it with bird seed. Finally, the person ties the bird feeder up with a twist tie.

(b) Video link: https://www.youtube.com/watch?v=0jRg9DRDnrU

Figure 6. Sample generations.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 2

[2] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European Conference on Computer Vision*, pages 640–658. Springer, 2022. 4

[3] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *arXiv preprint arXiv:2304.14108*, 2023. 1

[4] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023. 2

[5] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

[6] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *arXiv preprint arXiv:2308.09126*, 2023. 2

[7] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640, 2019. 1

[8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1

[9] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Xun Guo, Tian Ye, Yan Lu, Jenq-Neng Hwang, et al. Moviechat: From dense token to sparse memory for long video understanding. *arXiv preprint arXiv:2307.16449*, 2023. 4

[10] Jianlin Su. Bert position encoding. https://kexue.fm/archives/7947, 2020. 4

[11] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9777–9786, 2021. 2

[12] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 2

[13] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 2