VideoDescriptor: Video Understanding with Textual Descriptions

Sinclair Hudson

December 31, 2023

Abstract

While LLMs are now frequently extended to process visual information in the form of images, they are not yet commonly used to process video. This work explores a general pipeline that leverages large language models (LLMs) to convert video into textual descriptions, and then further retrieve clips relevant to a query using those textual descriptions. The pipeline is called the VideoDescriptor pipeline, and is evaluated on text-to-video retrieval as well as video summarization. While not as accurate as other methods, the VideoDescriptor pipeline is able to achieve reasonable results on both tasks, and is completely zero-shot. Code for all experiments is available at https://github.com/SinclairHudson/video-understanding.

1 Introduction

The performance of language models has surged in recent years, largely thanks to the predictable scaling of transformer-based models. In addition, multimodal vision-language models have been developed, creating shared latent representations between images and text [RKH+21] [YWV+22] [KBFT19]. These LLMs, with the ability to understand both images and text, can now perform a lot of traditional benchmark tasks "zero-shot", after being trained on vast amounts of semi-structured internet data [OA+23] [RKH+21] [TAB+23] [ADL+22]. Despite their success in understanding images, LLMs have not yet shown the same success in understanding video. Popular video understanding benchmarks such as MSR-VTT [XMYR16], MSVD [CD11], and ActivityNet [FCHN15] are still largely dominated by architectures custom-built for video understanding. While these architectures often use the same *techniques* of self-supervised pre-training and transfer learning, they do not use the same LLMs that have been so successful in image understanding. This work proposes a general pipeline for accomplishing video understanding tasks using multimodal LLMs, by converting visual information into text and then completing the analog task in the text domain.

2 Related Work

2.1 Video Understanding

Video understanding is a difficult machine learning task, for many reasons. The data is very high-dimensional and yet semantically sparse; most frames and pixels are redundant when analysing the video for content. Additionally, video data is extremely time-consuming to annotate. As such, very few video datasets exist, and are often much smaller than analogous image datasets [RDS+15] [LMB+15]. Nevertheless, videos are ubiquitous in digital life, on popular social media websites like YouTube and TikTok. The research domain of video understanding is active and diverse. Below, a few of the most relevant works are briefly introduced.

CLIP4Clip is a method that aims to extend the ideas of CLIP [RKH+21] to the video domain [LJZ+22]. It generally follows a bi-encoder structure, in which the video and text are encoded using separate transformers [VSP+23]. The video is split into different patches spatially, and each patch is encoded using a linear layer into an embedding vector, and then processed by the video encoder transformer. The text is tokenized and then processed by the text encoder transformer. Then, both the video and text latent representations are fed into a similarity calculator module. The system is trained end-to-end with video-text pairs and contrastive loss, as in CLIP [RKH+21]. CLIP4CLip is naturally suited for video retrieval, since it can compute a

similarity score between a video and a text query. Given a text query, the system computes the similarity between the query and each video in the dataset, and retrieves the videos in order of similarity.

X-CLIP takes a similar approach to CLIP4Clip, using contrastive pre-training to learn associations between video and textual descriptions [MXS+22]. However, X-CLIP goes further and models more fine-grained associations between video and text. For a given video-caption pair, the video is encoded frame-by-frame are then passed into a temporal encoder, producing both a vector for each frame and a vector for the entire video. Likewise, the caption is encoded using a transformer, producing both a vector for the entire caption and a vector for each word. X-CLIP explicitly models caption-video, caption-frame, word-video, and word-frame relationships, and uses these associations to learn very detailed representations of the video and text. These representations, along with task-specific fine-tuning, allow X-CLIP to perform very well on text-to-video and video-to-text retrieval tasks.

InternVideo is a recent attempt at a "foundation model" for video, being able to complete numerous downstream tasks. The authors train InternVideo with a combination of multimodal contrastive learning (as in CLIP [RKH+21]), as well as masked video reconstruction, inspired by VideoMAE [TSWW22]. The InternVideo video encoder is pretrained on an unprecedently large dataset of internet videos and movies, allowing it to learn very semantically rich representations of videos. This dataset is compiled by the authors using both pre-existing video datasets as well as videos scraped from the internet. In total, the dataset contains 12 million videos from 5 different domains [WLL+22]. With additional fine-tuning on specific tasks, InternVideo achieves state-of-the-art performance on many action understanding, video-language alignment, and video open understanding benchmarks [WLL+22]. Note that InternVideo is *not* a large language model, though it uses the same strategy of large-scale, self-supervised pretraining.

2.2 Large Language Models for Vision

As mentioned in the introduction, it is now common to see LLMs that can process visual information in the form of images. Below are a few of the most relevant works that were designed, in part, to process multiple images at once.

Flamingo is a family of "Visual Language Models", otherwise known as multimodal LLMs [ADL+22]. In Flamingo models, image embeddings from an image encoder (ResNet without normalizers [BDSS21]) are introduced at every layer of a transformer-based language model. The image encoder and language model are frozen, and only the cross-attention modules connecting the image features to the language features are trained for Flamingo. This significantly reduces the computation required to train the system. Flamingo is trained on many datasets, including image-text and video-text datasets. It is capable of taking as input multiple images or video frames. The authors note that adding relevant frames to the input incrementally improves performance on certain tasks, up to around 32 frames [ADL+22]. This indicates that Flamingo is able to gather and aggregate useful features from many images simultaneously.

Like Flamingo, LLaVA [LLWL23] is a large language model trained on a combination of text-image and text-video data. It is built upon the Vicuna language model [CLL+23], and incorporates the visual encoder of CLIP [RKH+21] to encode images into embeddings that the language model can take as input. It is fine-tuned end-to-end on instructional image-text data, and has proven to be exceptionally strong at visual question-answering tasks. Due to the whole model being optimized end-to-end, LLaVA outperforms Flamingo on many tasks, but is also much more computationally expensive to train. LLaVA is used in this work as the LLM for describing videos.

2.3 Large Language Models for Video

Most related to this work are attempts to leverage large language models for video understanding tasks.

VideoChat [LHW+23] focuses on an understanding of video that is amenable to multiple-round video question answering. VideoChat has two streams; VideoChat-Embed and VideoChat-Text. VideoChat-Embed uses InternVideo to encode the video into a semantically meaningful vector representation. VideoChat-Text uses multimodal language models to represent segments of the video as textual descriptions. These textual descriptions, in addition to the embedding from VideoChat-Embed, are fed into an LLM as context for question answering.

Video-ChatGPT [MRKK23] adapts LLaVA to process video instead of images. The Video-ChatGPT pipeline can be broken down into 3 sequential modules: the visual encoder, the video embedding module, and the LLM. Only the video embedding module is trained from scratch; the upstream visual encoder and downstream LLM remain frozen. As such, Video-ChatGPT can be seen as a parameter-efficient fine-tuning approach to equip an LLM model with a visual encoder; the video embedding module adapts the visual encoder to video. In their approach, the video embedding module embeds both frame-wise and spatial-patch-wise, allowing for the model to learn both temporal and spatial features. Like VideoChat, Video-ChatGPT is largely designed for video question answering tasks and chat-like interactions.

2.4 Video-Text datasets

MSR-VTT (Microsoft Research Video to Text) is a flagship video understanding dataset [XMYR16]. This dataset can be used for multiple tasks, including video question answering, video retrieval, and video captioning. Every video in the dataset has multiple captions, each written by a different human annotator after watching the short video. The dataset contains 10 000 videos, sourced from the internet by downloading video results of popular internet search queries. For the video retrieval task, 1000 queries (video captions) and videos are used for testing, with each query specifying exactly one video as the correct answer [YKK18]. Thus, the retriever must find the correct video from a pool of 1000 candidates. The videos in the video retrieval split are 14 seconds long on average, see Figure 1 for a histogram of video lengths.

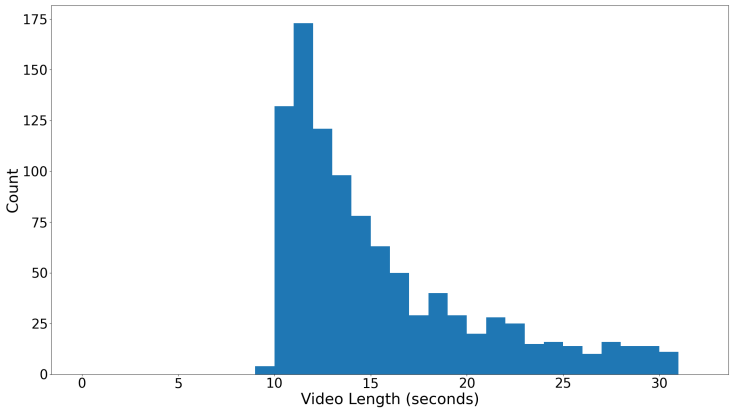

Figure 1: Length of videos in the MSR-VTT retrieval data split.

While not used in this paper, other datasets like MSVD [CD11] and ActivityNet [FCHN15] are also popular for video understanding tasks. Microsoft Research Video Description Corpus (MSVD) is similar to MSR-VTT, consisting of short video clips and associated descriptions of the actions in the clip. MSVD can be used for video captioning, video question answering, and video retrieval. ActiviyNet is an extremely large dataset used for action classification (human activity understanding). Given a video, the task is to classify the action being performed in the video, from a set of 200 classes. ActivityNet version 1.3 contains 19994 videos from YouTube, making it one of the largest video datasets available.

3 Method

The proposed VideoDescriptor pipeline is designed in such a way to be general to different video understanding tasks, with each of the 4 major steps allowing for multiple methods, or being optional for some tasks. The VideoDescriptor pipeline is similar to VideoChat-Text of VideoChat [LHW+23]; the strategy is to represent video as a sequence of textual descriptions, which are then used downstream. For the purposes of this work, a "clip" is a small segment of a video that is a few seconds long, and is highly cohesive in what it depicts. For example, in a movie, a clip might be a single shot. At a high level, the pipeline is as follows:

1. Partition the video into individual clips.
2. From each clip, select a small subset of frames to represent the clip.
3. Using the selected frames, generate a textual description for each clip using an LLM.
4. Using the textual descriptions, retrieve clips relevant to a question or query.

For brevity, these steps will be referred to as "Clip partitioning", "Frame selection", "Clip description", and "Clip retrieval" respectively. The two most applicable tasks for this pipeline are video retrieval and visual video summarization. In video retrieval, given a query, the goal is to retrieve the video in a video dataset that best matches the query. In visual video summarization, the goal is to edit a video down to a shorter length, containing only clips relevant to a given query.

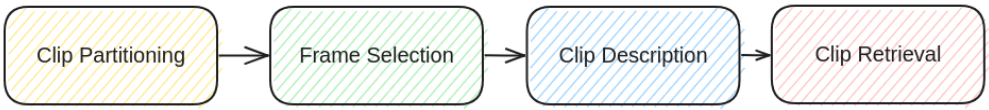

Figure 2: High level block diagram of the proposed video understanding pipeline.

3.1 Clip Partitioning

Videos can contain a lot of different clips, which may or may not be related. As such, it's desirable to process and describe each clip individually; clips are an atomic unit of video in this approach. The input for clip partitioning is the whole video, and the output is a list of breaks (frame numbers) between clips. The difficult aspect of clip partitioning is to *efficiently* find the boundaries between clips. A two-hour video at 24 frames per second contains 172,800 frames, and so processing each one is computationally expensive and often infeasible.

The simplest approach to clip partitioning is to simply partition the video into equal-sized clips, without regard for the content of the video. This approach is simple and fast to compute, but the resulting clips are not necessarily cohesive, since the resulting segments could contain multiple scenes and shots.

3.1.1 Coarse-to-fine clip partitioning

A more strategic approach identifies breaks as sequential frames with a large difference between them. However, even computing a simple difference such as an L1 or L2 norm between each pair of frames is computationally expensive, since a norm must be taken per frame in the video.

As such, the VideoDescriptor pipeline uses a coarse-to-fine approach to clip partitioning. The algorithm starts by computing the L1 distance between frames *1 second apart*, over the entire video. If the L1 distance between these frames is above a certain threshold, then its likely that there's a clip break between them. Thus, the 1-second segment is evaluated frame by frame, and if the distance between two frames is above the threshold, then a clip break is identified. Empirically, this approach saves 60% of the L1 computations compared to the naive frame-by-frame approach, and could save more if fewer coarse breaks are identified.

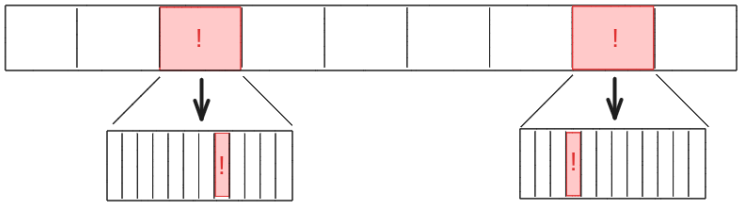

Figure 3: Coarse-to-fine clip partitioning. Vertical lines are frames in the video, and red boxes are identified clip breaks.

Figure 4: Pairs of sequential frames, separated by a clip break. Clip breaks are determined by the coarse-to-fine clip partitioning strategy.

See Figure 4 for resulting clip breaks from the coarse-to-fine approach. Qualitative results show that this approach is very effective at identifying clip breaks, with few false positives and false negatives.

3.2 Frame Selection

Even within a single clip, there are a lot of frames, most of which are very similar to their neighbours. To make the pipeline efficient, it's critical to select a subset of the most semantically relevant frames. The input to frame selection is a clip (small segment of video) and the output is a list of frames selected for further processing. Earlier works simply sample randomly [LLZ+21] or sample uniformly, at a coarse frame rate such as 1 frame per second [LJZ+22]. This work explores 4 different frame selection strategies: stratified selection, random sampling, greedy L1 selection, and stratified triplet selection.

3.2.1 Stratified Selection

To get a diverse set of frames, the first, last and middle frames of a clip are selected for stratified selection. For brevity, this sampling method is referred to as `3strat`. `5strat` is also explored, which selects the first frame, last frame, and 3 evenly spaced frames in between.

3.2.2 Random Sampling

A random sampling approach is also explored, with either 3 or 5 frames selected from the clip (with replacement). These are referred to as `3rand` and `5rand` respectively.

3.2.3 Greedy L1 Selection

The idea for greedy selection is to skip frames that are visually similar to the previously selected frame. Greedy L1 selection initially selects the first frame of a clip. Then, it selects an additional frame if the L1 distance between the new frame and the previously selected frame is greater than some threshold. The L1

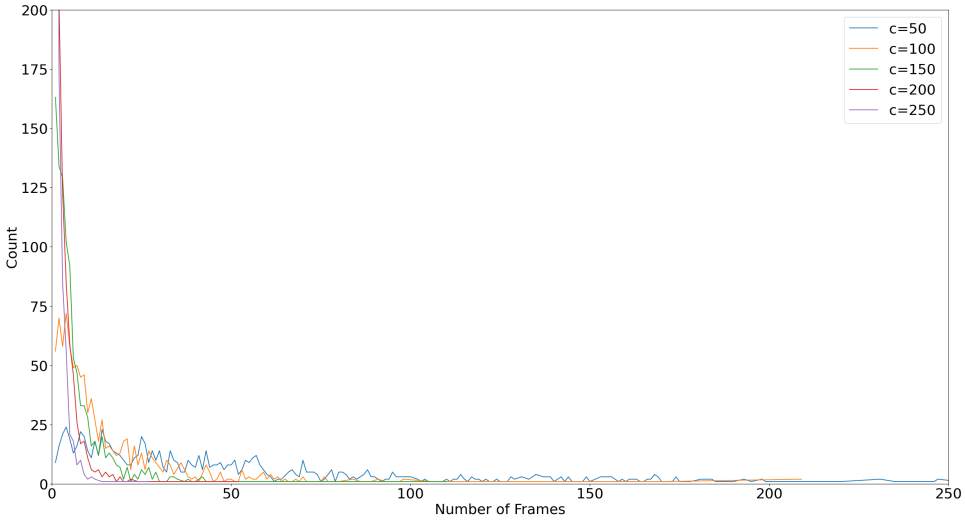

Figure 5: Number of frames selected per video in the MSR-VTT dataset retrieval split when using greedy L1 selection, for different thresholds.

distance is normalized by the number of pixels in the image, so that the threshold is independent of the video resolution. More formally, the current frame I_c is selected if:

$$c < \frac{|I_p - I_c|}{W \times H \times C} \tag{1}$$

where c is a pre-defined threshold, $|\cdot|$ is L1 distance, I_p is the previously selected frame, and W, H, C are the width, height, and channels of the frames, respectively. In the MSR-VTT dataset, most videos are quite short (see Figure 1), and often only contain a single static shot. As such, when using greedy L1 selection, a substantial number of videos only have the first frame selected (with L1 threshold 180). L1 distance was chosen over L2 because L2 is very sensitive to large changes in a few pixels; L1 gives a better sense of the overall change in the image.

3.2.4 Stratified Triplet Selection

It's possible that single frames at multiple points in the clip are not enough to capture a clip's dynamic content. For this reason, this work also explores stratified triplet selection, where at each stratified point in the clip, 3 frames are selected across 1 second. The motivation for this method is that the 3 frames in a single second will show what is changing in the scene, capturing the action in that second. Each triplet is fed in as a single input into the LLM during clip description. As a result, there is only one textual description per triplet when using this selection method.

3.3 Clip Description

The input to clip description is a prompt and multiple frames, and the output is a text description. The multimodal LLM used in testing this pipeline is LLaVA [LLWL23]. LLaVA is selected because it is free and open source, and can be run on a single consumer GPU, making it amenable to experimentation. While LLaVA can take multiple images as input, it is rarely tested and on multi-image inputs. So, apart from the stratified triplet selection experiments, most clip descriptions are the concatenation of the descriptions of individual frames.

Two prompts are tried in this work, one of which is is dubbed "concise" (C), while the other is "verbose" (V). The concise prompt is "Please describe the objects in this image." The verbose prompt is "Please describe the objects in this image. Be as descriptive as possible." The third option used for clip description is dubbed "temperature" (T), which uses the verbose prompt with a sampling temperature of 0.2. Two

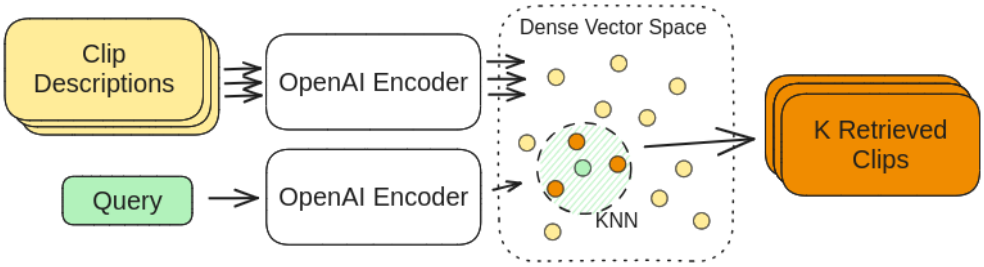

Figure 6: Bi-Encoder retrieval strategy, using OpenAI embedding API.

different responses are generated (using beam search) for each frame and concatenated when using the "temperature" option. In all cases, the caption for each frame (or triplet) is limited to 512 words, to bound the amount of computation required per frame.

3.4 Clip Retrieval

The last aspect of of the pipeline is, of course, retrieval. The input to clip retrieval is a query and the clip descriptions, and the output is a list of clips sorted by relevance. Because LLMs have described the content of the videos at this point in the pipeline, the retrieval task is reduced to that of text retrieval. The query is given, and the documents are the descriptions of the clips produced in the previous step. In this work, we explore two retrieval strategies: BM25 and Bi-Encoder, using OpenAI embeddings. BM25 is a highly performant sparse retriever, relying on term frequencies and other statistics to rank documents [RWJ+95]. It is a very fast baseline, retrieving from a thousand documents in milliseconds.

3.4.1 Bi-Encoder Retrieval

The Bi-Encoder approach is heavily inspired by DPR [KOM+20], but uses the same encoder for both documents and queries, and uses L2 distance as a similarity metric instead of dot product. At a high level, all the documents are encoded into a vector space using an LLM encoder, and the query is encoded into the same vector space using the same encoder. Then, the similarity between the query and each document is computed using L2 distance, and the top-k documents are returned in a retrieval. While different in application, it's very similar to the bi-encoder strategy used in the BLINK zero-shot entity linker [LW20]. See Figure 6 for a visual representation of the bi-encoder retrieval strategy. The encoder used to generate embedding vectors is the OpenAI model `text-embedding-ada-002`, which outputs 1536-dimensional vectors. The KNN is performed using the FAISS library [JDJ19], and all clip descriptions are embedded and stored in an index beforehand, making querying for nearest neighbours very fast.

4 Results

4.1 Video Retrieval

The VideoDescriptor pipeline is compared against other models in Table 1. Different configurations of the VideoDescriptor pipeline are compared in Table 2. For all MSR-VTT experiments, the videos are considered to be a single clip and thus clip partitioning is not used in the VideoDescriptor pipeline. Also note that the VideoDescriptor pipeline functions zero-shot, while all other models have been finetuned for text-to-video retrieval using the MSR-VTT training splits.

4.2 Qualitative Results on Video Summarization

For visual video summarization, the task is to select a subset of clips from a video relevant to the query, and edit them together into a shorter video. Sports games are desireable to summarize because they are long, with a few well-defined interesting events (goals, fouls, etc.). To test this method, a hockey game

Approach	FS	Prompt	Triple	**MSR-VTT** T2V		
				R@1	**R@5**	**R@10**
CLIP4Clip [LJZ+22]	-	-	-	0.445	0.714	0.816
X-CLIP [MXS+22]	-	-	-	0.493	0.758	0.848
InternVideo [WLL+22]	-	-	-	0.552	-	-
VideoDescriptor + BiEnc	5strat	C		0.293	0.531	0.642
VideoDescriptor + BM25	5strat	T		0.176	0.364	0.446

Table 1: Performance comparison of the best VideoDescriptor configurations against other models on MSR-VTT T2V retrieval.

Approach	Frame Sampling	Prompt	Triplet?	**MSR-VTT** T2V		
				R@1	**R@5**	**R@10**
VideoDescriptor + BiEnc	L1 greedy	V		0.215	0.439	0.544
VideoDescriptor + BiEnc	3strat	V		0.260	0.505	0.604
VideoDescriptor + BiEnc	3strat	C		0.276	0.494	0.598
VideoDescriptor + BiEnc	3strat	T		0.265	0.480	0.590
VideoDescriptor + BiEnc	3strat	V	✓	0.269	0.505	0.604
VideoDescriptor + BiEnc	3rand	V		0.282	0.490	0.596
VideoDescriptor + BiEnc	5strat	V		0.290	0.516	0.625
VideoDescriptor + BiEnc	5strat	C		**0.293**	**0.531**	**0.642**
VideoDescriptor + BiEnc	5strat	T		0.269	0.481	0.593
VideoDescriptor + BiEnc	5strat	V	✓	0.286	0.530	0.635
VideoDescriptor + BiEnc	5rand	V		0.272	0.476	0.604
VideoDescriptor + BM25	L1 greedy	V		0.125	0.266	0.341
VideoDescriptor + BM25	3strat	V		0.134	0.288	0.356
VideoDescriptor + BM25	3strat	C		0.138	0.284	0.366
VideoDescriptor + BM25	3strat	T		0.157	0.323	0.408
VideoDescriptor + BM25	3strat	V	✓	0.122	0.277	0.347
VideoDescriptor + BM25	3rand	V		0.142	0.280	0.340
VideoDescriptor + BM25	5strat	V		0.167	0.326	0.411
VideoDescriptor + BM25	5strat	C		0.150	0.331	0.417
VideoDescriptor + BM25	5strat	T		**0.176**	**0.364**	**0.446**
VideoDescriptor + BM25	5strat	V	✓	0.151	0.328	0.395
VideoDescriptor + BM25	5rand	V		0.149	0.294	0.368

Table 2: Performance of many VideoDescriptor configurations on MSR-VTT T2V retrieval.

and a soccer game were partitioned into clips, with each clip being described using `3strat` sampling. Each clip is described by the LLM using the prompt "Please describe what is going on in this image". Then, given a query, the top 30 clips were retrieved using the Bi-Encoder and edited together in order of relevance and rendered into a single video. The results can be found in the GitHub repository for download: https://github.com/SinclairHudson/video-understanding/tree/main/video_summaries.

Both the soccer game and hockey game were appproximately 2 hours long, and each video ended up being partitioned into over 700 clips. The clip partitioning at times struggles with large fast-moving, high contrast objects, which create large inter-frame differences and trigger a clip break.

During retrieval, the VideoDescriptor pipeline struggles to identify semantically meaningful clips, but is able to identify relevant objects easily. A manual inspection of the descriptions shows that the model is able to describe the frames of the video well, but doesn't get specific about the actions in the scene. For example, it seems easier for the pipeline to retrieve prompts like "goaltender", "referee", or "coach", and harder to retrieve prompts like "goal" or "penalty". Identifying clips with referees or goaltenders is easy, since they are visually distinct from the rest of the players and easy to identify in an image.

From a runtime perspective, clip partitioning, description, and indexing of a 2-hour sports game takes approximately 4 hours, using an NVIDIA GeForce RTX 3090 GPU for LLM inference. However, the retrieval step using the FAISS library is very fast, taking less than 170ms to retrieve the top 30 clips from the index. With more efficient clip partitioning and description, this pipeline could be used to understand videos in practical applications involving video search.

5 Discussion

5.1 Clip Partitioning

Qualitatively, the proposed clip partitioning approach struggles when large objects move quickly in the video, resulting in large differences between frames. Additionally, sudden changes in the video, like the flash of a camera, are also often falsely identified as a clip break due to the sudden difference in image brightness. Finally, slow transitions and fade transitions are often not detected as clip breaks because they gradually change from one clip to another. Animated transitions found in traditional sports broadcasts can cause such issues. Those are the 3 most common failure modes of the L1-based clip partitioning algorithm. In general, false positives will increase the number of clips, and fragment true clips into multiple clips. While false positives will slow down the retrieval part of the pipeline, some amount of false positives may be acceptable, depending on the application. False negatives are potentially more damaging, creating a segment that is really two clips. This could potentially introduce unrelated clips into a downstream video summary, for example. Using more sophisticated clip partitioning algorithms such as optical flow or scene change detection may help this part of the pipeline.

5.2 Video Retrieval

Below is a discussion of the results on the MSR-VTT retrieval dataset; relevant results can be found in Table 2. First and foremost, it would seem that the Bi-Encoder retriever performs better than BM25 for this retrieval task, especially for Recall@1. For frame selection, `5strat` is slightly more performant when compared to equivalent `3strat` experiments, indicating that the extra frames evenly spaced throughout the video are helpful for understanding the video. The random sampling doesn't perform significantly worse than stratified selection, contrary to expectation. This could be due to the fact that the videos in MSR-VTT are relatively short, and so all frame selection methods may return very similar sets of frames for each video. These videos often only contain one clip, so in this case random sampling isn't prone to catastrophically missing a scene in the video.

For prompt choice, it would seem that this task is relatively insensitive to prompts. The differences between the concise (C) experiments and equivalent verbose (V) ones are very small, though the BM25 retriever seems to benefit from the verbose prompt, while the Bi-Encoder does better with the concise prompt. We also see a significant improvement for the stochastic temperature prompting method (T), when using the BM25 retriever. Intuitively this makes sense; a higher sampling temperature will produce a

more diverse set of words in the description, which will help tf-idf-based retrievers like BM25 find the right document.

Interestingly, the experiments that use stratified triplet selection don't perform notably better than the experiments that use single frames in video retrieval. This seems to suggest that either the additional frames are redundant, or that LLaVA is not good at incorporating information from multiple images in a single generation step. This is similar to the findings in CLIP4clip, where the authors find that processing frames in batches as 3D tensors doesn't improve performance over frame-by-frame processing [LJZ+22]. The authors of VideoChat also note that their similar VideoChat-Text pipeline struggled to understand "intricate temporal reasoning and causal inference".

6 Future work

First and foremost, expanding to other datasets is critical to fully examine the generalization capability of the pipeline. The current pipeline is currently very slow, requiring many forward passes of a large language model for every frame, and multiple frames for every clip. It is experimental in nature, and has not been optimized for runtime performance. With additional engineering effort, the performance of the video processing and text processing elements of the pipeline could both be greatly improved. Future work could investigate further minimizing the number of frames processed, and prompting techniques to generate very dense captions, with a lot of information content in just a few tokens.

While this work focused on the visual features of video, audio is also often attached to video, and thus could be used to further improve the performance of the pipeline. Related work such as VideoLLaMA has shown that audio can provide a small boost to performance [ZLB23]. Incorporating audio would be particularly important for understanding videos that have an emphasis on dialogue, such as movies. If longer videos are to be processed, it would also be beneficial to explore more sophisticated clip partitioning algorithms, as mentioned in the discussion. While not as important for performance on benchmark datasets, properly partitioning a long video into clips is a critical step for understanding videos in the real world. Since the VideoDescriptor pipeline has 4 largely independent stages and many hyperparameters, there are many avenues for future work.

7 Conclusion

In this paper, the VideoDescriptor pipeline is proposed, consisting of 4 steps to process video into textual descriptions, and then retrieve clips relevant to a query. While supervised methods still perform much better than the VideoDescriptor pipeline on text-to-video retrieval, the pipeline shows potential, especially for a zero-shot method. Additionally, it can be used for video summarization, and generates compelling video summaries for queries relating to specific objects in the video. The 4 main steps of the pipeline are all relatively simple, requiring no task-specific training, and can be improved upon or modified in future work. Finally, since the VideoDescriptor pipeline is predominantly reliant on LLMs, it will likely benefit from future improvement of multimodal language models.

References

[ADL+22] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022.

[BDSS21] Andrew Brock, Soham De, Samuel L. Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization, 2021.

[CD11] David L. Chen and William B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL-2011)*, Portland, OR, June 2011.

[CLL+23] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.

[FCHN15] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.

[JDJ19] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[KBFT19] Douwe Kiela, Suvrat Bhooshan, Hamed Firooz, and Davide Testuggine. Supervised multimodal bitransformers for classifying images and text. *arXiv preprint arXiv:1909.02950*, 2019.

[KOM+20] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, November 2020. Association for Computational Linguistics.

[LHW+23] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding, 2023.

[LJZ+22] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 508:293–304, 2022.

[LLWL23] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

[LLZ+21] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling, 2021.

[LMB+15] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[LW20] Martin Josifoski Sebastian Riedel Luke Zettlemoyer Ledell Wu, Fabio Petroni. Zero-shot entity linking with dense entity retrieval. In *EMNLP*, 2020.

[MRKK23] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models, 2023.

[MXS+22] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval, 2022.

[OA+23] OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning,

Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2023.

[RDS+15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015.

[RKH+21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021.

[RWJ+95] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. Okapi at trec-3. *Nist Special Publication Sp*, 109:109, 1995.

[TAB+23] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub,

Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, James Lottes, Nathan Schucher, Federico Lebron, Alban Rrustemi, Natalie Clay, Phil Crone, Tomas Kocisky, Jeffrey Zhao, Bartek Perz, Dian Yu, Heidi Howard, Adam Bloniarz, Jack W. Rae, Han Lu, Laurent Sifre, Marcello Maggioni, Fred Alcober, Dan Garrette, Megan Barnes, Shantanu Thakoor, Jacob Austin, Gabriel Barth-Maron, William Wong, Rishabh Joshi, Rahma Chaabouni, Deeni Fatiha, Arun Ahuja, Ruibo Liu, Yunxuan Li, Sarah Cogan, Jeremy Chen, Chao Jia, Chenjie Gu, Qiao Zhang, Jordan Grimstad, Ale Jakse Hartman, Martin Chadwick, Gaurav Singh Tomar, Xavier Garcia, Evan Senter, Emanuel Taropa, Thanumalayan Sankaranarayana Pillai, Jacob Devlin, Michael Laskin, Diego de Las Casas, Dasha Valter, Connie Tao, Lorenzo Blanco, Adrià Puigdomènech Badia, David Reitter, Mianna Chen, Jenny Brennan, Clara Rivera, Sergey Brin, Shariq Iqbal, Gabriela Surita, Jane Labanowski, Abhi Rao, Stephanie Winkler, Emilio Parisotto, Yiming Gu, Kate Olszewska, Yujing Zhang, Ravi Addanki, Antoine Miech, Annie Louis, Laurent El Shafey, Denis Teplyashin, Geoff Brown, Elliot Catt, Nithya Attaluri, Jan Balaguer, Jackie Xiang, Pidong Wang, Zoe Ashwood, Anton Briukhov, Albert Webson, Sanjay Ganapathy, Smit Sanghavi, Ajay Kannan, Ming-Wei Chang, Axel Stjerngren, Josip Djolonga, Yuting Sun, Ankur Bapna, Matthew Aitchison, Pedram Pejman, Henryk Michalewski, Tianhe Yu, Cindy Wang, Juliette Love, Junwhan Ahn, Dawn Bloxwich, Kehang Han, Peter Humphreys, Thibault Sellam, James Bradbury, Varun Godbole, Sina Samangooei, Bogdan Damoc, Alex Kaskasoli, Sébastien M. R. Arnold, Vijay Vasudevan, Shubham Agrawal, Jason Riesa, Dmitry Lepikhin, Richard Tanburn, Srivatsan Srinivasan, Hyeontaek Lim, Sarah Hodkinson, Pranav Shyam, Johan Ferret, Steven Hand, Ankush Garg, Tom Le Paine, Jian Li, Yujia Li, Minh Giang, Alexander Neitz, Zaheer Abbas, Sarah York, Machel Reid, Elizabeth Cole, Aakanksha Chowdhery, Dipanjan Das, Dominika Rogozińska, Vitaly Nikolaev, Pablo Sprechmann, Zachary Nado, Lukas Zilka, Flavien Prost, Luheng He, Marianne Monteiro, Gaurav Mishra, Chris Welty, Josh Newlan, Dawei Jia, Miltiadis Allamanis, Clara Huiyi Hu, Raoul de Liedekerke, Justin Gilmer, Carl Saroufim, Shruti Rijhwani, Shaobo Hou, Disha Shrivastava, Anirudh Baddepudi, Alex Goldin, Adnan Ozturel, Albin Cassirer, Yunhan Xu, Daniel Sohn, Devendra Sachan, Reinald Kim Amplayo, Craig Swanson, Dessie Petrova, Shashi Narayan, Arthur Guez, Siddhartha Brahma, Jessica Landon, Miteyan Patel, Ruizhe Zhao, Kevin Villela, Luyu Wang, Wenhao Jia, Matthew Rahtz, Mai Giménez, Legg Yeung, Hanzhao Lin, James Keeling, Petko Georgiev, Diana Mincu, Boxi Wu, Salem Haykal, Rachel Saputro, Kiran Vodrahalli, James Qin, Zeynep Cankara, Abhanshu Sharma, Nick Fernando, Will Hawkins, Behnam Neyshabur, Solomon Kim, Adrian Hutter, Priyanka Agrawal, Alex Castro-Ros, George van den Driessche, Tao Wang, Fan Yang, Shuo yiin Chang, Paul Komarek, Ross McIlroy, Mario Lučić, Guodong Zhang, Wael Farhan, Michael Sharman, Paul Natsev, Paul Michel, Yong Cheng, Yamini Bansal, Siyuan Qiao, Kris Cao, Siamak Shakeri, Christina Butterfield, Justin Chung, Paul Kishan Rubenstein, Shivani Agrawal, Arthur Mensch, Kedar Soparkar, Karel Lenc, Timothy Chung, Aedan Pope, Loren Maggiore, Jackie Kay, Priya Jhakra, Shibo Wang, Joshua Maynez, Mary Phuong, Taylor Tobin, Andrea Tacchetti, Maja Trebacz, Kevin Robinson, Yash Katariya, Sebastian Riedel, Paige Bailey, Kefan Xiao, Nimesh Ghelani, Lora Aroyo, Ambrose Slone, Neil Houlsby, Xuehan Xiong, Zhen Yang, Elena Gribovskaya, Jonas Adler, Mateo Wirth, Lisa Lee, Music Li, Thais Kagohara, Jay Pavagadhi, Sophie Bridgers, Anna Bortsova, Sanjay Ghemawat, Zafarali Ahmed, Tianqi Liu, Richard Powell, Vijay Bolina, Mariko Iinuma, Polina Zablotskaia, James Besley, Da-Woon Chung, Timothy Dozat, Ramona Comanescu, Xiance Si, Jeremy Greer, Guolong Su, Martin Polacek, Raphaël Lopez Kaufman, Simon Tokumine, Hexiang Hu, Elena Buchatskaya, Yingjie Miao, Mohamed Elhawaty, Aditya Siddhant, Nenad Tomasev, Jinwei Xing, Christina Greer, Helen Miller, Shereen Ashraf, Aurko Roy, Zizhao Zhang, Ada Ma, Angelos Filos, Milos Besta, Rory Blevins, Ted Klimenko, Chih-Kuan Yeh, Soravit Changpinyo, Jiaqi Mu, Oscar Chang, Mantas Pajarskas, Carrie Muir, Vered Cohen, Charline Le Lan, Krishna Haridasan, Amit Marathe, Steven Hansen, Sholto Douglas, Rajkumar Samuel, Mingqiu Wang, Sophia Austin, Chang Lan, Jiepu Jiang, Justin Chiu, Jaime Alonso Lorenzo, Lars Lowe Sjösund, Sébastien Cevey, Zach Gleicher, Thi Avrahami,

Anudhyan Boral, Hansa Srinivasan, Vittorio Selo, Rhys May, Konstantinos Aisopos, Léonard Hussenot, Livio Baldini Soares, Kate Baumli, Michael B. Chang, Adrià Recasens, Ben Caine, Alexander Pritzel, Filip Pavetic, Fabio Pardo, Anita Gergely, Justin Frye, Vinay Ramasesh, Dan Horgan, Kartikeya Badola, Nora Kassner, Subhrajit Roy, Ethan Dyer, Víctor Campos, Alex Tomala, Yunhao Tang, Dalia El Badawy, Elspeth White, Basil Mustafa, Oran Lang, Abhishek Jindal, Sharad Vikram, Zhitao Gong, Sergi Caelles, Ross Hemsley, Gregory Thornton, Fangxiaoyu Feng, Wojciech Stokowiec, Ce Zheng, Phoebe Thacker, Çağlar Ünlü, Zhishuai Zhang, Mohammad Saleh, James Svensson, Max Bileschi, Piyush Patil, Ankesh Anand, Roman Ring, Katerina Tsihlas, Arpi Vezer, Marco Selvi, Toby Shevlane, Mikel Rodriguez, Tom Kwiatkowski, Samira Daruki, Keran Rong, Allan Dafoe, Nicholas FitzGerald, Keren Gu-Lemberg, Mina Khan, Lisa Anne Hendricks, Marie Pellat, Vladimir Feinberg, James Cobon-Kerr, Tara Sainath, Maribeth Rauh, Sayed Hadi Hashemi, Richard Ives, Yana Hasson, YaGuang Li, Eric Noland, Yuan Cao, Nathan Byrd, Le Hou, Qingze Wang, Thibault Sottiaux, Michela Paganini, Jean-Baptiste Lespiau, Alexandre Moufarek, Samer Hassan, Kaushik Shivakumar, Joost van Amersfoort, Amol Mandhane, Pratik Joshi, Anirudh Goyal, Matthew Tung, Andrew Brock, Hannah Sheahan, Vedant Misra, Cheng Li, Nemanja Rakićević, Mostafa Dehghani, Fangyu Liu, Sid Mittal, Junhyuk Oh, Seb Noury, Eren Sezener, Fantine Huot, Matthew Lamm, Nicola De Cao, Charlie Chen, Gamaleldin Elsayed, Ed Chi, Mahdis Mahdieh, Ian Tenney, Nan Hua, Ivan Petrychenko, Patrick Kane, Dylan Scandinaro, Rishub Jain, Jonathan Uesato, Romina Datta, Adam Sadovsky, Oskar Bunyan, Dominik Rabiej, Shimu Wu, John Zhang, Gautam Vasudevan, Edouard Leurent, Mahmoud Alnahlawi, Ionut Georgescu, Nan Wei, Ivy Zheng, Betty Chan, Pam G Rabinovitch, Piotr Stanczyk, Ye Zhang, David Steiner, Subhajit Naskar, Michael Azzam, Matthew Johnson, Adam Paszke, Chung-Cheng Chiu, Jaume Sanchez Elias, Afroz Mohiuddin, Faizan Muhammad, Jin Miao, Andrew Lee, Nino Vieillard, Sahitya Potluri, Jane Park, Elnaz Davoodi, Jiageng Zhang, Jeff Stanway, Drew Garmon, Abhijit Karmarkar, Zhe Dong, Jong Lee, Aviral Kumar, Luowei Zhou, Jonathan Evens, William Isaac, Zhe Chen, Johnson Jia, Anselm Levskaya, Zhenkai Zhu, Chris Gorgolewski, Peter Grabowski, Yu Mao, Alberto Magni, Kaisheng Yao, Javier Snaider, Norman Casagrande, Paul Suganthan, Evan Palmer, Geoffrey Irving, Edward Loper, Manaal Faruqui, Isha Arkatkar, Nanxin Chen, Izhak Shafran, Michael Fink, Alfonso Castaño, Irene Giannoumis, Wooyeol Kim, Mikołaj Rybiński, Ashwin Sreevatsa, Jennifer Prendki, David Soergel, Adrian Goedeckemeyer, Willi Gierke, Mohsen Jafari, Meenu Gaba, Jeremy Wiesner, Diana Gage Wright, Yawen Wei, Harsha Vashisht, Yana Kulizhskaya, Jay Hoover, Maigo Le, Lu Li, Chimezie Iwuanyanwu, Lu Liu, Kevin Ramirez, Andrey Khorlin, Albert Cui, Tian LIN, Marin Georgiev, Marcus Wu, Ricardo Aguilar, Keith Pallo, Abhishek Chakladar, Alena Repina, Xihui Wu, Tom van der Weide, Priya Ponnapalli, Caroline Kaplan, Jiri Simsa, Shuangfeng Li, Olivier Dousse, Fan Yang, Jeff Piper, Nathan Ie, Minnie Lui, Rama Pasumarthi, Nathan Lintz, Anitha Vijayakumar, Lam Nguyen Thiet, Daniel Andor, Pedro Valenzuela, Cosmin Paduraru, Daiyi Peng, Katherine Lee, Shuyuan Zhang, Somer Greene, Duc Dung Nguyen, Paula Kurylowicz, Sarmishta Velury, Sebastian Krause, Cassidy Hardin, Lucas Dixon, Lili Janzer, Kiam Choo, Ziqiang Feng, Biao Zhang, Achintya Singhal, Tejasi Latkar, Mingyang Zhang, Quoc Le, Elena Allica Abellan, Dayou Du, Dan McKinnon, Natasha Antropova, Tolga Bolukbasi, Orgad Keller, David Reid, Daniel Finchelstein, Maria Abi Raad, Remi Crocker, Peter Hawkins, Robert Dadashi, Colin Gaffney, Sid Lall, Ken Franko, Egor Filonov, Anna Bulanova, Rémi Leblond, Vikas Yadav, Shirley Chung, Harry Askham, Luis C. Cobo, Kelvin Xu, Felix Fischer, Jun Xu, Christina Sorokin, Chris Alberti, Chu-Cheng Lin, Colin Evans, Hao Zhou, Alek Dimitriev, Hannah Forbes, Dylan Banarse, Zora Tung, Jeremiah Liu, Mark Omernick, Colton Bishop, Chintu Kumar, Rachel Sterneck, Ryan Foley, Rohan Jain, Swaroop Mishra, Jiawei Xia, Taylor Bos, Geoffrey Cideron, Ehsan Amid, Francesco Piccinno, Xingyu Wang, Praseem Banzal, Petru Gurita, Hila Noga, Premal Shah, Daniel J. Mankowitz, Alex Polozov, Nate Kushman, Victoria Krakovna, Sasha Brown, MohammadHossein Bateni, Dennis Duan, Vlad Firoiu, Meghana Thotakuri, Tom Natan, Anhad Mohananey, Matthieu Geist, Sidharth Mudgal, Sertan Girgin, Hui Li, Jiayu Ye, Ofir Roval, Reiko Tojo, Michael Kwong, James Lee-Thorp, Christopher Yew, Quan Yuan, Sumit Bagri, Danila Sinopalnikov, Sabela Ramos, John Mellor, Abhishek Sharma, Aliaksei Severyn, Jonathan Lai, Kathy Wu, Heng-Tze Cheng, David Miller, Nicolas Sonnerat, Denis

Vnukov, Rory Greig, Jennifer Beattie, Emily Caveness, Libin Bai, Julian Eisenschlos, Alex Korchemniy, Tomy Tsai, Mimi Jasarevic, Weize Kong, Phuong Dao, Zeyu Zheng, Frederick Liu, Fan Yang, Rui Zhu, Mark Geller, Tian Huey Teh, Jason Sanmiya, Evgeny Gladchenko, Nejc Trdin, Andrei Sozanschi, Daniel Toyama, Evan Rosen, Sasan Tavakkol, Linting Xue, Chen Elkind, Oliver Woodman, John Carpenter, George Papamakarios, Rupert Kemp, Sushant Kafle, Tanya Grunina, Rishika Sinha, Alice Talbert, Abhimanyu Goyal, Diane Wu, Denese Owusu-Afriyie, Cosmo Du, Chloe Thornton, Jordi Pont-Tuset, Pradyumna Narayana, Jing Li, Sabaer Fatehi, John Wieting, Omar Ajmeri, Benigno Uria, Tao Zhu, Yeongil Ko, Laura Knight, Amélie Héliou, Ning Niu, Shane Gu, Chenxi Pang, Dustin Tran, Yeqing Li, Nir Levine, Ariel Stolovich, Norbert Kalb, Rebeca Santamaria-Fernandez, Sonam Goenka, Wenny Yustalim, Robin Strudel, Ali Elqursh, Balaji Lakshminarayanan, Charlie Deck, Shyam Upadhyay, Hyo Lee, Mike Dusenberry, Zonglin Li, Xuezhi Wang, Kyle Levin, Raphael Hoffmann, Dan Holtmann-Rice, Olivier Bachem, Summer Yue, Sho Arora, Eric Malmi, Daniil Mirylenka, Qijun Tan, Christy Koh, Soheil Hassas Yeganeh, Siim Põder, Steven Zheng, Francesco Pongetti, Mukarram Tariq, Yanhua Sun, Lucian Ionita, Mojtaba Seyedhosseini, Pouya Tafti, Ragha Kotikalapudi, Zhiyu Liu, Anmol Gulati, Jasmine Liu, Xinyu Ye, Bart Chrzaszcz, Lily Wang, Nikhil Sethi, Tianrun Li, Ben Brown, Shreya Singh, Wei Fan, Aaron Parisi, Joe Stanton, Chenkai Kuang, Vinod Koverkathu, Christopher A. Choquette-Choo, Yunjie Li, TJ Lu, Abe Ittycheriah, Prakash Shroff, Pei Sun, Mani Varadarajan, Sanaz Bahargam, Rob Willoughby, David Gaddy, Ishita Dasgupta, Guillaume Desjardins, Marco Cornero, Brona Robenek, Bhavishya Mittal, Ben Albrecht, Ashish Shenoy, Fedor Moiseev, Henrik Jacobsson, Alireza Ghaffarkhah, Morgane Rivière, Alanna Walton, Clément Crepy, Alicia Parrish, Yuan Liu, Zongwei Zhou, Clement Farabet, Carey Radebaugh, Praveen Srinivasan, Claudia van der Salm, Andreas Fidjeland, Salvatore Scellato, Eri Latorre-Chimoto, Hanna Klimczak-Plucińska, David Bridson, Dario de Cesare, Tom Hudson, Piermaria Mendolicchio, Lexi Walker, Alex Morris, Ivo Penchev, Matthew Mauger, Alexey Guseynov, Alison Reid, Seth Odoom, Lucia Loher, Victor Cotruta, Madhavi Yenugula, Dominik Grewe, Anastasia Petrushkina, Tom Duerig, Antonio Sanchez, Steve Yadlowsky, Amy Shen, Amir Globerson, Adam Kurzrok, Lynette Webb, Sahil Dua, Dong Li, Preethi Lahoti, Surya Bhupatiraju, Dan Hurt, Haroon Qureshi, Ananth Agarwal, Tomer Shani, Matan Eyal, Anuj Khare, Shreyas Rammohan Belle, Lei Wang, Chetan Tekur, Mihir Sanjay Kale, Jinliang Wei, Ruoxin Sang, Brennan Saeta, Tyler Liechty, Yi Sun, Yao Zhao, Stephan Lee, Pandu Nayak, Doug Fritz, Manish Reddy Vuyyuru, John Aslanides, Nidhi Vyas, Martin Wicke, Xiao Ma, Taylan Bilal, Evgenii Eltyshev, Daniel Balle, Nina Martin, Hardie Cate, James Manyika, Keyvan Amiri, Yelin Kim, Xi Xiong, Kai Kang, Florian Luisier, Nilesh Tripuraneni, David Madras, Mandy Guo, Austin Waters, Oliver Wang, Joshua Ainslie, Jason Baldridge, Han Zhang, Garima Pruthi, Jakob Bauer, Feng Yang, Riham Mansour, Jason Gelman, Yang Xu, George Polovets, Ji Liu, Honglong Cai, Warren Chen, XiangHai Sheng, Emily Xue, Sherjil Ozair, Adams Yu, Christof Angermueller, Xiaowei Li, Weiren Wang, Julia Wiesinger, Emmanouil Koukoumidis, Yuan Tian, Anand Iyer, Madhu Gurumurthy, Mark Goldenson, Parashar Shah, MK Blake, Hongkun Yu, Anthony Urbanowicz, Jennimaria Palomaki, Chrisantha Fernando, Kevin Brooks, Ken Durden, Harsh Mehta, Nikola Momchev, Elahe Rahimtoroghi, Maria Georgaki, Amit Raul, Sebastian Ruder, Morgan Redshaw, Jinhyuk Lee, Komal Jalan, Dinghua Li, Ginger Perng, Blake Hechtman, Parker Schuh, Milad Nasr, Mia Chen, Kieran Milan, Vladimir Mikulik, Trevor Strohman, Juliana Franco, Tim Green, Demis Hassabis, Koray Kavukcuoglu, Jeffrey Dean, and Oriol Vinyals. Gemini: A family of highly capable multimodal models, 2023.

[TSWW22] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training, 2022.

[VSP+23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[WLL+22] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, Sen Xing, Guo Chen, Junting Pan, Jiashuo Yu, Yali Wang, Limin Wang,

and Yu Qiao. Internvideo: General video foundation models via generative and discriminative learning, 2022.

[XMYR16] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[YKK18] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 471–487, 2018.

[YWV+22] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models, 2022.

[ZLB23] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding, 2023.