Neptune: The Long Orbit to Benchmarking Long Video Understanding

Arsha Nagrani,*[*] Mingda Zhang,*[*] Ramin Mehran, Rachel Hornung,
Nitesh Bharadwaj Gundavarapu, Nilpa Jha, Austin Myers, Xingyi Zhou,
Boqing Gong, Cordelia Schmid, Mikhail Sirotenko, Yukun Zhu, Tobias Weyand
Google Research [†]

Abstract

This paper describes a dataset for understanding long videos. Many existing video datasets are focused on short clips (10s-30s), often due to the high cost of annotating long videos. While some long video datasets do exist, they can often be solved by powerful image models applied per frame (and often to very few frames) in a video. In order to mitigate both these problems, we propose a scalable dataset creation pipeline which leverages large models (VLMs and LLMs), to automatically generate dense, time-aligned video captions, as well as tough question answer decoy sets for video segments (up to 15 minutes in length). Our dataset Neptune covers a broad range of long video reasoning abilities and we provide subsets that emphasize multimodal reasoning. Since existing metrics for open-ended question answering are either rule-based or may rely on proprietary models, we will release a new open source model-based metric (GEM) to score open-ended responses on Neptune.

1 Introduction

Videos are experiencing an *explosion* moment online, with new research constantly pushing the frontier for video and language tasks such as video question answering (VideoQA) [55, 68, 54, 57, 34]. Early video and language models, while adept at VideoQA, have largely focused on short, trimmed clips (less than 1 minute long [60, 54]). The recent release of powerful, longer context *multimodal* models (eg. Gemini 1.5 [41] and GPT4 [1]), however, has ushered in the promise of models being able to reason over millions of tokens, covering longer stretches of videos (many minutes long).

While promising, these claims, however, are often evidenced by qualitative examples, or results on small-size datasets – for example the 1H-VideoQA [41] benchmark, which while valuable, only consists of 125 questions. Most existing video benchmarks for question answering still tend to focus on short, trimmed clips (e.g., Next-QA [54]). Other datasets that *do* contain longer videos are often 'short-term' benchmarks disguised as long-term ones, evidenced by models that are able to solve them with a single (or a few) frames (eg. some tasks on the LVU dataset [52] such as scene prediction of movies), or contain strong linguistic biases, as shown by MoreVQA [35], which gets strong performance on the EgoSchema [34] dataset without access to the video at all.

There is clearly a distinct lack of objective long-video benchmarks in the field. A key challenge in creating a truly long form video understanding dataset is the significant manual cost required to select, watch, understand and annotate long videos with free-form natural language. Answering challenging questions about longer videos is often a *multimodal* (as it may involve listening to the audio track in addition to watching the video), and *non-linear* endeavour (as sometimes it is necessary to rewind and rewatch key parts to answer a question). Proposing suitable high-level questions that

[*]Equal Contribution
[†]Authors now at Google DeepMind

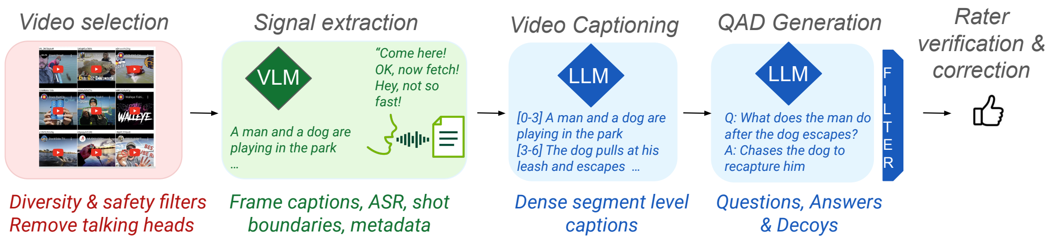


Figure 1: **Pipeline Overview:** Our pipeline consists of 5 key stages - (i) Video selection, where suitable videos are identified from YouTube, (ii) Signal extraction, (iii) Video level captioning, (iv) Question, answer and decoy (QAD) generation and (v) Manual rater verification. The first four stages are entirely automatic. Before rater verification, we automatically filter out QADs that can be solved by an LLM without access to the video content.

are not trivially solved by a few frames is also tricky for humans to do consistently and with adequate diversity. The key aim of this paper is to solve this challenge by collecting and annotating a long-form video dataset. Inspired by EgoSchema, we do this by proposing a scalable dataset creation pipeline (Fig. 1) that leverages strong foundational Video Language Models (VLMs) and Large Language Models (LLMs) with carefully designed prompts to generate tough question-answer-decoy (QAD) sets for videos of variable lengths up to 15 minutes long. Note that unlike EgoSchema, we carefully design stages in our pipeline to generate dense, time-aligned video captions automatically, from which the QADs can be automatically derived. This is done by extracting image captions, automatic speech recognition (ASR), shot boundaries and video metadata, and combining these signals with multi-stage, chain of thought prompting of an LLM. This is unlike the EgoSchema pipeline, which relies heavily on manually obtained dense captions for egocentric videos, and hence our method can be easily applied to any video online. Our dataset is called Neptune[3], and covers a diverse range of videos, is multimodal (requires audio and visual information), and poses challenging questions for videos that test a variety of reasoning abilities over long time horizons.

Neptune allows for two modes of evaluation: multiple-choice and open-ended question answering. Since existing metrics for open-ended question answering are either rule-based and derived from captioning (WUPS [53], CIDEr [47], etc) or are LLM-based evals that rely on proprietary APIs (such as ChatGPT[4]), we finetune an open source model on a generic answer equivalence dataset [6] to score question answering results and evaluate it as a metric on a manually annotated answer equivalence dev set. We also provide benchmarking with state-of-the-art video models.

To summarise, we make the following contributions: (i) We propose a new, scalable pipeline to generate complex QAD annotations for any video online. Our pipeline involves careful prompting of large VLMs. (ii) We use this pipeline to generate the Neptune evaluation-only dataset, which consists of 3,268 QAD annotations for 2,405 videos. We also release two *challenging* subsets, NEPTUNE-MMH and NEPTUNE-MMA, for which we attempt to remove linguistic bias and verify that *vision* plays an important role. (iii) We provide both multiple choice and open-ended evaluation metrics. For the latter, we propose a new open-ended metric called Gemma Equivalence Metric (GEM) which outperforms rule-based metrics on a manually annotated answer equivalence dataset; and finally (iv) We provide benchmarking and ablations of state-of-the-art VideoQA models on the Neptune sets. Benchmarking shows a significant gap between open-source video models and Gemini-1.5-pro, which sets the state-of-the-art on Neptune. All data will be released publicly to the research community.

2 Related Works

Video Question Answering: Video Question-Answering (VideoQA) is an important task for assessing multimodal video understanding systems' ability to reason about videos [55, 68, 54, 57, 34]. Vision and language models for this task can be broadly classified into three categories: (i) early end-to-end VLMs for this task which typically consists of strong vision and language encoders/decoders, such as Flamingo [2], BLIP2 [29], Video-Llama [66], GIT2 [48] and PALI [7–9]. These typically are

[3]Named after the planet with the longest orbit

[4]https://openai.com/index/chatgpt/

moderate sized models, and memory limits often lead to significant downsampling: *e.g.* temporally sampling a few frames with large strides [48, 8] or spatially subsampling each frame to a single token [58, 70, 49]; (ii) Socratic style models [65], which consists of combining various specialised *frozen* models with carefully prompted state-of-the-art VLMs and LLMs (eg. MoreVQA [35]) and (iii) end-to-end large multimodal models such as Gemini[18] and GPT-4 [1], which have long context lengths and can ingest multimodal data, including video, sound and text.

Video QA Benchmarks: Key datasets have pushed towards assessing reasoning for temporal questions [19, 54, 51], longer videos [60, 34], as well as focusing on diverse domains like instructional [57] and egocentric videos [16, 34]. We summarise existing VideoQA benchmarks in Table 1. Most datasets either focus on shorter videos (less than 100s), or are short video datasets 'in disguise', and can actually be solved with a few frames (*e.g.* ActivityNet-QA [61] or MovieQA [44]). 1H-VideoQA [41] consists of videos longer than 1 hour, but is limited to 125 questions and is closed-source. EgoSchema [34] is the closest work to ours in motivation (and indeed our pipeline is inspired by theirs), but there are some key differences: (i) it is limited to egocentric videos of exactly 3 minutes each, while Neptune covers many domains and follows a more natural length distribution for online videos (16s to 15min); (ii) and more importantly EgoSchema also has strong image and linguistic biases, while Neptune mitigates these (Sec. 5). Unlike other benchmarks which come with their own training sets (eg. MSR-VTT [56], ActivityNet [60]), we propose a generalisation-focused *zero-shot* evaluation regime. The goal for Neptune is to benchmark any model, pre-trained with any external dataset or task, in order to assess real-world domain transfer. Hence we release *test* sets only. More discussion on related datasets is provided in the appendix.

Name	Ann	Rater V	Avg. len (s)	# Vids (total/test)	# Samples (total/test)	Available
MovieQA [44]	QAD	✓	200	6,771/1,288	6,462/1,258	✗†
MSRVTT-QA [55]	QA	✗	15	10,000/2,990	243,680/72,821	✓
ActivityNet-QA [60]	QA	✓	180	5,800/1,800	58,000/18,000	✓
NExTQA [54]	QAD	✓	44	5,440/1,000	52,044/8,564	✓
IntentQA [28]	QAD	✓	44	4,303/430	16,297/2,134	✓
EgoSchema [34]	QAD	✓	180	5,063/5,063	5,063/5,063	✓‡
Perception Test [38]	QAD	✓	23	11,600	38,000	✓
MVBench [30]	QAD	✗	16	3,641	4,000	✓
Video-Bench [36]	QAD	✓	56	5,917	17,036	✓
AutoEval-Video [10]	QA	✓	14.6	327	327	✓
1H-VideoQA [41]	QAD	✓	6,300 (max)	125	125	✗
Neptune	QAD	✓	150/901*	2,405	3,268	✓
Neptune-MMH	QAD	✓	159/901*	1,000	1,171	✓
Neptune-MMA	QAD	✓	154/901*	1,043	1,157	✓

Table 1: **Comparison to Existing VideoQA datasets: Ann. Type:** Annotation Type, **QAD:** Question, Answer and Decoys, **Rater V:** Rater verified manually. † Movies are no longer available. ‡ Annotations are hidden behind a test server, 500 are public. *average/max length.

Metrics for open-ended VideoQA: Earlier QA datasets consisted of short answers [54] (sometimes a single word), typically from a closed set, and therefore metrics such as accuracy or accuracy with exact match (EM) can be applied. As datasets have evolved with more real-world annotation (longer, open-set answers), designing a metric becomes challenging. Existing rule-based metrics for captioning, such as BLEU [37], ROUGE [32] and CIDEr [47] can be applied, however they all primarily measure n-gram overlap, and do not capture the inherent subjectivity of the task, where different phrasing is often equally valid. Other metrics for captioning include SPICE [3] (adds action and object relationships), while model-based metrics using earlier language models or image-language models include BERT-Score [67], BERT-Score++ [59] (fine-tunes BERT for image captioning), LEIC [13], NUBIA [25], TIGEr [24], CLIPScore [22], and EMScore [42]. For answer equivalence specifically, token F1 and exact match (EM) have been used, but suffer many of the same shortcomings that rule-based metrics do, and EM is often too strict for open-ended eval. BEM [6] finetunes BERT on an answer-equivalence dataset, and shows that this provides a better score for QA. Recently, strong LLMs trained with reinforcement learning from human feedback (RLHF) that already exhibit strong human alignment [5] are used in works such as VideoChatGPT [33] and MovieChat [43] (LLM-as-a-judge). A challenge here, however, is that the models accessed (ChatGPT) are called via proprietary APIs, where the underlying model may be non-static, thereby leading to non-reproducability in the metric. Instead, we take a state-of-the-art open-sourced lightweight language

Figure 2: **Examples from Neptune:** We show examples from the dataset that highlight key question types from our dataset. We show 2 frames from each video. Correct answer is provided in green and decoys are shown in red. Best viewed zoomed in and in colour. Some decoys are summarised for brevity.

model [45] and finetune it on a public answer equivalence dataset [6], to create an open-source, static, model-based evaluation metric.

3 Neptune

In this section we describe our dataset generated by the pipeline described in Sec. 4. We first discuss motivating principles, which affect much of the prompt design in the pipeline stage (Sec. 4). Each video contains one or more annotation sets, which consists of a question, an answer to the question and four decoys (which are used for multiple choice evaluation). Our key motivation is that questions should not be answerable by: (i) looking at a single (or few) frames; (ii) using text-only LLMs alone (language, common sense) that have no access to the video; (iii) with only the video's speech transcript, and (iv) questions should cover a number of high-level "question types", which are discussed next and described in more detail in the appendix.

Question Types. Neptune covers a broad range of long video reasoning abilities, which are provided as 'question type' labels for each question. Examples are provided in Fig. 2, and the distribution of questions per question type is depicted in Fig. 3 (right). More information about the distribution of question types is provided in the appendix. Question types are obtained by carefully prompting an LLM (described in Sec. 4.3) and include *Video Summarisation*, which involves summarising and comparing long parts of the video, as well as identifying the most important segments of the video; *Visual Reasoning*, which involves understanding visual elements, as well as reasoning about why visual content is used (*e.g.* to convey a certain mood); *Temporal Ordering*, including the timeline of events; *State Changes*; *Counting* of higher level instances; *Cause and Effect*, and understanding the *Unspoken Message* or *Creator Intent* in a video.

Dataset Statistics. Our dataset consists of **3,268** questions from **2,405** videos, covering **100** hours of video. We truncate videos longer than 15 minutes, with the smallest video being 16 seconds and the average length of videos being 2.5 minutes. We show the distribution of video lengths in Fig. 3 (top, left). Note that greater than 12% of the videos are longer than 5 minutes (305 videos) and over 25% are longer than 3 minutes, which is the maximum length of videos in the EgoSchema dataset. The distribution of questions per question type is depicted in Fig. 3 (top, right). The most frequent question type is Temporal Ordering, followed by Summarization. Questions are on average 16.3 words long, while answers and decoys are 29.5 and 29.0 words long respectively. A full distribution

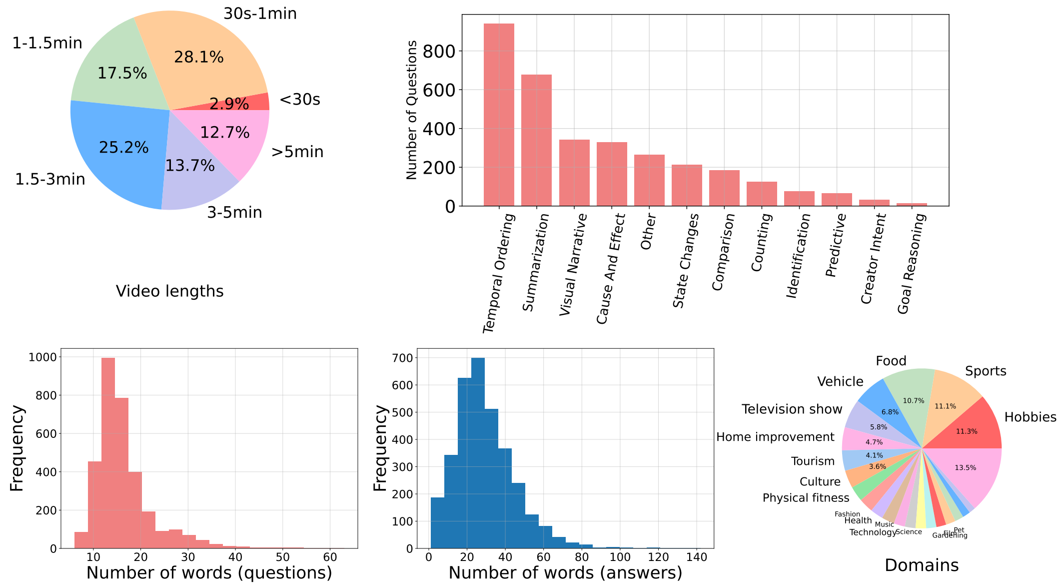


Figure 3: **Neptune Statistics:** We show, the distribution of video lengths (top, left), the number of questions per question type (top, right), the distribution question and answer lengths (bottom, left and middle) and the domains in Neptune (bottom, right). Note that greater than 12% of the videos are longer than 5 minutes (305) and over 25% are longer than 3 minutes. An expanded plot of the video domains is provided in the appendix.

of lengths can be seen in Fig. 3 (bottom, left). We also note that the videos in Neptune cover a diverse range of topics (Fig. 3 – bottom, right), an expanded version of this plot is provided in the appendix.

4 Dataset Creation Pipeline

An overview of our pipeline can be found in Fig. 1. In order to reduce human effort, we leverage automatic tools to (i) find suitable videos (ii) extract useful signals and then (iii) automatically generate video level captions and QADs. We then send the data to human raters for the final manual verification stages. Each stage is described in detail below.

4.1 Video Selection and Signal Extraction

Video Selection: We begin with the YT-Temporal-1Bn [63] dataset. Because this dataset was constructed to include videos with strong speech and visual alignment, it consists of a lot of videos where *'talking heads'* dominate the screen (eg. VLOGs, product placements, etc). We attempt to reduce the number of such videos in our dataset in order to capture more interesting scenes, objects and actions. This is done by extracting face detections with frontal gaze where face bounding-box height is greater than 20%, and removing videos where more than 30% of frames have such frontal gaze. This allows us to get a better distribution of videos with more complex actions/activities. We then apply safety filters to remove racy, local controversy content etc, as well as applying filters to maximise semantic and person diversity. Details about these processes are provided in the appendix.
Signal Extraction: For each video we extract the following signals: (i) *Frame captions:* A visual description of each frame (extracted at 1fps) is obtained from PaLI-3 [9]. (ii) *ASR:* the speech is transcribed using the YouTube API; (iii) *Metadata:* We obtain the YouTube title and the description for each video; and (iv) *Shot boundaries* for each video.

4.2 Automatic Video Captioning

The signals described above (frame captions, ASR, title and description, shot boundaries) are automatically combined to create video-level captions using a custom prompted LLM (Gemini-1.0-Pro [18]). Note that this is a key difference between our pipeline and the EgoSchema [34] pipeline, as EgoSchema relies on manually annotated captions for egocentric videos that were obtained by raters narrating their actions. By automating this stage, our pipeline can be applied to any YouTube video. Video captions are obtained using the following steps (summarised in a figure provided in the appendix):

Shot Visual Captions: Using the shot boundaries, the *frame captions* are summarized into shot-level descriptions (*shot captions*) by prompting the same LLM. We then create a script for each video containing the shot timestamps, the shot visual captions and the ASR transcript.

Topic and Description Pairs: If ASR exists, an initial list of structured topics for the video (as well as a short description for each topic) is formed by prompting an LLM with the ASR and a custom prompt (provided in the appendix). Note that this yields decent topics as the initial list of videos have already been selected (by the YT-Temporal-1Bn authors) to have a strong correlation between ASR and visual content.

Shot Clustering: Shots are then clustered using an LLM prompted with the semantic topics obtained above. In each cluster, there may be one or many shots that correspond to that topic. The exact prompt used is provided in the appendix.

Segment Captions: Consecutive shots of the same topic are then merged as one segment. Shots of the same topic that are not contiguous are treated as separate segments (see appendix for an example). We then generate dense captions for each segment using a custom prompt (see appendix).

Adding Visual Support: To extract a better visual description of the segment that will be used for QA generation in the next phase, an extra step is performed to get visual support for each segment. That visual support is stored separately in conjunction with the dense caption for the segment. For this purpose, the dense caption from the previous step is used alongside the shot level visual captions. The LLM prompt used is provided in the appendix, and the the LLM used for all the above steps is Gemini-1.0-Pro [18].

4.3 QAD (Question-Answer-Decoy) Generation

We automatically generate questions, answers and decoys (QADs) using the video captions from above and custom prompted LLMs. Our prompts are inspired by the EgoSchema dataset pipeline [34], with key modifications to generate more visually focused questions, as well as to generate questions belonging to a set of different question types. The exact prompts used are provided in the appendix. We generate QADs in two stages: (i) Given the video captions from the previous step, we first generate questions and answers; (ii) in the second stage we generate six decoys given the questions and answers from the previous stage. We found this 2-stage method to work better empirically than generating the QADs all in one go.

4.4 LLM-based Blind Filters

QAD filter: LLM-based generation can sometimes yield question-answer-decoy triplets that can be answered from common sense or external world knowledge without the video as context. In particular, we observed that LLMs are often capable of inferring the correct answer from subtle cues in the answer candidates, for example if the correct answer is a positive sentiment while the decoys are negative sentiments. To remove such questions, we employ an LLM-based blind filter similar to the "blind filtering baseline" in [34]. We prompt an LLM (Gemini-1.0-pro) to rank the answer candidates to a question. The exact prompt can be found in the supplementary material. To avoid false rejections due to random correct guesses, we repeat this process three times and only filter out questions where the model predicted the correct answer at least two times out of three (this number was selected to maximise number of videos left given the accuracy trade-off and is discussed in the appendix). We find that chain-of-thought reasoning improves accuracy so we ask the model to provide a rationale alongside its ranking.

ASR filter: After applying the above filter, many of the remaining questions can still be answered solely based on spoken content. To ensure that questions require understanding of the visual content, we apply the above filter with the ASR transcript of the video as additional context. This is used to create a hard subset of the dataset, described in Sec. 5.1.

4.5 Manual Rater Verification

The final stage involves manual human verification. Raters are first asked to rate the quality of the question based on 4 criteria (details in the appendix). If the question is not suitable, the entire QAD set is discarded. If the question is accepted, the raters are then asked what modalities are required to answer the question. Choices are: "audio+video", "video-only", or "audio-only". Next, raters are asked to either accept the answer as-is or modify it. Decoys are annotated in a final stage. Given the six LLM-generated decoy candidates, raters are asked to verify that they are actually incorrect answers to the question and select the four most challenging ones. If less than four decoys are suitable, we provide a text field for raters to write their own decoys. Screenshots of the rater UI are provided in the appendix. We noticed that rater corrections reintroduce a small amount of questions that can be answered without context, so as a final step we repeat the QAD filter described above to create harder subsets. This is described in Sec. 5.1. More details about rater training, replication (multiple raters per question) and pipelining are provided in the appendix. We applied two rounds of manual rater verification to improve dataset quality.

5 Experiments

We first introduce three Neptune sets and our evaluation metrics and then present evaluations on both baseline and state-of-the-art models.

5.1 Neptune Sets

Because we seeded our dataset from the YT-Temporal-1Bn [64] videos, we note that it contains some videos where ASR can play a big role in contributing to the video content. In order to create a more challenging *visual* benchmark, we create two Neptune-MM (multimodal sets), where we identify videos where vision should play an important role. The first subset Neptune Multimodal Human (NEPTUNE-MMH) is created by using the rater annotations for what modalities are required to answer the question (described in Sec. 4.5), and discarding questions which the raters marked can be solved by audio-only. The second subset Neptune Multimodal Automatic (NEPTUNE-MMA) is obtained by applying the ASR filter described in Sec. 4.4. Note the QAD filter is applied to both sets (as we definitely want to filter out questions that can be solved without access to the video at all). The statistics for both sets are provided in Table 1. We encourage the community to evaluate on these *harder* subsets.

5.2 Evaluation and Metrics

We explore two different protocols for evaluation of question answering - multiple choice evaluation (which involves selecting the correct answer amidst 4 decoys), and open-ended evaluation, which involves producing an answer directly without any decoys and assessing answer quality directly. While the former has the advantage of easier metrics (simple accuracy), the latter removes any potential confounding biases in the decoys. In the next section, we outline our process for creating a new open-ended metric called GEM.

Gemma Equivalence Metric (GEM): As discussed in Sec. 2, existing metrics for open-ended QA either lack robustness or rely on proprietary LLM APIs that can change over time. We therefore aim to produce a static open-ended metric. Towards this, we first manually construct a labelled dev-set with 292 (question, reference answer, candidate answer) triplets, with equivalence scores between 0 and 1. See appendix for details on the construction of the dev set. We then benchmark a number of rule-based and model-based metrics on this set in Table 2. To demonstrate the two ends of the scale, we first note that rule-based metrics such as CIDEr [47] and ROUGE-L [32] obtain F1-Scores of 56.4 and 62.2, while an LLM-based metric using Gemini-1.5-pro [41] gets an F1-Score of 72.8 (but is closed-source). Next, we apply static open-source lightweight language models, namely the Gemma family of models i.e. Gemma-2B [45], Gemma-7B [45] and Gemma-9B [46] to judge the answers in a zero-shot setting and find that performance improves with model size, with Gemma-9B bridging the gap well between traditional metrics and the Gemini-1.5-pro based metric. Finally, we fine-tune Gemma-9B on the open-source BEM answer equivalence dataset [6], and find that Gemma-9B finetuned on the BEM dataset performs the best on our dev-set among the Gemma models. We call the metric obtained with this model Gemma Equivalence Metric (GEM). Note that this metric takes into account the question when comparing whether two answers are equivalent, which is unlike captioning metrics such as CIDEr which omit the question entirely. In Table 3, we report open-ended evaluations using our proposed GEM metric in addition to closed-ended MCQ accuracy. We will release GEM publicly to enable reproducible open-ended evaluations.

Metric	Fine-tuning data	F1-Score
CIDEr [47]	None	56.4
ROUGE-L [32]	None	62.2
BEM [6]	BEM [6]	61.5
Gemma-2B-IT [45]	None	56.3
Gemma-7B-IT [45]	None	65.2
Gemma-9B-IT [46]	None	70.3
Gemma-9B-IT [46] (GEM)	BEM [6]	71.2
Gemini-1.5-pro [41]	None	**72.8**

Table 2: Evaluation of open-ended metrics on GEM answer equivalence dev set.

5.3 Benchmarks

We describe all benchmarks used below. Implementation details are provided in the appendix.
Blind Baselines: We evaluate Gemini-1.5-pro [41] using a text-only prompt in two settings: (i) we feed only the question, answer and decoys to the model (QAD baseline). (ii) we also feed ASR as an input for a QAD+ASR baseline. This helps identify questions that can be answered by prior or commonsense knowledge, or ASR only without obtaining information from video.
Image Models: We use the BLIP2-T5-XL [29] model, which contains a 1B vision encoder [14] and a 3B text-decoder [39]. We feed the center frame of the video as the visual input, with prompt "Answer in one letter" followed by the question and shuffled answer and decoys. We also evaluate Gemini-1.5-pro [41], feeding only the center frame.
Video Models:: We experiment with 3 different categories of VideoQA models:
(i) Short Context MLLMs - Video-LLaVA [31], and VideoLLaMA2 [12]. We also experiment with a simple socratic JCEF (Just Caption Every Frame) [35], which consists of a VLM to extract per-frame captions and an LLM to perform reasoning on top of these captions to answer the question.
(ii) Long Context MLLMs which are open-source, including MA-LMM [20], MiniGPT4-Video [4], and MovieChat [43].
(iii) Long Context MLLMs which are closed-source, namely the Gemini 1.5 model family [41].
Implementation Details: For Video-LLaVA [31] we feed 8 uniformly sampled frames (resized to a minimum side length of 320 pixels) along with the question. We reimplement JCEF from the original paper [35] with updated components - i.e. 16 uniformly sampled frame captions obtained using PaLI-3 [8], and feed them as a text prompt to Gemini-1.0-pro along with the question and decoys. For MiniGPT4-Video, we use the public codebase[5] which routes videos longer than 3 minutes to a Goldfish model and those shorter to their older MiniGPT-video model. We evaluate both the Gemini-1.5-pro and Gemini-1.5-flash models described in [41]. We also experiment with feeding in ASR to the Gemini-1.5-pro model as well. Frame selection is as other model except that MA-LMM has 20 and 120 and MiniGPT4-Video has default 45 with LLaMA-Video checkpoint. For MA-LMM we feed in 120 uniformly sampled frames. Further details on the models used are provided in the appendix.

5.4 Results

Results for all the baselines applied to all the 3 Neptune sets (Sec. 5.1) are provided in Table 3.
Single frame baselines: We examine model performance using both the BLIP2 image-only model and Gemini-1.5-pro with only a single frame (the middle frame from the video). The larger Gemini model outperforms BLIP-2, however performance with only a single frame is much lower than with multiple frames, as expected. We also show results using Gemini-1.5-pro on the first frame of the video in Fig. 4 (right), and find that using the middle frame performs better.
Video Models: We see a significant gap between open-source models and Gemini-1.5-pro, which sets the state-of-the-art on Neptune. Interestingly, we find that open-source models that are designed specially for longer context video understanding (MA-LMM [20], MiniGPT4-Video [4] and MovieChat [43]) perform worse than VideoLLaMA2, which only take 8 frames. This observation was also found by concurrent datasets such as MLVU [69] and LVBench [50]. The gap between many open-source and proprietary large MLLMs is also shown on concurrent datasets, *e.g.* LVBench [50], where MovieChat gets near-random results and Gemini-1.5-pro is the state-of-the-art.

[5] https://github.com/Vision-CAIR/MiniGPT4-video

Method	Modalities	Neptune-Full Acc. %	Neptune-Full GEM	Neptune-MMA Acc. %	Neptune-MMA GEM	Neptune-MMH Acc. %	Neptune-MMH GEM
Random	-	20.00		20.00		20.00	
Single Frame							
BLIP2 [29]	RGB (center frame)	34.80	9.20	23.85	5.40	28.10	8.50
Gemini-1.5-pro [41]	RGB (center frame)	55.57	14.11	45.64	14.91	51.75	13.27
Short Context MLLMs							
Video-LLaVA [31]	RGB (8 frames)	25.79	10.66	20.66	9.47	24.00	5.48
VideoLLaMA2 [11]	RGB (8 frames)	45.41	13.43	32.89	10.38	39.89	11.11
VLM captions + LLM (JCEF) [35]	VLM captions (16 frames)	58.51	12.27	48.31	14.04	56.45	11.50
Long Context MLLMs - open-source							
MA-LMM [20] (ActivityNet-QA fine-tuned)	RGB (120 frames)‡	20.22	10.67	17.63	5.93	19.51	5.04
MiniGPT4-Video [4]	RGB (45 frames)‡	24.63	5.26	21.43	6.03	22.89	6.19
MovieChat [43]	RGB (150 frames)	28.96	3.79	24.27	1.94	30.30	1.01
Long Context MLLMs - closed-source							
Gemini-1.5-pro [41]†	QAD only	51.53	15.64	39.07	14.04	41.84	11.50
Gemini-1.5-pro [41]†	QAD+ASR only	76.68	48.77	59.38	42.11	65.76	41.59
Gemini-1.5-pro [41]	RGB (150 frames)	69.31	29.45	59.46	27.19	66.70	23.89
Gemini-1.5-pro [41]	RGB (all frames)	68.94	28.83	57.48	24.56	65.58	26.55
Gemini-1.5-pro [41]	RGB (all frames + ASR)	**80.66**	48.77	**66.90**	**39.47**	**75.32**	**43.36**
Gemini-1.5-flash [41]	RGB (all frames + ASR)	76.90	**50.61**	59.72	37.72	71.05	39.82

Table 3: **Benchmarking performance on Neptune. All frames:** Visual frames extracted at 1fps. † Blind baselines with no access to the video. ‡ MCQ performance is close to random. This is discussed in the text. GEM results were computed on 10% of the dataset. We will update the paper with GEM results on the full set.

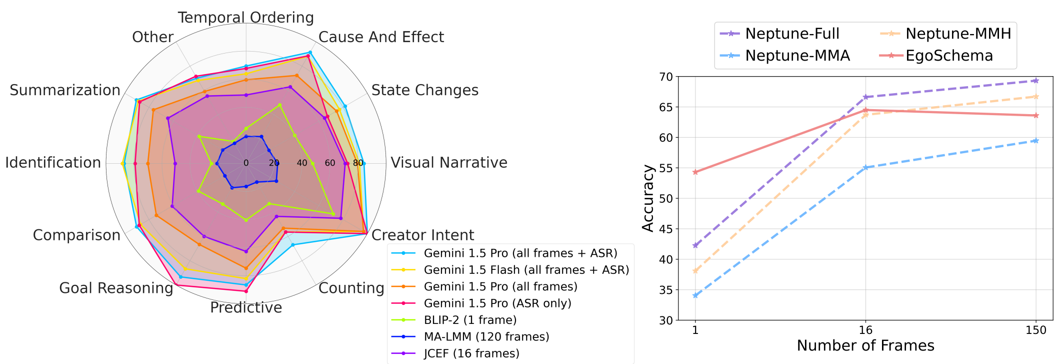


Figure 4: **Performance of different models across question types on Neptune-Full (left) and Neptune Vs Egoschema with different frame rates (right).** On the right we show Gemini 1.5 Pro's accuracy when linearly subsampling to 1, 16 or 150 frames. We note that (i) performance on the Neptune sets increases as more frames are provided while on EgoSchema it saturates after 16 frames and (ii) Neptune-MMA is more challenging than EgoSchema.

One reason for this near random performance may be the domain gap between the training sets of these models [20, 43] and Neptune – MovieChat is trained on movies and MA-LMM is designed to be fine-tuned on downstream QA datasets. By not providing a training set, we intentionally aim to assess generalization via zero-shot performance. We also note that the simple JCEF baseline, which consists of frame captions fed to an LLM for reasoning, outperforms all open-source models. The low performance of existing open-source models suggests Neptune may be a challenging benchmark for the future development of open-source models for long videos.

Gemini-1.5-pro Modality Ablations: Performance of Gemini-1.5-pro with QAD+ASR only as inputs is higher than performance with multiple video frames. However, the best performance is obtained with both frames and ASR, showcasing the complementary nature of the modalities. We also note that Gemini-1.5-pro with multiple frames outperforms the JCEF baseline, even though the JCEF baseline uses the same Gemini model (albeit applied to only a single frame at a time). This shows that frames must be processed together (by a video model), and a socratic baseline that simply looks at each frame individually performs much worse on this benchmark. This is unlike other datasets such as Next-QA where JCEF style baselines are almost state-of-the-art [35].

Challenging splits: We note that performance falls for all models on the Neptune-MMH and Neptune-MMA sets, demonstrating the challenging nature of these sets and the promising nature of our filters. The Neptune-MMA set (filtered automatically) is more difficult than the Neptune-

MMH set (filtered by human judgements of modalities needed). We hypothesize that this is because (i) it is easy for raters to underestimate how much information is in the ASR, e.g. if the video uses visual aides, people may falsely assume that the video was necessary to extract certain information; (ii) human and model assessments of difficulty vary.

Video Coverage: In this section we investigate Gemini 1.5 Pro's accuracy when linearly subsampling the video to 1, 16, or 150 frames. For 1 frame, we take the first frame of the video. We show results for all Neptune splits and compare them to results on EgoSchema in Fig. 4. Gemini 1.5 Pro's performance on Neptune increases as more frames are provided, while on EgoSchema it saturates after 16 frames, suggesting Neptune is better at requiring *long* video reasoning. Note that every video in EgoSchema has 180 frames (3 mins), whereas Neptune has variable lengths, with videos up to 15 minutes long. Results with the first frame on all Neptune splits (e.g. 34.05 on NEPTUNE-MMA) are also much lower than those on EgoSchema (54.3), pointing to higher image bias in the latter.

Open-ended results: We find that in general, results with GEM mirror the trends demonstrated by the multiple choice eval, with the exception of the Gemini-1.5-flash and Gemini-1.5-pro results, as well as the performance of the long context open-source models. Here we find that the FLASH model actually slightly exceeds the performance of the PRO model on the FULL set, and MovieChat performs worse on the open-ended task than other baselines, while better on the MCQ evaluation. A qualitative examination of the scores with the highest disparity shows that the FLASH model seems to indeed provide better open-ended answers. Examples of this are provided in the appendix.

Results per question type: Performance of different models across the different question types are shown in Fig. 4. We find that "Counting", "Temporal Ordering" and "State Change" questions are challenging for all models, pointing to areas for future work for video-language models, while "Cause and Effect" is easier. Interestingly, the Gemini-1.5-Pro model applied only to ASR without access to video frames is the best at "Goal Reasoning", which may be because human goals in videos are often mentioned in speech. Yet as expected, it is worse at the "Visual Narrative" questions, where Gemini-1.5-Pro models with access to RGB frames do much better.

6 Conclusion

We present Neptune, a new benchmark for VideoQA with a focus on *multimodal*, *high-level* understanding of *long videos*. Neptune is created using a scalable pipeline for arbitrary videos that minimizes (though not omits) human verification. Benchmarks are evaluated using MCQ and open-ended evals – for which we provide a new, open-source metric.

Limitations: The dataset may inherit biases of the Gemini model used to generate QADs. While VideoQA is a good proxy for video understanding, our dataset could be further improved by additional annotations – such as manually annotated temporal grounding, dense captions or entity labels.

Societal Impact is discussed in the appendix.

Acknowledgements: We are grateful to Antoine Yang, Anelia Angelova, Mario Lucic and Karel Lenc for their support with Gemini, as well as Anja Hauth for help with hosting the data. We thank Jannis Bulian and Christian Buck for their help with the BEM dataset. We'd also like to thank Brandon Black, Amanda Sadler, Michael Smouha, Rachel Stigler, Huisheng Wang, Hartwig Adam and Rahul Sukthankar for their support.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1, 3

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35: 23716–23736, 2022. 2

[3] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, pages 382–398. Springer, 2016. 3

[4] Kirolos Ataallah, Xiaoqian Shen, Eslam Abdelrahman, Essam Sleiman, Mingchen Zhuge, Jian Ding, Deyao Zhu, Jürgen Schmidhuber, and Mohamed Elhoseiny. Goldfish: Vision-language understanding of arbitrarily long videos. *arXiv preprint arXiv:2407.12679*, 2024. 8, 9

[5] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023. 3

[6] Jannis Bulian, Christian Buck, Wojciech Gajewski, Benjamin Boerschinger, and Tal Schuster. Tomayto, tomahto. beyond token-level answer equivalence for question answering evaluation. *arXiv preprint arXiv:2202.07654*, 2022. 2, 3, 4, 7, 8, 26

[7] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022. 2

[8] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. Pali-x: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*, 2023. 3, 8

[9] Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, et al. Pali-3 vision language models: Smaller, faster, stronger. *arXiv preprint arXiv:2310.09199*, 2023. 2, 5

[10] Xiuyuan Chen, Yuan Lin, Yuchen Zhang, and Weiran Huang. Autoeval-video: An automatic benchmark for assessing large vision language models in open-ended video question answering. *arXiv preprint arXiv:2311.14906*, 2023. 3

[11] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arxiv:2406.07476*, 2024. 9, 27

[12] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024. 8

[13] Yin Cui, Guandao Yang, Andreas Veit, Xun Huang, and Serge Belongie. Learning to evaluate image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5804–5812, 2018. 3

[14] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *CVPR*, 2023. 8

[15] Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024. 16

[16] Difei Gao, Ruiping Wang, Ziyi Bai, and Xilin Chen. Env-qa: A video question answering benchmark for comprehensive understanding of dynamic environments. pages 1675–1685, October 2021. 3

[17] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 2021. 32

[18] Gemini Team Google. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 3, 5, 6

[19] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. Agqa: A benchmark for compositional spatio-temporal reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11287–11297, 2021. 3

[20] Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13504–13514, 2024. 8, 9

[21] Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. *arXiv:2404.05726*, 2024. 26

[22] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *EMNLP*, 2021. 3

[23] De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin, Pavlo Molchanov, Zhiding Yu, and Jan Kautz. LITA: Language Instructed Temporal-Localization Assistant. *arXiv preprint arXiv:2403.19046*, 2024. 16

[24] Ming Jiang, Qiuyuan Huang, Lei Zhang, Xin Wang, Pengchuan Zhang, Zhe Gan, Jana Diesner, and Jianfeng Gao. Tiger: Text-to-image grounding for image caption evaluation. *arXiv preprint arXiv:1909.02050*, 2019. 3

[25] Hassan Kane, Muhammed Yusuf Kocyigit, Ali Abdalla, Pelkins Ajanoh, and Mohamed Coulibali. Nubia: Neural based interchangeability assessor for text generation. *arXiv preprint arXiv:2004.14667*, 2020. 3

[26] Zu Kim, André Araujo, Bingyi Cao, Cam Askew, Jack Sim, Mike Green, N Yilla, and Tobias Weyand. Improving fairness in large-scale object recognition by crowdsourced demographic information. *arXiv preprint arXiv:2206.01326*, 2022. 17

[27] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017. 16

[28] Jiapeng Li, Ping Wei, Wenjuan Han, and Lifeng Fan. Intentqa: Context-aware video intent reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11963–11974, 2023. 3

[29] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 2, 8, 9

[30] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. *arXiv preprint arXiv:2311.17005*, 2023. 3

[31] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 8, 9, 26

[32] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. 3, 7, 8

[33] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023. 3

[34] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. 1, 2, 3, 5, 6, 30

[35] Juhong Min, Shyamal Buch, Arsha Nagrani, Minsu Cho, and Cordelia Schmid. Morevqa: Exploring modular reasoning models for video question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 3, 8, 9

[36] Munan Ning, Bin Zhu, Yujia Xie, Bin Lin, Jiaxi Cui, Lu Yuan, Dongdong Chen, and Li Yuan. Video-bench: A comprehensive benchmark and toolkit for evaluating video-based large language models. *arXiv preprint arXiv:2311.16103*, 2023. 3

[37] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002. 3

[38] Viorica Patraucean, Lucas Smaira, Ankush Gupta, Adria Recasens, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Mateusz Malinowski, Yi Yang, Carl Doersch, et al. Perception test: A diagnostic benchmark for multimodal video models. *Advances in Neural Information Processing Systems*, 36, 2024. 3, 16

[39] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020. 8

[40] Ruchit Rawal, Khalid Saifullah, Ronen Basri, David Jacobs, Gowthami Somepalli, and Tom Goldstein. CinePile: A Long Video Question Answering Dataset and Benchmark. *arXiv preprint arXiv:2405.08813*, 2024. 16

[41] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. 1, 3, 7, 8, 9, 26, 27

[42] Yaya Shi, Xu Yang, Haiyang Xu, Chunfeng Yuan, Bing Li, Weiming Hu, and Zheng-Jun Zha. Emscore: Evaluating video captioning via coarse-grained and fine-grained embedding matching. In *CVPR*, pages 17929–17938, 2022. 3

[43] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Xun Guo, Tian Ye, Yan Lu, Jenq-Neng Hwang, and Gaoang Wang. MovieChat: From dense token to sparse memory for long video understanding. *arXiv:2307.16449*, 2023. 3, 8, 9

[44] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4631–4640, 2016. 3

[45] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024. 4, 7, 8, 26

[46] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. 7, 8, 26

[47] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, pages 4566–4575, 2015. 2, 3, 7, 8

[48] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022. 2, 3

[49] Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. End-to-end dense video captioning with parallel decoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6847–6857, 2021. 3

[50] Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Shiyu Huang, Bin Xu, Yuxiao Dong, Ming Ding, et al. Lvbench: An extreme long video understanding benchmark. *arXiv preprint arXiv:2406.08035*, 2024. 8

[51] Bo Wu, Shoubin Yu, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan. Star: A benchmark for situated reasoning in real-world videos. In *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 3

[52] Chao-Yuan Wu and Philipp Krahenbuhl. Towards long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1884–1894, 2021. 1

[53] Z Wu and M Palmer. Verbs semantics and lexical selection. inproceedings of the 32nd annual meeting on association for computational linguistics (pp. 133-138). In *Association for Computational Linguistics*, 1994. 2

[54] 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. 1, 2, 3

[55] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1645–1653, 2017. 1, 2, 3

[56] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*, pages 5288–5296, 2016. 3

[57] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1686–1697, 2021. 1, 2, 3

[58] Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *CVPR*, 2023. 3

[59] Yanzhi Yi, Hangyu Deng, and Jinglu Hu. Improving image captioning evaluation by considering inter references variance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 985–994, 2020. 3

[60] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9127–9134, 2019. 1, 3

[61] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *AAAI*, pages 9127–9134, 2019. 3

[62] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models. *NeurIPS*, 2021. 33, 35

[63] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. Merlot reserve: Neural script knowledge through vision and language and sound. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16375–16387, 2022. 5

[64] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. MERLOT reserve: Multimodal neural script knowledge through vision and language and sound. In *CVPR*, 2022. 7, 17

[65] Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, et al. Socratic models: Composing zero-shot multimodal reasoning with language. *ICLR*, 2023. 3

[66] Hang Zhang, Xin Li, and Lidong Bing. Video-LLaMA: An instruction-tuned audio-visual language model for video understanding. In *EMNLP 2023 Demo*, 2023. 2

[67] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with bert. In *ICLR*, 2020. 3

[68] Yaoyao Zhong, Junbin Xiao, Wei Ji, Yicong Li, Weihong Deng, and Tat-Seng Chua. Video question answering: Datasets, algorithms and challenges. *arXiv preprint arXiv:2203.01225*, 2022. 1, 2

[69] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*, 2024. 8

[70] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *CVPR*, 2018. 3

A Related Works

Here we provide an additional discussion of related works that were omitted from the main paper due to lack of space. The recently released Perception Test [38] consists of script-based recorded videos with manual annotations focusing on 4 broad skill areas - Memory, Abstraction, Physics, Semantics, however videos are only 23s long (avg). Like Neptune, ActivityNet-RTL [23] was constructed in a semi-automatic fashion by querying GPT-4 to generate comparative temporal localization questions from the captions in ActivityNet-Captions [27]. CinePile [40] was generated by prompting an LLM to generate multiple-choice questions. Because it is based on movie clips, it can leverage available human-generated audio descriptions. Both AtivityNet-RTL and CinePile cover only limited domains and rely on existing annotations while Neptune covers a much broader spectrum of video types and its pipeline is applicable to arbitrary videos. Another recently released dataset (concurrent with our submission) is the Video-MME dataset [15]. The motivation of this dataset is similar to ours, namely it covers videos of variable lengths, with 2,700 QADs covering a wide range of different question types. The main difference between Video-MME and Neptune is that the former is entirely manually annotated by the authors, while we propose a scalable pipeline which can be applied to new videos and domains automatically, and can be tweaked to include different question types with reduced manual effort.

B The Neptune Dataset

B.1 Additional Information on Question Types

Neptune covers a broad range of long video reasoning abilities, which are summarised below. These question types are obtained in the Question and Answer generation stage, for which the prompt is provided in Sec. C.2.3. We provide further insights into the motivations of some of the question areas provided in the prompt below.
Video Summarisation: Summarise and compare long parts of the video, as well as identify the most important segments of the video.
Visual Reasoning: Recognize and understand visual elements in different parts of the video, as well as reason about why visual content is used (*e.g.* to convey a certain mood).
Temporal Ordering: Understand the timeline of events and the plot in the video.
Counting: Count objects, actions and events. Here we focus on higher-level counting where the same instance does not occur in all/every frame and actions are sufficiently dissimilar.
Cause and Effect: Understand and reason about cause and effect in the video.
Message: Understand the unspoken message that the audience may perceive after watching the video, which may require common sense knowledge to infer.
State Changes: Understand object states change over time, such as a door opening and food being eaten.

Since the questions are proposed automatically by an LLM, the question types are also generated in an open-set manner by the LLM. Hence sometimes, the LLM will generate the question type label using different phrasing - eg. 'temporal ordering' or 'timeline event'. We use simple manual postprocessing to group similar question types into the same category, with a few question types that do not fall into any of the categories grouped as 'Other'. The final *question types* released with the dataset are shown in Fig. 3 of the main paper.

B.1.1 Question Type Distribution

We explain the reasons for Neptune's current question type distribution:
(i) We prompted the LLM that generated the questions with a set of examples of different question types and let the model choose which questions to generate.
(ii) The model's selection of question types depends strongly on the given video. For example, while it is always possible to ask for a video summary, it is not always possible to ask about a person's goals, or cause and effect, because not all videos allow for these types of reasoning. This naturally leads to an imbalance of possible question types.
(iii) Additionally, we observed that the quality of questions produced by the LLM varies strongly by question type. Therefore, after quality checking by raters, the distribution changes significantly. The

strongest difference was for counting questions, as LLM-proposed questions were often too easy, e.g. counting the number of times a certain word is mentioned.

B.2 Domains in Neptune

A full graph of the domains in Neptune are provided in Fig. 5.

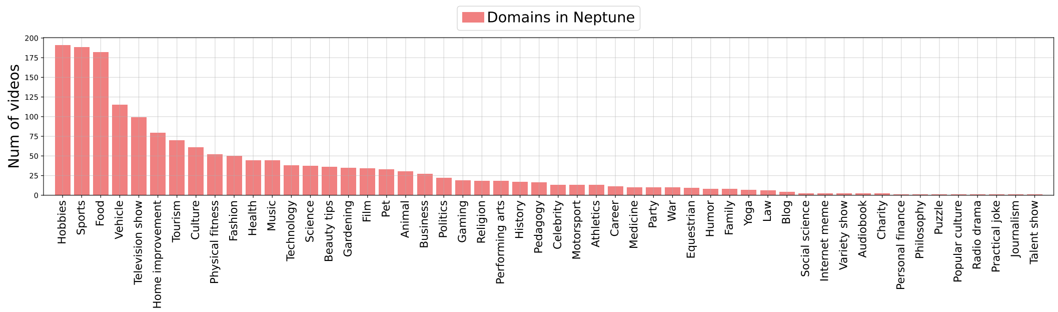


Figure 5: **Domains in Neptune:** We show the number of videos per domain category in NEPTUNE-FULL.

C Implementation Details

C.1 Video Selection

We choose the YT-Temporal-1Bn dataset [64] as the source for Neptune, because of its large and diverse corpus, and because of the high correlation between vision and audio transcripts.

Safety & Content Filters: We filter out videos with less than 100 views, that are uploaded within 90 days, and those tagged by YouTube content filters to contain racy, mature or locally controversial content. We then identify and remove static videos (eg. those that consist of a single frame with a voiceover) by clustering similar frames in a video and ensure that there is more than 1 cluster. We also identify and remove videos comprising primarily of "talking heads". To achieve this, we apply a per-frame frontal-gazing face-detector at 1fps and mark the frames where the bounding box height is greater than 20% as *talking head frames*. Then, we filter out videos where more than 30% of the frames are talking head frames. These thresholds are chosen based on an F1-score on a small dev set of 50 manually annotated videos.

Diversity Sampling: From the filtered set of videos, we sub-sample $100,000$ videos to boost both semantic and demographic diversity. First, we cluster the videos based on video-level semantic embeddings and tag each video with a cluster id. Second, we tag each video with the perceived age and gender demographic information contained in the video. Third, we obtain a joint distribution of semantics (cluster id) and demographics (perceived age and gender) and apply a diversity boost function [26] on the joint distribution. Finally, we sample from videos from this distribution. Fig. 6, shows the down-sampling of over-represented cluster ids before and after applying the filter. We then uniformly sub-sample the videos further to reach the desired dataset size.

C.2 Prompts for Data Generation

In this section we provide some of the prompts used for generating Neptune.

C.2.1 Prompt for Frame Captioning

We use the following prompt to obtain a caption for each video frame:

```
Answer the following questions about the given image. Then use the
information from the answers only, and write a single sentence as caption.
Make sure you do not hallucinate information.

Question(Mood): Describe the general mood in the image as succinctly as
possible. Avoid specifying detailed objects, colors or text.
```

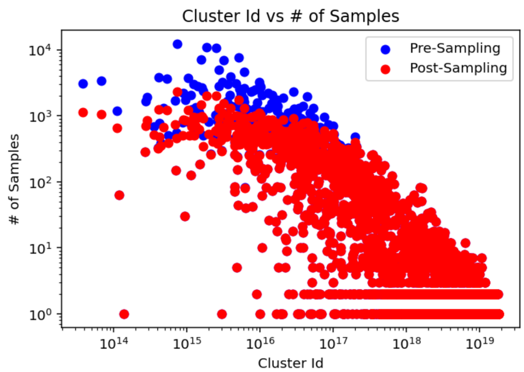


Figure 6: **Diversity sampling:** We show the change in cluster distribution after diversity sampling.

`Question(Background): Describe the background of the image as succinctly as possible. Avoid specifying detailed objects, colors or text. Eg: The background is a parking lot, playground, kitchen etc.`
`Question(Person): Is there any person in the image. If yes, describe them and what are they doing here? If no, say no person.`
`Question(General): Describe the image as succinctly as possible. Avoid specifying detailed objects, colors or text.`
`Question(Text): Is there any text? What does it say?`

`Result template:`

`Answer(Mood): A succinct description of what is happening in the image with the general mood.`
`Answer(Background): A succinct description of the background scene in the image and what is happening.`
`Answer(Person): If there are people in the image, a succinct description.`
`Answer(General): A succinct description of the image.`
`Answer(Text): Reply if there is any text, where it is placed and how it is related to what is happening in the image.`

`Caption: A couple of sentences summarizing the information given by the answers about mood, background, person, general and text.`

`With the above format as template, generate the response for the new image next.`

C.2.2 Prompts for Automatic Video Captioning

A visual overview of the video captioning stage is provided in Fig. 7. We describe the prompts for each stage below:
Shot level captions:
Using the shot boundaries the 1fps frame captions are summarized into shot level descriptions with the following prompt:
`Summarize these sentences in dense short sentences: [list of frame captions in the shot]`

Topic and Description Pairs:
If ASR exists, topic and description pairs are obtained from ASR using the following prompt:
`**Task:** Take a deep breath and give me the structural topics of the Youtube video below using the transcript. Give up to 5 Topic and Description pairs using output format. **Transcript:** transcript`

Shot Clustering:
`Take a deep breath and identify the sequential topic structure of this video using the "{head_topic}" in Scenes. A part of the video script is provided as a set of Scenes and in each scene, visual captions and`

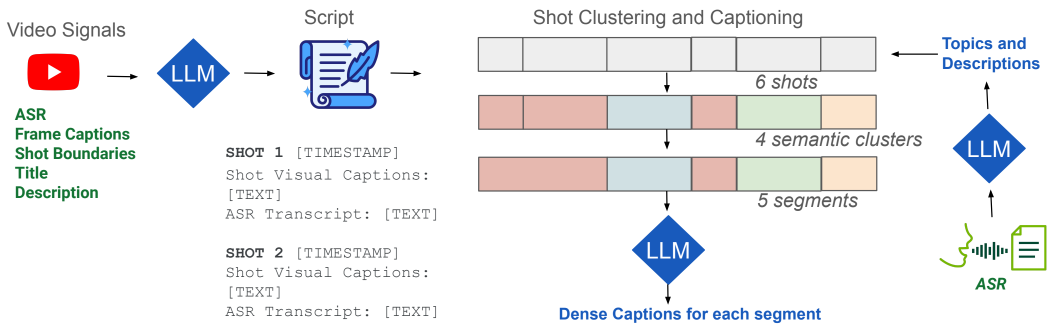


Figure 7: **Video Captioning:** We extract dense segment level captions automatically for each video. This is done by prompting an LLM using video signals (ASR, frame captions, shot boundaries and metadata) with various different steps and prompts.

transcript sentences are provided. The overall suggested structure from the transcript is provided as well. Assign every scene in this part of the script to one topic structure. For each scene, the visual captions should support and relate the topic. If the support or relation is not strong create a new topic and assign the scene to it. Reevaluate the suggested structure from the transcript and make sure all scenes are assigned to the best associated topics. Keep output length to be less than {max_output_characters} characters.

Output Format: XML output where topic has the following children (description, topic_scenes, story) <topic> <description>The description of the topic</description> <topic_scenes>Comma separated scene number(s) related to this topic<topic_scenes> <story>Summarized caption that describes what happens and what's shown for this topic in the scenes by combining visual caption and transcript sentences of the related scenes</story> </topic>

Suggested Structure: {initial_structure_from_ASR_if_exists}

Context: {summary_of_title_and_description}

Video Script: {video_script}

Segment Captions:

Consecutive shots of the same topic are then merged as one segment. Shots of the same topic that are not contiguous are treated as separate segments (see Fig. 7). We then generate dense captions for each segment using the prompt below.

Task: You are the expert in video description writing. Use the information "Partial Script" to improve the "Initial Description" by adding the missing information either from visual or transcript. The video context is also given to help you interpret the script. Only add information that is in the "Partial Script". Make the output concise and compact with less or the same length as Initial Description. The updated video description is plain text. Your answer should follow the output format. Keep output length to be less than max_output_characters characters.

Initial Description:* shared_topic_cluster_caption

Output Format: XML format like below <updated_description>updated video description text</updated_description>

Partial Script: doc_segment

Visual Support Caption

To extract better visual description of the segment that will be used for QA generation in the next phase, an extra step is performed to get visual support for each segment. That visual support is stored separately in conjunction with the dense caption for the segment. For this purpose, the dense caption from the previous step is used alongside the shot level visual captions. The following LLM prompt is used to extract the visual support:

```
**Task:** I provide video scene information and your job is to summarize
the exact elements from "Visual Captions" that directly support the "Scene
Story" of the scene below. The visuals of the scene is broken down to
shots and each shot is described in a line of text in the Visual Captions.

**Scene Story:** dense_caption_for_the_segment

**Visual Captions:** visual_captions_of_the_segment

**Output Format:** Plain text with at most 200 words summarizing the
supporting visual elements.
```

C.2.3 Generating Questions and Answers

```
I want you to act as a rigorous teacher in the "Long-term Video
Understanding" class. Let's test your students' in-depth comprehension!

Understanding: I'll provide you with the following:

- Dense Captions: A detailed breakdown of the video, including key moments
and timestamps. Analyze this carefully.

Your Task: Craft {target_number} Challenging Short-Answer Questions

Requirement:

- Challenge: Demonstrate your ability to create challenging, insightful
short-answer questions about the video. These shouldn't test simple recall
only. Aim to probe understanding of relationships, motives, subtle details,
and the implications of events within the video.
- Diversity: Design a variety of question types (more on this below).
- Specificity: Each question must be self-contained and laser-focused on
a single concept or event from the video. Avoid compound or overly broad
questions.
- Answers: Model the ideal answer format: Brief, accurate, and rooted
directly in evidence from the video's content.
- Video-Centric: Stay true to what's explicitly shown or stated in
the video. Avoid relying on outside knowledge or speculation. Design
questions so the correct answer cannot be easily determined without
carefully analyzing the video.
- Minimize Information Leakage: For question types like ranking or
ordering, ensure that the order of candidates or options listed in the
question doesn't inadvertently reveal the correct answer. Shuffle them to
maintain neutrality.
- Content-First: Timestamps and section titles within the captions are
there for guidance. Do not explicitly refer to those markers in your
questions or answers. Focus on the events and elements themselves.
- Unambiguous: Ensure each question has a single, clearly defined correct
answer. Avoid questions that are open to multiple interpretations (e.g.,
counting elements where viewers might disagree).
- Visual Elements: Questions focused on visual reasoning or visual
narratives should emphasize the interpretation of the visuals. Keep the
question minimal, letting the answer describe the specific visual elements
in detail.

You want to test students' capabilities of understanding the video,
including but not limited to the following aspects:
```

Ability: Summarize and compare long parts of the video.
Ability: Compress information from the video rather than just listing the actions that happened in the video.
Ability: Identify the most important segments of the video.
Ability: Recognize and understand the visual elements in different parts of the video.
Ability: Understand the timeline of events and the plot in the video.
Ability: Count objects, actions and events. Focus on higher-level counting where the same instance does not occur in all/every frame and actions are sufficiently dissimilar.
Ability: Understand and reason about cause and effect in the video.
Ability: Understand the unspoken message that the audience may perceive after watching the video, which may require common sense knowledge to infer.
Ability: Understand the visual reasoning of why and how important visual content is shown in the video.
Ability: Understand the visual narrative of the video and the mood of the video and which visual elements do contribute to that.
Ability: Understand object states change over time, such as door opening and food being eaten.

Presentation

- QUESTION: Introduce each question as "QUESTION 1, 2, 3: (capability) full question". - ANSWER: Follow the format "CORRECT ANSWER: correct answer".

Good example questions: - Question (counting): How many ingredients are added to the bowl in total throughout the video? Correct Answer: 3.

- Question (goal reasoning): What is the purpose of the man standing in front of the whiteboard with a diagram on it? Correct Answer: To explain the features and capabilities of the vehicle.

- Question (cause and effect): How does the document help people to be happier? Correct Answer: It helps people to identify and focus on the things that make them happy, and to develop healthy habits.

- Question (timeline event): In what order are the following topics discussed in the video: history of pantomime, importance of pantomime, mime as a tool for communication, benefits of pantomime? Correct Answer: Mime as a tool for communication, history of pantomime, importance of pantomime, benefits of pantomime.

- Question (predictive): What happens after the man jumps up and down on the diving board? Correct Answer: He jumps into the pool.

- Question (summarization): What is the overall opinion of the reviewers about Hawaiian Shaka Burger? Correct Answer: The food is good, but the patties are frozen.

- Question (creator intent): What message does the video creators try to send to the viewers? Correct Answer: Nature is essential for human well-being.

- Question (visual-temporal): What color is the scarf that Jessica wears before she enters the restaurant? Correct Answer: Red.

- Question (visual narrative): How does John's overall facial expression contribute to the explanation of the financial situation that is described in the video? Correct Answer: He shows sad feelings and expression when he described the financial collapse of the company which adds to the sense of empathy that video describes.

- Question (visual reasoning): What was shown to support the effects of a high cholesterol diet in the video? Correct Answer: Video demonstrates how cholesterol gradually clogs blood vessels, using an animation to illustrate the cross-section of vessels and the buildup of plaque.

Bad example questions because it can be answered by common sense. - Question (counting): How many players are there in a soccer team? Correct Answer: 11.

Bad example questions because it asks for trivial details. - Question (counting): How many times the word 'hurricane' is said in the video? Correct Answer: 7.

Bad example questions because the summary of topics are subjective and ambiguous. - Question (timeline event): List the sequence of topics Grace discusses in the video, starting with the earliest. Correct Answer: Getting ready for a photoshoot, attending a baseball game, showing off her new outfit, playing a Wayne's World board game, and discussing her upcoming week.

Dense Caption with Timestamps: {video_inputs_str}

C.2.4 Generating Decoys from Questions and Answers

Role: You are a rigorous teacher in a "Long-term Video Understanding" class. You will assist students in developing strong critical thinking skills. This requires creating sophisticated test questions to accompany video content.

Understanding: I will provide:

- Dense Captions: A breakdown of the video, including structure, key events, and timestamps. - Target Questions & Answers: A set of {target_number} questions about the video, along with their correct answers.

Task: Generate High-Quality Multiple-Choice Questions

1. Analyze: Carefully study the dense captions, questions, and correct answers. Familiarize yourself with the nuanced details of the video content.

2. Decoy Design: For each target question, generate {decoy_number} incorrect answers (distractors). These distractors must be:
- Challenging: Plausible to the point where students need deep content understanding and critical thinking to choose the correct answer.
- Stylistic Match: Mimic the style, tone, and complexity of the correct answer.
- Similar Length: Keep length close to that of the correct answer, preventing students from eliminating choices based on length differences.
- Factually Relevant: Related to the video content, even if slightly incorrect due to a detail change, misinterpretation, or logical fallacy.
- Reasonable: Each decoy should be something that could be true, making simple elimination impossible.

Specific Techniques for Distractor Creation

- Subtle Tweaks: Alter a minor detail from the correct answer (e.g., change a time, location, or name).
- Confusing Similarity: Use a concept from elsewhere in the video that seems related but applies to a different context.
- Misdirection: Introduce a true statement related to the video's theme but not directly answering the question.
- Order Shuffling: If the question involves the order of events, subtly rearrange the order within the distractors.

Presentation:

```
- QUESTION: Repeat the provided question faithfully (e.g., "QUESTION 1
(Capability): ...")
- CORRECT ANSWER: Repeat the correct answer (e.g., "CORRECT ANSWER: ...")
- WRONG ANSWERS: List each wrong answer on a separate line without using
letters to label choices (e.g., "WRONG ANSWER 1: ...", "WRONG ANSWER 2:
...")

*GOOD* Example: Question: What are the three main challenges that the
college is taking on? Correct Answer: Food scarcity, pollution, and
disease. Wrong Answer 1: Global warming, deforestation, and poverty.
Wrong Answer 2: Hunger, homelessness, and crime. Wrong Answer 3: Obesity,
malnutrition, and food insecurity. Wrong Answer 4: Food waste, water
shortages, and air pollution.

*BAD* examples where the decoys format is different from correct answer:
Question: What color is the shirt that the woman is wearing? Correct
Answer: Black. Wrong Answer 1: The woman is wearing a white shirt.
Wrong Answer 2: The woman is wearing a blue shirt. Wrong Answer 3: The
woman is wearing a green shirt. Wrong Answer 4: The woman is wearing a
red shirt.

*BAD* examples because only the correct answer is in positive sentiment.
Question: What is the overall sentiment of the man in the video? Correct
Answer: He is overjoyed with his new gift. Wrong Answer 1: He is upset
his gift is not big enough. Wrong Answer 2: He is sad about life in
general. Wrong Answer 3: He is upset the gift is not great. Wrong Answer
4: He seems down and unhappy.

Dense Caption with Timestamps: {video_inputs_str}

Question and Correct Answer: {question_and_answer_str}
```

C.2.5 QAD Filtering

The following prompt is used to filter out questions that can solve from QADs alone.

```
Instructions:

Carefully analyze the following question and options. Rank the options
provided below, from the most likely correct answer to the least likely
correct answer. Please respond with "ANSWER" and "EXPLANATION".

Your response should be in the following format:
ANSWER: [Letter of the ranking, split by greater than symbol. (e.g.,
"ANSWER: A > B > C > D > E")].
EXPLANATION: [Provide a brief explanation of your choice. Do not repeat
the option.]

QUESTION: {question_str}

Options: {options_str}

Please provide your response below.
```

C.3 Human Rating and Correction of QADs

We provide a screenshot of the UI used by raters to annotate automatically generated QADs in Fig. 8. Note that if any of the four options under the ‘Is the question valuable’ field are not selected, then the question is discarded from the dataset. We made sure to train raters using training raters (with detailed decks and feedback rounds), as well as applying rater replication (we used 3 raters per question independently), and rater pipelining (having an experienced rater verify the answer from a previous rater) in order to correct hallucinations and other mistakes, and discard QADs that were inappropriate. Overall, of the total 11,030 QADs that we obtained automatically, 7,762 (70%) were discarded by raters.

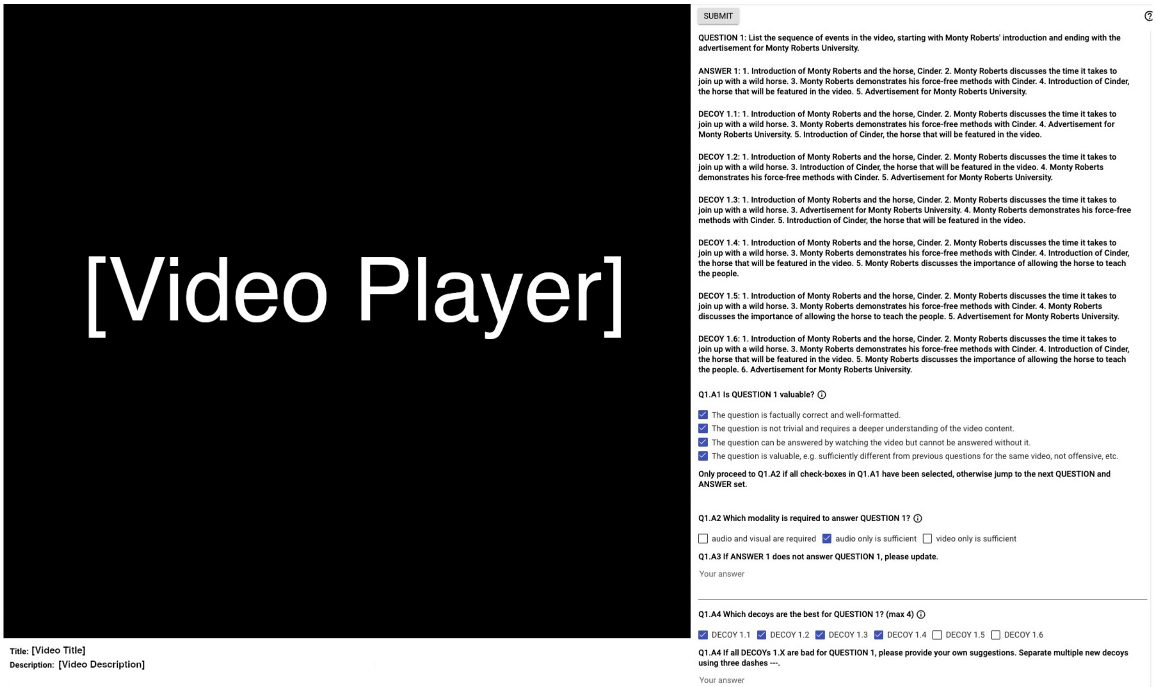


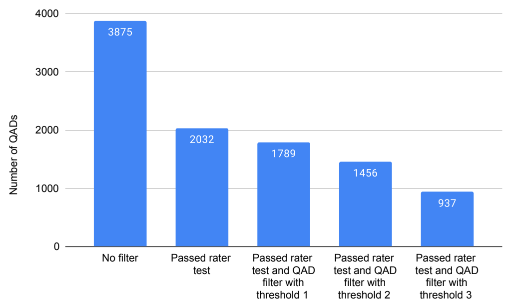
Figure 8: **Screenshot of rater UI.**

C.4 Filtering Subsets

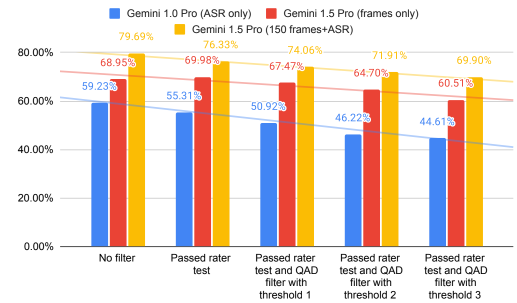
Here we provide details for how we select the thresholds used to create the NEPTUNE-MMH and NEPTUNE-MMA subsets. For both subsets, we filtered NEPTUNE-FULL with the QAD filter described in Sec. 4.4. For NEPTUNE-MMA, we additionally filtered out QADs that human raters marked as requiring only the audio modality and answer (see Sec. 4.5). We refer to this as the "rater test". For NEPTUNE-MMH, we instead applied the ASR filter (Sec. 4.4). Both QAD and ASR filters were run by prompting an LLM (Gemini 1.0 Pro) three times, each time with a different random seed and then removing QADs that the LLM answered correctly at least X out of three times, where X is the threshold for the test.

Fig. 9 shows how choosing different thresholds affects dataset size and accuracy scores. The top row shows the choices for the NEPTUNE-MMH subset. Raters marked almost half of the questions as answerable from audio only, so the rater filter already cuts the dataset size in half. Successively applying the QAD filter with increasing thresholds reduces data size up until less than 25%. We benchmark three models on the different subsets that have access to ASR only, vision only, or both vision and ASR, respectively. As expected, all three models show declining performance, with the ASR-only model showing the biggest losses. This suggests that all models were inferring the correct answer from the QAD only, which the filter successfully mitigates. The vision-only model gains slightly from removing QADs that fail the rater rest, which is expected as the test removes QADs that rely on audio, which the model does not have access to. However, like for the other models, its accuracy declines when adding the QAD test.

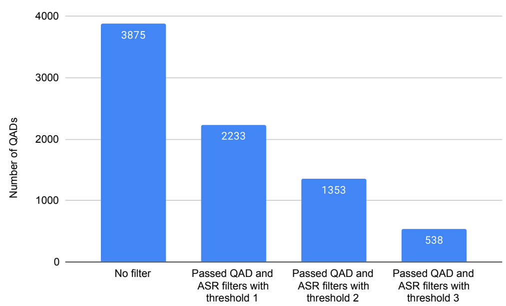
The bottom row of Fig. 9 shows the choices for the NEPTUNE-MMA subset where we use the ASR filter and the QAD filter with identical thresholds. This filter set has a stronger effect on the dataset size, reducing it to less than 15% of its original size at the highest threshold. Because the ASR-only model was used for the ASR filter, we exclude it from the accuracy comparison. The vision-only and vision+ASR models both show declining accuracy with increasing thresholds. As expected, the accuracy of the vision+ASR model declines faster. The effect of this filter set on the accuracy is much stronger than that of the above filter set, suggesting that it increases the difficulty of the dataset more strongly. Even the vision-only model declines faster than above, suggesting that this filter set generally removes easier questions, even those that rely on vision only.



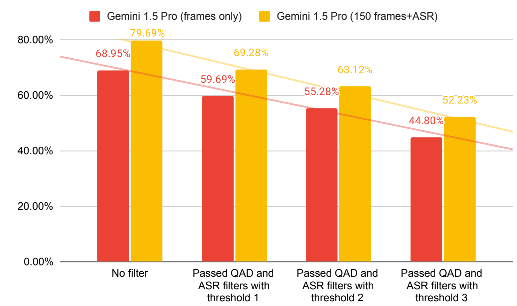
(a) Effect of rater and QAD filters on dataset size



(b) Effect of rater and QAD filters on accuracy scores



(c) Effect of ASR and QAD filters on dataset size



(d) Effect of ASR and QAD filters on accuracy scores

Figure 9: Effect of filtering thresholds for the NEPTUNE-MMH (top row) and NEPTUNE-MMA (bottom row) subsets.

For both filtered sets, we opted to set the threshold to two, which in both cases significantly increases the dataset difficulty while still preserving enough QADs for statistically meaningful evaluation metrics. We noticed that when setting the threshold to three, there were less than five QADs left for some question types, preventing robust accuracy estimation for these tasks.

C.5 Implementation Details for Benchmarks

C.5.1 Blind Baselines

For the Gemini-1.5-pro baseline with text only the prompt used was: “Carefully analyze the question and all available options then pick the most probable answer for this question”

C.5.2 Video-LLaVA

For Video-LLaVA the following prompt was used - "Pick a correct option to answer the question. Question: question Options: options ASSISTANT:".

C.5.3 VideoLLama2

During inference, we uniformly sampled 8 frames from each video. Each frame undergoes padding and resizing to a standardized dimension. The pre-processed frames are then fed into the image encoder. These steps are set as default in the inference script provided by videoLlama2.

QAD Prompt: _PROMPT_TEMPLATE = """Pick a correct option number to answer the question. Question: {question} Options: {options}:"""

OE Prompt: Question: {question}

Output post processing: We eliminated extra characters and spaces using regex to get the final ID of the predicted option.

C.5.4 MiniGPT4-Video

We set the 300 maximum number of output tokens to be 300 for the open-ended task and 10 for the multiple choice eval. The prompts are as follows:

_PROMPT_TEMPLATE_MCQ = """Question: select the correct option for this task: question Options: options. Output format: [OPTION]: [Reason]"""

_PROMPT_TEMPLATE_OPEN_ENDED = """Question: question Answer:"""

C.5.5 MA-LMM

We set the 300 maximum number of output tokens to be 300 for the open-ended task and 300 for the multiple choice eval. The prompts are as follows:

_PROMPT_TEMPLATE_MCQ = """Question: select the best choice for this task: question Options: options Answer:"""

_PROMPT_TEMPLATE_OPEN_ENDED = """Question: question Answer:"""

C.6 Compute Resources

The compute heavy part of the project was image frame captioning (as this involves reading high dimensional pixel data). The rest of the pipeline involves largely text-only LLMs and hence was less compute heavy. We estimate that the entire project in total took roughly 256 TPU v5e running over a period of 50 days.

D Additional details for GEM

D.1 Creation of GEM equivalence dev set

To create a development set that allows us to estimate the accuracy of different open-ended question answering metrics on Neptune, we sampled 97 question-answer pairs from the dataset and generated 3 candidate answers per question by prompting VideoLLAVA [31], Gemini-1.5-pro [41] and MA-LMM [21] to write a free-form answer for each question without looking into the decoys or ground truth. We then manually annotated these responses between 0 and 1 by comparing it to the ground truth answer. We made sure that the annotators are blind to the model to avoid any bias. The resulting set has 292 equivalence pairs with an average score of 0.32, with 85 examples having score greater 0.5 and 206 examples with score less than 0.5

D.2 Benchmarking on the dev set

In Table. 2, we evaluate several open-ended metrics on our dev set. The task of the metric is to classify whether the open-ended response and ground-truth answer are equivalent or not. We report F1-scores to balance false-positives and false-negatives. We evaluate both traditional rule-based metrics such as CIDEr and ROUGE-L, as well as established model-based metrics such as BEM[6]. We also try using Gemini-1.5-pro [41] as an LLM based equivalence metric (by prompting it to estimate equivalence). First, we note that as expected, Gemini-1.5-pro correlates well with the human ground-truth annotation of the set, achieving a high F1-score of 72.5. However, given that Gemini is not open-source and proprietary, any change in the model can affect all the prior results in an external leader-board making it challenging as a metric. Traditional rule-based metrics perform much worse than Gemini-1.5-pro on this dev set as they are n-gram based and struggle to handle the diversity of domains and styles in the open-ended responses. The BERT model based BEM metric [6] performs similarly, achieving an F1-score of 61.5.

Next, we evaluate lightweight open-source language models Gemma-2B [45], Gemma-7B [45] and Gemma-9B [46] in a zero-shot setting and find that performance improves with model size, with Gemma-9B bridging the gap well between traditional metrics and the Gemini-1.5-pro based metric. Finally, we fine-tune Gemma-9B on the open-source BEM answer equivalence dataset [6], and find that Gemma-9B finetuned on the BEM dataset performs the best on our dev-set. We name this metric *GEM*.

D.3 Implementation Details

We use instruction-tuned variants of the Gemma models (gemma-it-2b, gemma-it-7b and gemma-it-9b) for our experiments. To develop a prompt, we experiment with several variations in a zero-shot setting and measure the performance on the dev-set. Our final prompt is shown below. To ensure responses occur in a standard format, we simply measure the softmax-probability over "TRUE" response indicating the statements are equivalent and "FALSE" response indicating the statements are not equivalent. For each model, the threshold over probability is chosen to maximize the F-1 score on dev set. To finetune Gemma models on BEM dataset, we tokenize the same prompt as used in the zero-shot setting and train it using prefix-LM tuning for 10000 iterations using a learning rate of $1e-6$. For evaluation, we truncate the open-ended responses to 100 words, use a decode cache size of 1024 and threshold the softmax probability of the LM using the chosen threshold from dev-set.

```
<start_of_turn>user
Answer Equivalence Instructions:

Carefully consider the following question and answers.
You will be shown a "gold-standard" answer from a human annotator,
referred to as the "Reference Answer" and a "Candidate Answer".
Your task is to determine whether the two answers are semantically
equivalent.

In general, a candidate answer is a good answer in place of the "gold"
reference if both the following are satisfied:
  1. The candidate contains at least the same (or more) relevant information
     as the reference,taking into account the question; in particular it
     does not omit any relevant information present in the reference.
  2. The candidate contains neither misleading or excessive superfluous
     information not present in the reference, taking into account the
     question.

Your response should be one word, "TRUE" or "FALSE", in the following
format:
ANSWERS_ARE_EQUIVALENT: [TRUE or FALSE]

Question:
"{}"

Candidate Answer:
"{}"

Reference Answer:
"{}"

Please provide your response below.
<end_of_turn>
<start_of_turn>model
ANSWERS_ARE_EQUIVALENT:
```

D.4 Qualitative Examples for Metric

Below, we provide some qualitative examples of our metric applied to the predictions provided by the baseline models Video LLaMA2[11], Gemini-1.5-flash [41] and Gemini.1.5-pro [41]. We find that Gemini-1.5-flash responses are qualitatively slightly better than Gemini-1.5-pro responses as they contain less hallucinations. Furthermore, we find that with a single frame as input, Gemini-1.5-pro usually responds with a variation of "This information is not present is the provided frame" instead of making an educated guess. This is unlike the case of multiple choice questions where the model commits to one of the choices. These aspects are reflected in the results presented in Table.3. We also find that the metric fails often in the questions about listing items / events in order. In such cases, the model is biased towards accepting the response (false positives) even if the order is incorrect provided the items / events are present in the candidate response.

```
Question: Compare the Pampered Chef Veggie Strip Maker to a food
    processor in terms of their effectiveness in making cauliflower
    rice.
Answer: The Veggie Strip Maker is more effective than a food
    processor in making cauliflower rice because it prevents the
    cauliflower from becoming mushy.
Video LLaMA2: Both the Pampered Chef Veggie Strip Maker and a food
    processor can be effective in making cauliflower rice, but they
    differ in their methods of operation and the resulting texture of
    the rice.    The Pampered Chef Veggie Strip Maker is a kitchen
    appliance designed specifically for making vegetable strips,
    including cauliflower rice. It uses a grating mechanism to create
    rice-like pieces by shredding the cauliflower. This method can
    produce a relatively uniform and rice-like texture.    On the other
    hand, a food processor is a versatile kitchen appliance that can
    also be used to make ...
GEM Equivalence: False
Gemini-1.5-flash: The Pampered Chef Veggie Strip Maker makes
    cauliflower rice that is less mushy than a food processor.
GEM Equivalence: True
Gemini-1.5-pro: The Pampered Chef Veggie Strip Maker is more
    effective than a food processor because it doesn't make the
    cauliflower mushy.
GEM Equivalence: True


Question: How does the use of behavioral training help in the
    process of taking foot x-rays on elephants?
Answer: Behavioral training encourages elephants to willingly
    participate in their healthcare, making the process of taking foot
    x-rays less stressful for both the elephants and the caretakers.
Video LLaMA2: The use of behavioral training plays a crucial role in
    the process of taking foot x-rays on elephants. By training the
    elephants to follow commands and stand on a platform, it allows the
    caretakers to safely and effectively capture the x-rays of the
    elephants' feet. This training helps to ensure that the elephants
    remain still and cooperative during the x-ray process, minimizing
    any potential harm or discomfort. Additionally, the behavioral
    training helps to build a positive relationship between the
    elephants and their caretakers, fostering trust and cooperation.
    Overall, the use of behavioral training enhances the accuracy and
    efficiency of the
GEM Equivalence: True
Gemini-1.5-flash: Behavioral training allows the elephants to
    willingly participate in the training sessions and participate in
    their own health care.
GEM Equivalence: True
Gemini-1.5-pro: Behavioral training allows the elephants to
    willingly participate in the x-ray process by presenting their feet
    for imaging.
GEM Equivalence: True

Question: What are the key features of the Adam Purvis Progeny
    knife's blade?
Answer: The Adam Purvis Progeny knife's blade is made of 3.4-inch
    M390 steel. The blade shape is useful, simple, and highly effective,
```

the edge is very sharp, featuring a stonewash finish on the primary bevels and a horizontal satin on the flats. It is exceptionally sharp and has a 150 thousandths stock thickness, making it suitable for everyday carry tasks.

Video LLaMA2: The Adam Purvis Progeny knife's blade is made of S35VN steel, which is known for its high strength and excellent edge retention. The blade has a flat ground and a hollow grind, which allows for a thin and sharp edge. Additionally, the blade has a black and silver color scheme, giving it a sleek and modern appearance.

GEM Equivalence: False

Gemini-1.5-flash: The Adam Purvis Progeny knife's blade is made of M390 steel, has a modified sheepsfoot style, and is ground very thinly for a sharp edge.

GEM Equivalence: False

Gemini-1.5-pro: The Adam Purvis Progeny knife features a 3.4-inch M390 steel blade with a modified sheepsfoot shape, thin grind, sharp edge, and a stonewashed finish on the bevels with a satin finish on the flats.

GEM Equivalence: True

Question: How does interacting with civilians in nearby villages enhance cadets' leadership skills?

Answer: Engaging with civilians in nearby villages allows cadets to develop respectful communication and relationship-building skills, which are essential for effective leadership in various situations.

Video LLaMA2: Interacting with civilians in nearby villages enhances cadets' leadership skills by providing them with real-world experiences and opportunities to apply their knowledge and training in a practical setting. By engaging with the local community, cadets learn to communicate effectively, build trust, and foster positive relationships with people from diverse backgrounds. This hands-on experience helps them develop their interpersonal skills, cultural awareness, and problem-solving abilities, all of which are crucial for successful leadership in various situations. Additionally, interacting with civilians allows cadets to gain a deeper understanding of the challenges and needs of the community, which can inform their decision-making and ...

GEM Equivalence: True

Gemini-1.5-flash: Interacting with civilians in nearby villages teaches cadets how to speak to people respectfully and build trust.

GEM Equivalence: True

Gemini-1.5-pro: It teaches them how to speak to people.

GEM Equivalence: False

Question: List the sequence of topics discussed in the video, starting with the earliest.

Answer: 1. Introduction of the GAN356 AIR cube and its unique features. 2. Discussion of the GAN356 AIR's weight and GES (Gan Elasticity System) springs. 3. Lubrication, tensioning, and personal experience with the GAN356 AIR's performance. 4. Assessment of the GAN356 AIR's performance, including its smoothness, speed, corner-cutting, and tendency to lock up. 5. Overall evaluation of the GAN356 AIR, including its strengths and weaknesses, and recommendations for potential buyers.

Video LLaMA2: 1. Introduction to the video and the Rubik's Cube. 2. Explanation of the different types of Genuine Handcubes. 3.

```
    Demonstration of how to solve the Rubik's Cube, including the
    various algorithms and techniques used.  4. Showcasing of the
    different colors and types of Rubik's Cubes available.  5.
    Conclusion and closing credits.
GEM Equivalence: False
Gemini-1.5-flash: The video starts with an introduction of the GAN
    356 Air cube, then discusses its features, including the GES system
    and its light weight. The reviewer then shares their opinion on the
    different GES springs and their effect on the cube's feel. The video
    continues with a discussion on the cube's performance, including its
    smoothness, speed, and tendency to lock up. Finally, the reviewer
    concludes with a recommendation for viewers who might be interested
    in purchasing the cube.
GEM Equivalence: True
Gemini-1.5-pro: The video discusses the main features of the Gan 356
    Air, the different types of springs, the weight and price of the
    cube, the reviewer's experience with the cube, and the reviewer's
    overall opinion of the cube.
GEM Equivalence: False
```

```
Question: In what order are the following events mentioned in the
    video: introduction of the wallet, features of the wallet, and care
    instructions?
Answer: Introduction of the wallet, features of the wallet, care
    instructions.
Video LLaMA2: The video begins with the introduction of the wallet,
    followed by a description of its features, and concludes with
    instructions on how to care for the wallet.
GEM Equivalence: True
Gemini-1.5-flash: The speaker introduces the wallet, then describes
    its features, and finally mentions the care instructions.
GEM Equivalence: True
Gemini-1.5-pro: Introduction of the wallet, care instructions,
    features of the wallet.
GEM Equivalence: False
```

E Model-Based Temporal Certificate

The idea of a temporal certificate was introduced by EgoSchema [34] as a way of capturing the intrinsic temporal understanding length for *long video* QA datasets. It is defined as 'the length of the video a human verifier needs to observe to be convinced of the veracity of the marked annotation'. While the authors used it to uncover flaws in existing long video QA datasets, as well as to provide a difficulty measure independent of video length, we find that is has the following drawbacks: (i) it does not take into account the *length of time* or the *effort* taken by the annotator themselves, to find the correct time span in videos; (ii) it requires manual annotation from expert annotators to measure; and finally (iii) is subjective.

As an attempt to mitigate these issues, we introduce a slightly modified version of the temporal certificate, which is *Model-Based*. We calculate this certificate using 129 samples from Neptune and EgoSchema, respectively. For this experiment we used Gemini 1.5 Pro, with one "driver" model run to answer the question and two other model runs with different random seeds to verify if the answer was not correct by random chance. Along with the question and options, we provided video clips of various lengths from the center of the video, and at various fps, as shown in Fig. 10.

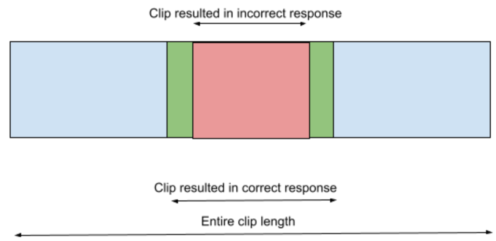


Figure 10: **Model-based Temporal Certificate:** Illustration of video clip querying for the model-based temporal certificate experiment. The red clip is the clip length that resulted in an incorrect response. As we increased the clip length wider, and the model correctly answered the question, we logged the frame count for incorrect response and correct response, and stopped querying. Besides clip length, we vary the fps of the query clip.

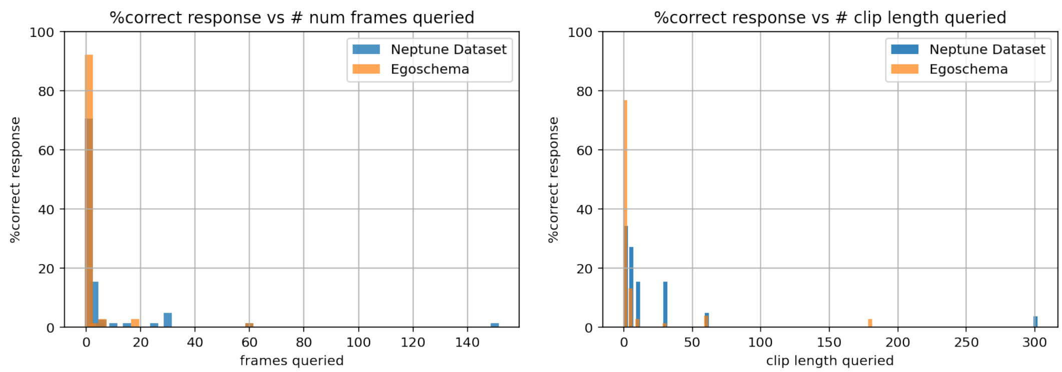


Figure 11: **Frame level temporal certificate:** We compared our dataset sample with EgoSchema to evaluate the number of frames needed by model to answer questions correctly. The figures above show the distribution of the minimum number of frames required to achieve the correct response.

Since this experiment queried a set of frames over various clip lengths, we defined it as the "needle in haystack" problem. Here, the needle is defined as a frame or set of frames needed to answer the question correctly, matching a human's ground truth response, while the haystack is a set of frames which need to be watched to find the needle frames. Iteratively, we increase the video length and fps for the query until the model achieves the correct response.

As shown in Fig. 11, we find that the model needs more frames to answer the question correctly for the Neptune dataset as compared to EgoSchema. This resulted in a mean of 5.39 as certificate frames for Neptune which is 3.37 times the mean certificate frame number of 1.6 for EgoSchema. On the clip length level this translated to a mean of 21.22s of clip needed to respond correctly on the Neptune dataset, whereas for EgoSchema the mean was 9.07s. The model-based certificate lengths turn out to be much smaller than the certificate lengths reported by EgoSchema, where humans needed close to 100s to answer the questions for EgoSchema.

In addition, we define the *effort score* as the fraction of the maximum number of frames needed to be watched before answering the question correctly, as defined in Equation 1. An effort score closer to 0 suggests that the needle isn't very small compared to the haystack, i.e. most of the frames contain the answer to the question; while a high effort score means a high percentage of haystack frames needs to be included before we cover all frames required to answer correctly.

$$\text{EFFORT SCORE} = \frac{\text{MAX NUMBER OF FRAMES RESULTING IN AN INCORRECT RESPONSE}}{\text{MIN NUMBER OF FRAMES RESULTING IN A CORRECT RESPONSE}} \quad (1)$$

For Neptune, the mean effort score was 0.47, whereas for EgoSchema, it was 0.19. This suggests that Neptune requires 2.47 times the effort compared to EgoSchema according to the definition above, which closely corroborates the above results for the mean clip lengths needed to solve the questions from the respective datasets.

F Societal Impact

Our data may match the distribution of videos and text on the internet. As such, it will mirror known biases on that source of data. For at least this reason, this data set should not be used for training models and is only intended for academic evaluation purposes. Risks regarding biases are detailed in the datasheet provided in Sec. H. To create the dataset, we run large Gemini models, which has a negative externality of energy usage and carbon emissions. For benchmarking, we use existing models. These models are likely to inherit the biases of the data distribution and the pre-trained weights used in their original training.

G Responsibility Statement

We the authors of this work, bear full responsibility for any violations of rights arising from this submission. We also confirm that the released dataset is under Creative Commons Attribution 4.0 (CC BY 4.0)[6] license. All hosting, maintenance and licensing is the responsibility of the authors and is outlined in the datasheet provided in Sec. H.

H Datasheet for Neptune

Datasheets for datasets introduced by Gebru et al. [17] serve as a medium of communication between the creators and users of a dataset. They effectively consolidate the motivation, creation process, composition, and intended uses of a dataset as a series of questions and answers. In this Section, we provide a datasheet for the Neptune dataset.

Motivation

Q1. **For what purpose was the dataset created?** *Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.*

The Neptune dataset was created to test long video understanding via the medium of video question answering. This fills a gap in the current set of benchmarks available for video understanding, as most datasets are still focused on short-form video clips. We believe Neptune proposes a significant challenge for, and hence can provide key insights for the development of VLMs applied to long video understanding.

Q2. **Who created this dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?**

The Neptune dataset was created by researchers at Google LLC.

Q3. **Who funded the creation of the dataset?** *If there is an associated grant, please provide the name of the grantor and the grant name and number.*

The Neptune dataset was funded by Google LLC.

Q4. **Any other comments?**

No.

Composition

Q5. **What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)?** *Are there multiple types of instances (e.g., movies, users, and ratings; people and interactions between them; nodes and edges)? Please provide a description.*

Each instance in Neptune represents a YouTube video, and annotations in the form of question, answer and 4 decoys (QAD).

[6]http://creativecommons.org/licenses/by/4.0

Q6. **How many instances are there in total (of each type, if appropriate)?**

There are 3,268 instances in Neptune.

Q7. **Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?** *If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable).*

Neptune is a small sample drawn from all the data uploaded to YouTube. Millions of videos are uploaded on YouTube every day. We start from a subset of YouTube video candidates from the YT-Temporal-1Bn dataset [62], which is biased towards videos where the ASR has a strong correlation with the visual content. Our final subset (2,405 videos) was created after a number of filtering and processing stages, which aim to increase the diversity of samples in the data. Hence the Neptune data does not fully represent the distribution of videos uploaded to YouTube.

Q8. **What data does each instance consist of? "Raw" data (e.g., unprocessed text or images) or features?** *In either case, please provide a description.*

Each instance in Neptune consists of four metadata fields:

- `"video_id"`: Unique alphanumeric ID of the video (assigned by YouTube).
- `"url"`: Static URL for downloading the video, e.g., `https://www.youtube.com/watch?v=<video_id>`.
- `"question"`: A hard question about the video.
- `"answer"`: An answer to the above question.
- `"decoys"`: Four decoy answers intended to be used in conjunction with the question and answer for multiple choice evaluation.

Q9. **Is there a label or target associated with each instance?** *If so, please provide a description.*

We provide QADs, though it might also be also possible to use auxiliary information (like video titles or tags).

Q10. **Is any information missing from individual instances?** *If so, please provide a description, explaining why this information is missing (e.g., because it was unavailable). This does not include intentionally removed information, but might include, e.g., redacted text.*

No and yes. No, because all the metadata fields for every instance are filled with valid values. Yes, because the `"url"` for some instances may not retrieve the underlying video. This may happen if the YouTube user (author) removes the video from YouTube. Such deletions reduce our dataset size over time, however, video deletions are rare.

Q11. **Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)?** *If so, please describe how these relationships are made explicit.*

Relationships between individual instances (e.g., videos made by the same creator) are not made explicit in our work, though this is a possibility for future work.

Q12. **Are there recommended data splits (e.g., training, development/validation, testing)?** *If so, please provide a description of these splits, explaining the rationale behind them.*

Our entire dataset is meant to be a test set only. We provide challenging subsets (MMA and MMH) focusing on videos that require multimodal understanding by LLMs and humans, respectively.

Q13. **Are there any errors, sources of noise, or redundancies in the dataset?** *If so, please provide a description.*

Neptune may be noisy as each QAD annotation was created automatically by machine learning methods first. While each annotation has been checked by a human rater, there may be some unavoidable human error in the annotation process.

Q14. **Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)?** *If it links to or relies on external resources,*

(a) Are there guarantees that they will exist, and remain constant, over time?

(b) Are there official archival versions of the complete dataset (i.e., including the external resources as they existed at the time the dataset was created)?

(c) *Are there any restrictions (e.g., licenses, fees) associated with any of the external resources that might apply to a future user? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, as appropriate.*

The dataset relies on videos hosted on YouTube. We do not distribute videos of our dataset to respect YouTube's terms of service. Instead, we provide video URLs (`"url"`, Q8) that point to videos hosted on YouTube servers. In response to sub-questions:
(a) There are no guarantees that videos will remain available on YouTube.
(b) There are no archived versions of the dataset.
(c) We refer to the YouTube ToS for details on content restrictions https://www.youtube.com/static?template=terms.

Q15. **Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor-patient confidentiality, data that includes the content of individuals non-public communications)?** *If so, please provide a description.*

No, the videos included in Neptune do not cover topics that may be considered confidential. All videos were publicly shared on YouTube prior to inclusion in Neptune.

Q16. **Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?** *If so, please describe why.*

While we cannot be certain that our raters were able to remove all videos mentioned above, we explicitly ask them to do so in their annotation guidelines. We also note that YouTube removes videos that contain offensive content or do not follow their community guidelines.

Q17. **Does the dataset relate to people?** *If not, you may skip remaining questions in this section.*

The dataset pertains to people in that people have uploaded the videos to YouTube. Furthermore, most videos in Neptune have people speaking and/or appearing.

Q18. **Does the dataset identify any subpopulations (e.g., by age, gender)?** *If so, please describe how these subpopulations are identified and provide a description of their respective distributions within the dataset.*

Neptune does not explicitly identify any subpopulations. Since most videos contain people and questions are free-form natural language questions produced by automatic machine learning models, it is possible that some annotations may identify people appearing in individual videos as part of a subpopulation.

Q19. **Is it possible to identify one or more natural persons, either directly or indirectly (i.e., in combination with other data) from the dataset?** *If so, please describe how.*

Yes, our data includes celebrities, or other YouTube-famous people. All of the videos that we use are of publicly available data, following the Terms of Service (`https://www.youtube.com/static?template=terms`) that users agreed to when uploading to YouTube.

Q20. **Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)?** *If so, please provide a description.*

This is highly unlikely, as YouTube removes videos that contain offensive content or do not follow their community guidelines.

Q21. **Any other comments?**

No.

Collection Process

Q22. **How was the data associated with each instance acquired?** *Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part-of-speech tags, model-based guesses for age or language)? If data was reported by subjects or indirectly inferred/derived from other data, was the data validated/verified? If so, please describe how.*

See Q7 for an explanation of how the candidate video IDs were chosen. These video IDs were provided by the YT-Temporal-1Bn dataset providers [62] and then filtered by our automatic filtering process decribed in the main paper. The `"video_id"` and `"URL"` are directly observable from YouTube. The annotations were obtained by machine learning models and then verified by human annotators.

Q23. **What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)?** *How were these mechanisms or procedures validated?*

We collected all data using compute resources provided by Google LLC. The code involved running various machine learning models to obtain the annotations. The code was validated by checking several data samples from Neptune. All annotations were then verified by human raters.

Q24. **If the dataset is a sample from a larger set, what was the sampling strategy?**

See Q7.

Q25. **Who was involved in data collection process (e.g., students, crowd-workers, contractors) and how were they compensated (e.g., how much were crowd-workers paid)?**

The first few stages of our data collection pipeline are fully automatic and do not require any human annotators. The final stage involved contractor raters.

Q26. **Over what timeframe was the data collected? Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)?** *If not, please provide a description of the timeframe.*

We collected all data in early 2024. As mentioned in Q22, videos are sampled from the YT-Temporal-1Bn dataset [62] which was collected in 2021 so videos are uploaded no later than 2021.

Q27. **Were any ethical review processes conducted (e.g., by an institutional review board)?** *If so, please provide a description of these review processes, including the outcomes, as well as a link or other access point to any supporting documentation.*

We did not conduct a formal ethical review process via institutional review boards. However, as described in Section 4.1 and Q16 we employed several filtering mechanisms to tag instances that could be problematic.

Q28. **Does the dataset relate to people?** *If not, you may skip remaining questions in this section.*

Yes, see Q17.

Q29. **Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)?**

We collected data submitted by YouTube users indirectly through the YouTube API. However, users agree with YouTube's Terms of Service regarding the redistribution of their data by YouTube.

Q30. **Were the individuals in question notified about the data collection?** *If so, please describe (or show with screenshots or other information) how notice was provided, and provide a link or other access point to, or otherwise reproduce, the exact language of the notification itself.*

Users were not notified about to the use of their data in our dataset. However, by uploading their data on YouTube, they consent that it would appear on the YouTube platform and will be accessible via the official YouTube API (which we use to collect Neptune).

Q31. **Did the individuals in question consent to the collection and use of their data?** *If so, please describe (or show with screenshots or other information) how consent was requested and provided, and provide a link or other access point to, or otherwise reproduce, the exact language to which the individuals consented.*

See Q30.

Q32. **If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses?** *If so, please provide a description, as well as a link or other access point to the mechanism (if appropriate).*

Users have full control over the presence of their data in our dataset. If users delete the underlying YouTube video – it will be automatically removed from Neptune since we distributed videos as URLs.

Q33. **Has an analysis of the potential impact of the dataset and its use on data subjects (e.g., a data protection impact analysis) been conducted?** *If so, please provide a description of this analysis, including the outcomes, as well as a link or other access point to any supporting documentation.*

No.

Q34. **Any other comments?**

No.

Preprocessing, Cleaning, and/or Labeling

Q35. **Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)?** *If so, please provide a description. If not, you may skip the remainder of the questions in this section.*

After human raters annotated the dataset, we ran the following postprocessing and cleaning stages - blind and ASR cleaning. Details are provided in the main paper.

Q36. **Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)?** *If so, please provide a link or other access point to the "raw" data.*

No, we only provide the filtered data.

Q37. **Is the software used to preprocess/clean/label the instances available?** *If so, please provide a link or other access point.*

The exact model used is not available, but we explain the process in detail in the paper, allowing other researchers to reproduce our pipeline.

Q38. **Any other comments?**

No.

Uses

Q39. **Has the dataset been used for any tasks already?** *If so, please provide a description.*

We have only used the dataset to evaluate machine learning models as described in the paper.

Q40. **Is there a repository that links to any or all papers or systems that use the dataset?** *If so, please provide a link or other access point.*

We do not maintain such a repository. However, citation trackers like Google Scholar and Semantic Scholar would list all future works that cite our dataset.

Q41. **What (other) tasks could the dataset be used for?**

The dataset can only be used to evaluate video question answering.

Q42. **Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?** *For example, is there anything that a future user might need to know to avoid uses that could result in unfair treatment of individuals or groups (e.g., stereotyping, quality of service issues) or other undesirable harms (e.g., financial harms, legal risks) If so, please provide a description. Is there anything a future user could do to mitigate these undesirable harms?*

This is very difficult to anticipate. Future users of our dataset should be aware of YouTube's user demographics which might subtly influence the types of videos, languages, and ideas that are present in the dataset. Also, note that our dataset is mainly composed of English videos, hence models trained on this dataset might perform worse on videos in other languages.

Q43. **Are there any tasks for which the dataset should not be used?** *If so, please provide a description.*

Broadly speaking, our dataset should only be used for evaluating models. Our dataset should not be used for any tasks that involve identifying features related to people (facial recognition, gender, age, ethnicity identification, etc.) or making decisions that impact people (mortgages, job applications, criminal sentences; or moderation decisions about user-uploaded data that could result in bans from a website).

Q44. **Any other comments?**

No.

Distribution

Q45. **Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created?** *If so, please provide a description.*

Yes, our dataset is publicly available.

Q46. **How will the dataset will be distributed (e.g., tarball on website, API, GitHub)** *Does the dataset have a digital object identifier (DOI)?*

We distribute our dataset as a JSON file containing annotations hosted on GitHub. Users will have to download the videos by themselves. All uses of Neptune should cite the paper as the reference.

Q47. **When will the dataset be distributed?**

The dataset is publicly available at https://github.com/google-deepmind/neptune.

Q48. **Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)?** *If so, please describe this license and/or ToU, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms or ToU, as well as any fees associated with these restrictions.*

Uses of our dataset are subject to YouTube API terms (`https://www.youtube.com/static?template=terms`). This data is licensed by Google Inc. under a Creative Commons Attribution 4.0 International License. Users are allowed to modify and repost it, and we encourage them to analyze and publish research based on the data.

Q49. **Have any third parties imposed IP-based or other restrictions on the data associated with the instances?** *If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms, as well as any fees associated with these restrictions.*

The videos corresponding to our instances are legally owned by YouTube users. Use of the dataset and containing videos are subject to YouTube ToS (`https://www.youtube.com/static?template=terms`).

Q50. **Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?** *If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any supporting documentation.*

No.

Q51. **Any other comments?**

No.

Maintenance

Q52. **Who will be supporting/hosting/maintaining the dataset?**

The authors will maintain the dataset. The dataset is hosted on Google cloud. All the information about the dataset, including links to the paper and future announcements will be accessible at the dataset website on GitHub.

Q53. **How can the owner/curator/manager of the dataset be contacted (e.g., email address)?**

The contact emails of authors are available on the dataset website.

Q54. **Is there an erratum?** *If so, please provide a link or other access point.*

There is no erratum for our initial release. We will version all errata as future releases (Q55) and document them on the dataset website.

Q55. **Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)?** *If so, please describe how often, by whom, and how updates will be communicated to users (e.g., mailing list, GitHub)?*

We will update our dataset periodically and announce updates on the dataset website. These future versions would remove instances that were requested to be removed via the opt-out form (Q32).

Q56. **If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were individuals in question told that their data would be retained for a fixed period of time and then deleted)?** *If so, please describe these limits and explain how they will be enforced.*

Rather than directly distributing videos, we distribute URLs that point to the original videos uploaded by YouTube users. This means that users retain full control of their data – any post deleted from YouTube will be automatically removed from Neptune (see also Q10, Q14, Q31).

Q57. **Will older versions of the dataset continue to be supported/hosted/maintained?** *If so, please describe how. If not, please describe how its obsolescence will be communicated to users.*

A new version release of Neptune will automatically deprecate its previous version. We will only support and maintain the latest version at all times. We decided to deprecate old versions to ensure that any data that is requested to be removed (Q32) will be no longer accessible in future versions.

Q58. **If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?** *If so, please provide a description. Will these contributions be verified? If so, please describe how. If not, why not? Is there a process for communicating/distributing these contributions to other users? If so, please provide a description.*

Anyone can extend Neptune by using our automatic pipeline described in the main paper.