# Language Repository for Long Video Understanding

**Kumara Kahatapitiya, Kanchana Ranasinghe, Jongwoo Park, Michael S. Ryoo**
Stony Brook University

## Abstract

Language has become a prominent modality in computer vision with the rise of LLMs. Despite supporting long context-lengths, their effectiveness in handling long-term information gradually declines with input length. This becomes critical, especially in applications such as long-form video understanding. In this paper, we introduce a Language Repository (LangRepo) for LLMs, that maintains concise and structured information as an interpretable (*i.e.*, all-textual) representation. Our repository is updated iteratively based on multi-scale video chunks. We introduce write and read operations that focus on pruning redundancies in text, and extracting information at various temporal scales. The proposed framework is evaluated on zero-shot visual question-answering benchmarks, showing state-of-the-art performance at its scale. Our code is available at `github.com/kkahatapitiya/LangRepo`.

## 1 Introduction

Video data is central to learning systems that can interact and reason about the world. Yet, they also associate with significant challenges such as increased compute requirements and redundant information, to name a few. This is especially critical in long-form videos. Even so, recent literature on video understanding have progressed so far, enabling reasoning capabilities in hours-long video streams (Team et al., 2023; Islam et al., 2024), in contrast to very-limited temporal spans (*e.g.* seconds or minutes) just a few years ago. Work on efficient spatio-temporal attention mechanisms (Arnab et al., 2021; Bertasius et al., 2021), memory management (Wu et al., 2022; Ryoo et al., 2023), and large-language-models (LLMs) (Wang et al., 2022a; Yu et al., 2024; Team et al., 2023) have been key ingredients for such improvements.

LLMs, or more-specifically, vision-large-language-models (VLLMs) have been outperforming pure vision models in recent years in all facets, including image-based reasoning (Liu et al., 2024; Zheng et al., 2024a; Li et al., 2023b), grounding (Lai et al., 2023; Rasheed et al., 2023), video understanding (Wang et al., 2022a; Ye et al., 2023; Yu et al., 2024), and even robotics (Zeng et al., 2022; Ahn et al., 2022; Liang et al., 2023; Li et al., 2024b). The sheer model scale and the vast pretraining data have enabled such frameworks to capture world knowledge and semantics, beyond what is possible with visual data only. Besides, the ability to process long context-lengths is also key, as it helps modeling long-term dependencies that are crucial for more-complex reasoning and interactions. However, recent studies show that despite the availability of such context-lengths, the effectiveness of models declines with longer input sequences (Levy et al., 2024). This promotes the search for alternate representations that can compress input language data without losing meaningful information, essentially managing the context utilization of LLMs.

Moreover, the use of text (*i.e.*, language) in modeling has shown numerous benefits such as rich semantics (Wang et al., 2022b; Menon and Vondrick, 2022; Kahatapitiya et al., 2023), ease of information sharing between different specialized-models (Zeng et al., 2022) or modalities (Liu et al., 2024; Girdhar et al., 2023), and interpretability (Zhao et al., 2023a; Singh et al., 2024). Among such, interpretability has a huge societal impact in the age of LLMs, to manage adversities such as bias (Liang et al., 2021; Ferrara, 2023) and hallucinations (Zhang et al., 2023b; Dhuliawala et al., 2023). Simply put, it enables human observers to understand and monitor what really happens within models. Hence, interpretable representations have also been of interest to the community, in place of latent representations (Wu et al., 2022; Ryoo et al., 2023).

Motivated by the above, we introduce Language Repository (LangRepo), an *all-textual* (hence, interpretable) representation for LLMs that updates



*Findings of the Association for Computational Linguistics: ACL 2025*, pages 5627–5646
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

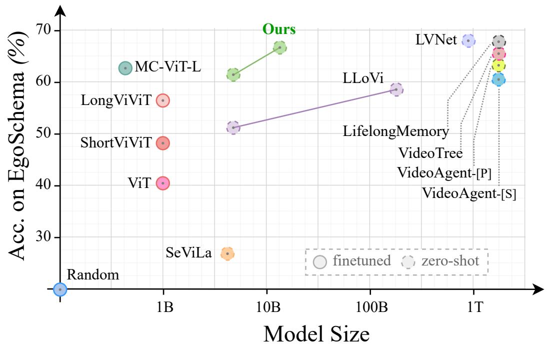



Figure 1: Comparison with prior-art on EgoSchema (Mangalam et al., 2024) subset: `LangRepo` (ours) outperforms finetuned and zero-shot pipelines of similar scale, while being competitive with much-larger proprietary models. Note the log-scale in x-axis.

iteratively. It consumes input captions corresponding to video chunks and relies on text-based operations to write and read information. The write operation (`write-to-repo`) prunes redundant text, creating concise descriptions that keep the context-utilization of LLMs in-check. Its iterative application with increasingly-longer chunks enables it to learn high-level semantics (*e.g.* long temporal dependencies). The read operation (`read-from-repo`) extracts such stored language information at various temporal scales, together with other optional metadata within the repository. Altogether, our proposed framework is applied to long-term video reasoning tasks such as visual question-answering (VQA) on EgoSchema (Mangalam et al., 2024), NExT-QA (Xiao et al., 2021) and IntentQA (Li et al., 2023a), showing strong performance at its scale (see Fig. 1).

## 2 Related work

**Long-video understanding:** Video models have progressed over the years, going from primitive recognition tasks (Kuehne et al., 2011; Soomro et al., 2012) to complex and fine-grained reasoning tasks (Sigurdsson et al., 2016; Grauman et al., 2022) over long horizons. Both convolutional baselines (Carreira and Zisserman, 2017; Feichtenhofer et al., 2019) and transformer architectures (Arnab et al., 2021; Bertasius et al., 2021) have explored research directions such as multi-scale representations (Fan et al., 2021; Liu et al., 2022), efficiency concerns associated with heavy spatio-temporal computations (Li et al., 2019; Duke et al., 2021), and handling redundant information within video inputs (Chen et al., 2018; Kahatapitiya and Ryoo, 2021). More recently, long-video understanding has made a leap forward thanks to benchmark datasets (Xiao et al., 2021; Mangalam et al., 2024) and model improvements (Zhang et al., 2023a; Yu et al., 2024), validating the importance of modeling complex interactions that happen over long periods of time. Still, the sub-par performance of SOTA models on such benchmarks suggests the potential for further improvements.

**Long-context models:** Even before the age of LLMs, models based on convolutions (Wang et al., 2018; Piergiovanni and Ryoo, 2018, 2019), recurrent blocks (Chung et al., 2014; Greff et al., 2016; Hutchins et al., 2022) or transformers (Wu et al., 2022; Ryoo et al., 2023) have exploited long-term dependencies, especially in the context of video understanding (Wang et al., 2018; Wu et al., 2022) and robotics (Chen et al., 2021; Shang et al., 2022). With the rise of LLMs, scaling laws have revealed the importance of longer contexts even more (Team et al., 2023; Reid et al., 2024), and, thanks to the breakthroughs such as sparse processing (Shazeer et al., 2017; Fedus et al., 2022), caching (Khandelwal et al., 2018; Ge et al., 2023), model-sharding (Lepikhin et al., 2020; Zhao et al., 2023b), and efficient attention (Dao et al., 2022; Lefaudeux et al., 2022), such long-context LLMs have become a reality. Even with very large context lengths, effectively reasoning over longer inputs is challenging (Xiong et al., 2023; Shi et al., 2023; Levy et al., 2024). This motivates us to think about concise representations that can better-utilize LLM context.

**Compressing representations:** When handling heavy inputs, deep learning models have relied on compressed representations. It may come in the form of pruning (Ryoo et al., 2021; Bolya et al., 2022), latent memory (Graves et al., 2014; Wu et al., 2022; Ryoo et al., 2023), or external feature banks (Wu et al., 2019), to name a few. Despite the efficiency gains of such techniques, it is challenging to realize which information gets preserved, and how semantically-meaningful they are, post-compression. A compressed representation that is also interpretable, may shed light on such details.

**Language as an interpretable modality:** More-recently, language has emerged as a dominant modality in computer vision due to its strong generalization capabilities (Radford et al., 2021; Jia et al., 2021). It can act as a bridge between various domain-specific models (Zeng et al., 2022), other modalities (Girdhar et al., 2023; Liu et al., 2024), and even human instructions (Surís et al., 2023; Gupta and Kembhavi, 2023), showing intriguing

| Dataset | Captions per-video | | |
|---|---|---|---|
| | $0.5\times$ | $1\times$ | $2\times$ |
| EgoSchema (Mangalam et al., 2024) | 49.8 | 48.8 | 46.8 |
| NExT-QA (Xiao et al., 2021) | 48.2 | 48.2 | 46.9 |
| IntentQA (Li et al., 2023a) | 47.1 | 46.9 | 45.2 |

Table 1: **Observations on increasing input length:** We evaluate the VQA performance of an LLM (Jiang et al., 2023) at different input lengths, on multiple long-video benchmarks. Even with a sufficient context length, the effectiveness of predictions decreases with longer input. Here, $1\times$ corresponds to captions generated at a standard frame-rate (and, $0.5\times/2\times$ corresponds to a compression/expansion by a factor of 2).

applications in domains such as chat agents (*e.g.* ChatGPT, Bard) and robotics (Ahn et al., 2022; Liang et al., 2023). It has also supported two-stage VQA pipelines with an intermediate text modality (Ma et al., 2024; Liao et al., 2024; Li et al., 2023c). Since language is interpretable, it enables humans to interact with models naturally and make sense of model predictions.

Motivated by the above, we introduce an interpretable language representation that can (1) prune redundant information, and (2) extract multi-scale (or, high-level) semantics, enabling better context-utilization within LLMs.

## 3 Observations on Long-range Inputs

In this section, we investigate how LLMs perform with increasing inputs lengths (*i.e.*, #tokens). Recent LLMs with very-large context lengths such as Gemini-Pro-1.5 (Team et al., 2023) (1M tokens) or Claude-2.1 (200k tokens), can support extremely long input sequences. Yet, when feeding longer inputs, the reasoning capabilities (especially, long-term reasoning) of such models diminish. This behavior is also observed in concurrent work (Levy et al., 2024), and evident in benchmark results of state-of-the-art models (Ye et al., 2023; Yu et al., 2024) (*i.e.*, better performance with shorter inputs, or fewer video frames). To better investigate this in our setup, we evaluate VQA performance on standard long-term video understanding benchmarks while varying the input length (see Table 1). We consider frame/short-clip captions extracted using a VLLM at a baseline framerate ($1\times$) as inputs (introduced in (Zhang et al., 2023a)). We either subsample ($0.5\times$) or replicate ($2\times$) the captions, decreasing/increasing the input lengths of a question-answering LLM, namely, Mistral-7B (Jiang et al., 2023) with 8k (or, theoretical 128k) context length. All inputs fit within the context, without any overflow. The observation from this study is consistent: even though the context length of the LLM is sufficient to process given inputs, the effectiveness of its predictions (shown by VQA performance) drops with longer inputs (see Table A.1a for more details). This motivates us to introduce a concise language representation that preserves important details of long-range inputs, while pruning any redundant information.

## 4 Language Repository

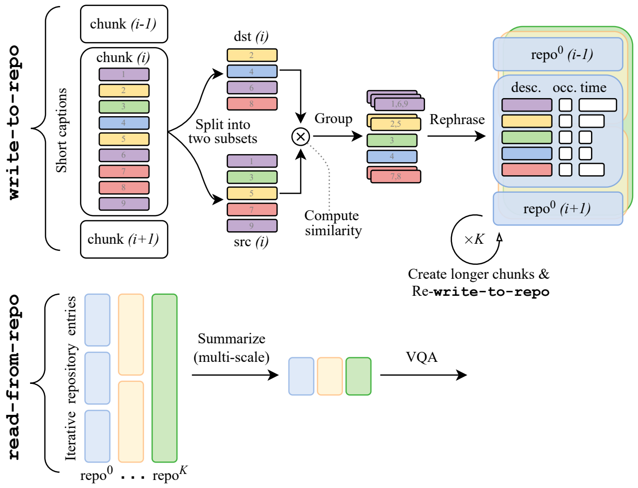
We present a Language Repository (`LangRepo`) that iteratively updates with multi-scale descriptions from video chunks. In contrast to external feature banks (Wu et al., 2019) or learnable latent memory representations (Wu et al., 2022; Ryoo et al., 2023; Balažević et al., 2024), our proposal has a few key advantages: (1) it requires no training (*i.e.*, zero-shot), and (2) it is compatible with both LLM-based processing and human interpretation, as it is fully-textual, *i.e.*, it exists in language-space instead of a latent-space. `LangRepo` consists of two main operations: (1) information writing (`write-to-repo`), which prunes redundancies and iteratively updates language descriptions based on increasingly-longer video chunks, and (2) information reading (`read-from-repo`), which extracts preserved descriptions (with any optional metadata) in multiple temporal scales. We show a detailed view of these operations in Fig. 2, and further elaborate in the following subsections.

Consider a long video that is split in to $n$ non-overlapping chunks, denoted as $V = \{v_i \mid i = 1, \cdots, n\}$. Assume that we already have frame or short-clip captions extracted by a VLLM (*e.g.* LLaVA (Liu et al., 2024)) corresponding to such chunks, denoted by $C^0 = \{c_i^0 \mid i = 1, \cdots, n\}$. Here, each chunk may consist of $p$ such captions as in $c_i^0 = \{c_{ij}^0 \mid j = 1, \cdots, p\}$. Altogether, $V$ is represented by $n \times p$ captions which we consider as inputs to our framework.

### 4.1 Writing to repository

We intend to create a concise, all-textual representation with multiple scales (or, semantic-levels) of information. Hence, our writing operation is text-based, and applied iteratively on different scales of input. In the first iteration, it consumes low-level details in each chunk $i$, in the form of captions $c_i^0$, generating initial entries to the repository $\text{repo}^0(i)$, or $r_i^0$.

$$r_i^0 = \texttt{write-to-repo}(c_i^0)\ . \tag{1}$$




**Algorithm** Long-video VQA pipeline in LangRepo

**require** captions of video chunks $C^0 = \{c_i^0 \mid i = 1, \cdots, n\}$, number of iterations $K$.

**def** `write-to-repo`$(c)$:
  $c_{\text{dst}},\ c_{\text{src}} = \texttt{split}(c)$
  $\text{sim}_{\text{src-dst}} = \texttt{similarity}(\texttt{encode}(c_{\text{src}}),\ \texttt{encode}(c_{\text{dst}}))$
  $c_{\text{grp}} = \texttt{group}(c_{\text{dst}},\ c_{\text{src}},\ \text{sim}_{\text{src-dst}})$
  $c_{\text{reph}} = \texttt{rephrase}(\texttt{template}_{\texttt{reph}}(c_{\text{grp}}))$
  $r = (c_{\text{reph}},\ t,\ o)$ *// t: timestamps, o: occurances*
  **return** $r$

**def** `read-from-repo`$(r)$:
  $d = \texttt{summarize}(\texttt{template}_{\texttt{sum}}(r))$
  **return** $d$

$r_i^0 = \texttt{write-to-repo}(c_i^0)$
$d_i^0 = \texttt{read-from-repo}(r_i^0)$

**for** $k$ in range$(K)$: *// iterative write and read*
  $C^{k+1} = \texttt{re-chunk}([\cdots,\ r_i^k,\ \cdots])$
  $r_{i'}^{k+1} = \texttt{write-to-repo}(c_{i'}^{k+1})$
  $d_{i'}^{k+1} = \texttt{read-from-repo}(r_{i'}^{k+1})$

ans $= \texttt{question-answer}(q, [\ d_i^0, \cdots])$ *// q: query*

Figure 2: **Detailed view of our Language Repository (LangRepo):** Here we present the write and read operations within LangRepo. Given short-captions corresponding to video chunks, `write-to-repo` first prunes redundant captions within each chunk. The same process is iteratively applied on increasingly longer (or, higher-level) chunks— that are already within the repository— to generate multi-scale repository entries. Pruning consists of two stages: (1) grouping most similar captions based on embedding (*e.g.* CLIP (Radford et al., 2021)) similarities between two subsets, and (2) rephrasing grouped captions with an LLM-call. The resulting LangRepo will include rephrased-captions and any optional metadata (*e.g.* #occurrences, timestamps). Next, `read-from-repo` generates concise descriptions for different semantic levels by summarizing the multi-scale language representation, which is also an LLM-call.

In each subsequent iteration $k+1$, previous repo entries of iteration $k$ are re-combined into longer chunks and processed in the same way, generating information for higher semantic-levels.

$$[c_1^{k+1}, \cdots, , c_m^{k+1}] = \texttt{re-chunk}([r_1^k, \cdots, r_n^k]) \ , \quad (2)$$

$$r_{i'}^{k+1} = \texttt{write-to-repo}(c_{i'}^{k+1}) \ . \quad (3)$$

Here, $\texttt{re-chunk}(\cdot)$ denotes the creation of longer (and, fewer, *i.e.*, $m < n$) chunks within the repository. More specifically, we simply concatenate (denoted by $[\cdot]$) all entries from previous iteration, and split them again into fewer number of chunks (hence, longer chunk size). Note that $i'$ in the above equation is not the same as the previous chunk indexing $i$, as we may have different (usually, fewer) number of chunks in each subsequent iteration. Each write operation involves two stages: (1) Grouping redundant text, and (2) Rephrasing, which are detailed below.

**Grouping redundant text:** Given textual descriptions of a video chunk (*i.e.*, captions in the first write iteration, or previous repo descriptions in subsequent iterations), we plan to identify most-similar ones and merge them as a single description. Without loss of generality, let us consider the first write iteration, for which the input is in the form of $c_i^0 = \{c_{ij}^0 \mid j = 1, \cdots, p\}$. Inspired by (Bolya et al., 2022), we first split the captions of each chunk into two sets, namely, source (*src*) captions $c_{\text{src},i}^0$ and destination (*dst*) captions $c_{\text{dst},i}^0$. Let us drop the chunk index $(i)$ and iteration index $(0)$ for brevity. Here, dst captions $c_{\text{dst}}$ are sampled uniformly distributed across the temporal span of a chunk, while all the rest are considered as src captions $c_{\text{src}}$ (see Fig. 2 top-left).

$$c_{\text{dst}},\ c_{\text{src}} = \texttt{split}(c) \ . \quad (4)$$

Here, we usually have fewer dst captions (*i.e.*, $|c_{\text{dst}}| < |c_{\text{src}}|$). Next, we embed all captions using a text-encoder (*e.g.* CLIP (Radford et al., 2021)), and compute the cosine similarity of each pair between src-dst sets to find most-similar matches.

$$\text{sim}_{\text{src-dst}} = \texttt{similarity}(\texttt{encode}(c_{\text{src}}),\ \texttt{encode}(c_{\text{dst}})) \ . \quad (5)$$

Based on the similarity matrix above ($\text{sim}_{\text{src-dst}}$), we then prune the highest $x\%$ similarities by grouping such source captions with their corresponding destination matches, forming a set of grouped descriptions $c_{\text{grp}}$ for the given chunk. Refer to the color-coded captions after 'Group' in Fig. 2.

$$c_{\text{grp}} = \texttt{group}(c_{\text{dst}},\ c_{\text{src}},\ \text{sim}_{\text{src-dst}}) \ . \quad (6)$$

Here, an additional hyperparameter (*i.e.*, $x$) decides the grouping ratio. Finally, such grouped descriptions go through a rephrasing operation prior to entering the repository.

Rephrase ($\texttt{template}_{\texttt{reph}}$)

```
[INST] <<SYS>> You are a helpful expert in first person view video analysis. <</SYS>>

You are given a list of ${num_of_groups} language descriptions for a first person view video. Each sentence
describes a ${clip_length}s clip. Here are the descriptions as a list: ${grouped_captions}.

Please summarize and rephrase each item in the list as a single sentence of ${num_words} words. Keep the
same original subject. Keep all information intact without leaving anything out. Return only the rephrased list of
${num_of_groups} descriptions in the same order, without additional details. [/INST]
The rephrased list is as follows:
```

Summarize ($\texttt{template}_{\texttt{sum}}$)

```
[INST] <<SYS>> You are a helpful expert in first person view video analysis. <</SYS>>

You are given some language descriptions of a first person view video. The video is ${duration}s long.
The descriptions cover the whole video exactly. Here are the descriptions: ${language_repository_entry}.

Please give me a ${num_words} words summary. When doing summarization, remember that your summary
will be used to answer this multiple choice question: ${question} [/INST]
```

Figure 3: **LLM prompt templates in LangRepo:** Here, we show the zero-shot prompt templates used for rephrasing ($\texttt{template}_{\text{reph}}$) and summarizing ($\texttt{template}_{\text{sum}}$) operations. Rephrase prompt needs a list of grouped captions as input, while its output adheres to more-strict requirements (*e.g*. same order, same number of list items) needed for correct parsing. Summarize prompt takes in each repository entry and generates a more-flexible (*i.e*., open-ended) output, while optionally conditioning on the question.

**Rephrasing:** Grouped captions $c_{\text{grp}}$ of each chunk are rephrased via an LLM-call. This allows redundant information within each group to be dropped, while generating a concise and coherent description. We first form a list of grouped captions, where each list item corresponds to a single group (*i.e*., a dst caption and any one or more src captions matched to it), and feed it to the LLM, wrapped in a rephrasing-template ($\texttt{template}_{\text{reph}}$) as shown in Fig. 3 (top).

$$c_{\text{reph}} = \texttt{rephrase}(\texttt{template}_{\text{reph}}(c_{\text{grp}}))\ . \tag{7}$$

Here, the LLM output ($c_{\text{reph}}$) is restricted to be a list in the same order with the same number of items, where each item is a single concise sentence. Finally, such rephrased descriptions together with other metadata such as timestamps ($t$) and number of occurrences ($o$) are written in the repository.

$$r = \{(c_{\text{reph},j},\ t_j,\ o_j) \mid j = 1,\ \cdots,\ p'\}\ . \tag{8}$$

Note that here $p' < p$ as we have grouped and rephrased a pre-defined ratio (*e.g*. 50%) of most-similar captions. Alongside each description in a repository entry, $t$ maintains a list of timestamps corresponding to its founding captions, whereas the occurrences counter ($o$) keeps track of the number of captions grouped together. A qualitative example of a repository entry is given in Fig. 4.

In subsequent iterations, the same operations apply when writing multi-scale entries. The only difference is the change in input, which now constitutes of previous repo entries re-combined into high-level chunks (*i.e*., $c^0 \to c^k$). Each new iteration generates information corresponding to a higher semantic-level (*i.e*., going from short-range to long-range dependencies), forming our multi-scale language representation.

### 4.2 Reading from repository

As we make a single VQA prediction for a given long video— instead of making predictions every chunk— our read operation (`read-from-repo`) is applied after fully-forming each scale of multi-scale repository (*i.e*., after writing all chunks). The repo entries from $K$ scales can be denoted as $\{r^k \mid k = 0, \cdots, K\}$ where each scale ($r^k$) may consist of multiple entries $\{\cdots,\ r^k_{i-1},\ r^k_i,\ r^k_{i+1},\ \cdots\}$. When reading, we generate summaries for each entry in the repo separately, allowing it to focus on varying temporal spans. More specifically, each entry goes through a summarizing-template ($\texttt{template}_{\text{sum}}$) as shown in Fig. 3 (bottom), and the resulting prompt is fed to the LLM.

$$\begin{aligned} d^k_i &= \texttt{read-from-repo}(r^k_i) \\ &= \texttt{summarize}(\texttt{template}_{\text{sum}}(r^k_i))\ . \end{aligned} \tag{9}$$

Here, $d^k_i$ corresponds to the output description of each entry $i$ in the repository, at the respective scale $k$. Optionally, we can make use of additional metadata such as timestamps and #occurrences, by prompting the read operation with descriptions of repo entries formatted as "`[timestamps] description (×#occurrences)`" (see Fig. 4). Finally, we concatenate all output descriptions and prompt the LLM again to generate the answer.

$$\text{ans} = \texttt{question-answer}(q,\ [\cdots, d^k_i, \cdots])\ . \tag{10}$$

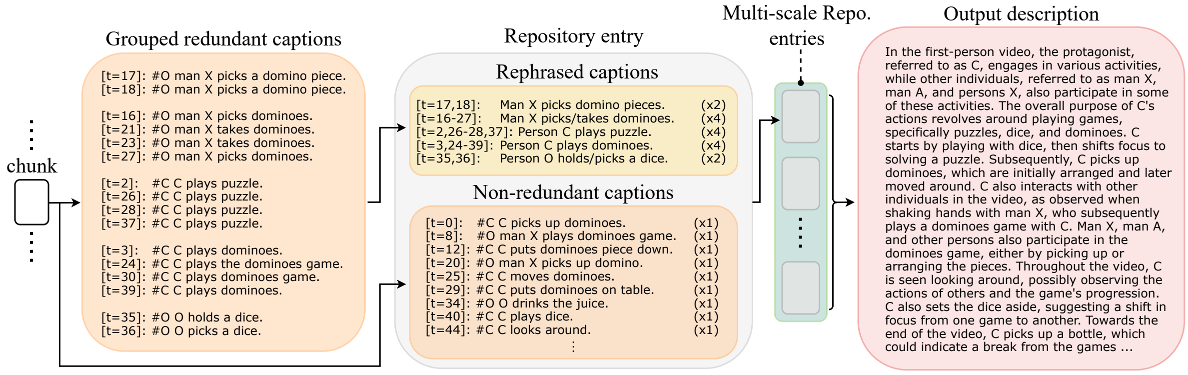



Figure 4: **A qualitative example of a LangRepo entry:** Given a video chunk, redundant captions are first grouped together during pruning operation. During rephrasing, such groups are more-concisely written to the repository, along with additional metadata. Other non-redundant captions are written directly. This process is continued iteratively with increasingly-longer chunks, creating multi-scale repository entries (refer Fig. A.1 for a more-detailed view). Finally, such descriptions from various temporal scales are read to generate the output.

## 5 Experiments

In our experiments, we rely on captions pre-extracted using VLLMs, as given in (Zhang et al., 2023a). As for the LLM, we use either Mistral-7B (Jiang et al., 2023) (w/ 7B parameters) or Mixtral-8×7B (Jiang et al., 2024) (w/ 12B active parameters) by default. As the text encoder in similarity-based pruning, we use CLIP-L/14 (Radford et al., 2021). Note that all the models used in our framework are open-source and within a reasonable model-scale, making our work accessible even in academic settings. We do zero-shot inference on all datasets without any finetuning, evaluating the performance on long-form video VQA benchmarks.

For evaluations, we consider three challenging long-video VQA benchmarks in our evaluations. EgoSchema (Mangalam et al., 2024) derived from Ego4D (Grauman et al., 2022), consists of 3-minute long clips, each with a question and 5 answer-choices. Its public validation subset consists of 500 videos, whereas the held-out fullset has 5K videos. NExT-QA (Xiao et al., 2021) contains videos up to 2 minutes long (at an average of 44 seconds), annotated with 52k open-ended questions and 48k close-ended questions (*i.e.*, multiple-choice with 5 answer options). The questions belong to either temporal, causal, or descriptive categories, evaluating different reasoning capabilities of models. We consider zero-shot evaluation on the validation set. IntentQA (Li et al., 2023a) is based on the same NExT-QA videos, yet focuses more on intent-related questions (*e.g.* why?, how? or before/after) with a total of 16k multiple-choice questions on 4.3k videos. Here, we consider zero-shot setting on the test set.

### 5.1 Main results

**EgoSchema:** In Table 2 (left), we present the VQA performance of LangRepo on standard EgoSchema (Mangalam et al., 2024) splits, comparing with other state-of-the-art frameworks. Here, we focus on zero-shot evaluation. We consider Mistral-7B (Jiang et al., 2023) and Mixtral-8×7B (Jiang et al., 2024) as the choice of LLMs in our setup, both with reasonable model scales (7B and 12B active parameters, respectively). We de-emphasize the comparisons with multi-modal LLMs that use video-caption pretraining. LangRepo shows significantly-better performance compared to other methods at a similar scale, validating its effectiveness. We achieve +7.8% on fullset over mPLUG-Owl (Ye et al., 2023), +12.0% on subset over pure Mistral LLM baseline (Jiang et al., 2023), +10.0% on subset and +5.4% on fullset over LLoVi (7B) (Zhang et al., 2023a) (w/ Mistral (Jiang et al., 2023)), +4.5% on fullset over Vamos (Wang et al., 2023) (w/ Llama2 (Touvron et al., 2023)), and +4.8% on subset over Tarsier (7B) (Wang et al., 2024a).

**NExT-QA:** In Table 2 (right), we report the performance of LangRepo on standard NExT-QA (Xiao et al., 2021) validation set. On zero-shot evaluation, our framework outperforms other methods consistently. Compared to smaller models, we gain +11.8% over InternVideo (Wang et al., 2022a) and +9.4% over VFC (Momeni et al., 2023). Compared to models of similar scale, we gain +3.5% over baseline Mistral LLM (Jiang et al., 2023) and +2.7% over LLoVi (12B) (Zhang et al., 2023a). We de-emphasize the comparisons with multi-modal LLMs pretrained with video captions.

| Model | Params | Subset | Fullset |
|---|---|---|---|
| *with proprietary LLMs* | | | |
| Vamos (Wang et al., 2023) | 175B | - | 41.2 |
| LLoVi (Zhang et al., 2023a) | 175B | 57.6 | 50.3 |
| ProViQ (Choudhury et al., 2023) | 175B | - | 57.1 |
| MoReVQA (Min et al., 2024) | 340B | - | 51.7 |
| LVNet (Park et al., 2024) | <1.8T | 68.2 | 61.1 |
| Vamos (Wang et al., 2023) | 1.8T | - | 48.3 |
| VideoAgent-[S] (Wang et al., 2024b) | 1.8T | 60.2 | 54.1 |
| VideoAgent-[P] (Fan et al., 2024) | 1.8T | 62.8 | - |
| IG-VLM (Kim et al., 2024) | 1.8T | - | 59.8 |
| VideoTree (Wang et al., 2024e) | 1.8T | 66.2 | 61.1 |
| LifelongMemory (Wang et al., 2024d) | 1.8T | 68.0 | 62.1 |
| *with open-source LLMs* | | | |
| InternVideo (Wang et al., 2022a) | 478M | - | 32.1 |
| FrozenBiLM (Yang et al., 2022) | 890M | - | 26.9 |
| SeViLA (Yu et al., 2024) | 4B | 25.7 | 22.7 |
| mPLUG-Owl (Ye et al., 2023) | 7B | - | 31.1 |
| Mistral (Jiang et al., 2023) | 7B | 48.8 | - |
| LLoVi (Zhang et al., 2023a) | 7B | 50.8 | 33.5 |
| Tarsier (Wang et al., 2024a) | 7B | 56.0 | 49.9 |
| VideoLLaMA 2 (Cheng et al., 2024) | 12B | - | 53.3 |
| Vamos (Wang et al., 2023) | 13B | - | 36.7 |
| InternVideo2 (Wang et al., 2024c) | 13B | - | 60.2 |
| Tarsier (Wang et al., 2024a) | 34B | 68.6 | 61.7 |
| LangRepo (ours) | 7B | 60.8 | 38.9 |
| LangRepo (ours) | 12B | 66.2 | 41.2 |

| Model | Params | NExT-QA | IntentQA |
|---|---|---|---|
| *with proprietary LLMs* | | | |
| ViperGPT (Surís et al., 2023) | 175B | 60.0 | - |
| ProViQ (Choudhury et al., 2023) | 175B | 64.6 | - |
| MoReVQA (Min et al., 2024) | 340B | 69.2 | - |
| LVNet (Park et al., 2024) | <1.8T | 72.9 | 71.1 |
| IG-VLM (Kim et al., 2024) | 1.8T | 68.6 | 64.2 |
| LLoVi (Zhang et al., 2023a) | 1.8T | 67.7 | 64.0 |
| TraveLER (Shang et al., 2024) | 1.8T | 68.2 | - |
| VideoAgent-[S] (Wang et al., 2024b) | 1.8T | 71.3 | - |
| VideoTree (Wang et al., 2024e) | 1.8T | 73.5 | 66.9 |
| *with open-source LLMs* | | | |
| VFC (Momeni et al., 2023) | 164M | 51.5 | - |
| InternVideo (Wang et al., 2022a) | 478M | 49.1 | - |
| SeViLA (Yu et al., 2024) | 4B | 63.6 | 60.9 |
| Mistral (Jiang et al., 2023) | 7B | 51.1 | 50.4 |
| LLoVi (Zhang et al., 2023a) | 7B | 54.3 | 53.6 |
| LLoVi (Zhang et al., 2023a) | 12B | 58.2 | 56.6 |
| Tarsier (Wang et al., 2024a) | 34B | 79.2 | - |
| LangRepo (ours) | 7B | 54.6 | 53.8 |
| LangRepo (ours) | 12B | 60.9 | 59.1 |

Table 2: **Main results** (Left) on EgoSchema (Mangalam et al., 2024), and (Right) on NExT-QA (Xiao et al., 2021) and IntentQA (Li et al., 2023a): We focus on the zero-shot video VQA. LangRepo shows a strong performance at its scale. Open-source multi-modal LLMs with video-caption pretraining are de-emphasized for fair comparison.

**IntentQA:** In Table 2 (right), we evaluate our zero-shot framework against other state-of-the-art models on IntentQA (Li et al., 2023a) test set. LangRepo outperform comparable models with similar scale consistently, showing gains of $+3.4\%$ over baseline Mistral LLM (Jiang et al., 2023) and $+2.5\%$ over LLoVi (12B) (Zhang et al., 2023a).

We include evaluations on additional video VQA benchmarks in the supplementary (see Sec. A.4).

### 5.2 Ablation study

**Choice of backbone LLM, text encoder and classifier:** We ablate the choice of LLM-backbones within the framework in Zhang et al. (2023a) in Table 3a. We observe that Mistral-7B (Jiang et al., 2023) is significantly better at video reasoning compared to LLama2-13B (Touvron et al., 2023). Next, we consider different text encoders to embed our text descriptions prior to pruning, such as CLIP-L/14 (Radford et al., 2021) or Sentence-T5-XL (Reimers and Gurevych, 2019) in Table 3b. Surprisingly, CLIP outperforms Sentence-T5 that is trained with a sentence-level objective (which is expected to better align with our caption-similarity computation). Finally, we evaluate different classifiers used for close-ended (*i.e.*, multiple-choice question) VQA setups (see Table 3c). Despite commonly-used in LLM literature, generative classifier performs worse than log-likelihood classifier. Such performance is also intuitive as the latter constrains predictions within the given answer choices (hence, less hallucination). More discussion on this is in the supplementary (see Sec. A.2).

**Repository setup and metadata:** In the formulation of LangRepo we ablate different hyperparameter settings related to the number of repo-updates (#iterations), the number of video chunks in each iteration (#chunks), and multiple temporal-scales considered when reading data in repository. In Table 3d, we make two observations: (1) more update iterations with finer chunks (higher #chunks per iteration) can preserve more-useful information, and (2) reading information in multiple temporal-scales is consistently better. Moreover, we consider optional metadata to help preserve information that may get lost when pruning (*e.g.* temporal ordering, or repetitive captions), namely, timestamps and #occurrences (*i.e.*, the number of captions grouped within each repo description). We see in Table 3e that #occurrences help weigh each description when summarizing, resulting in better performance. However, timestamps do not provide meaningful improvement in our setup, in the context of EgoSchema VQA.

**Efficiency in a multi-query setup:** We also ablate the efficiency of our concise representation

| LLM | Scale | Acc. |
|---|---|---|
| Mistral (Jiang et al.) | 7B | 50.8 |
| Llama2 (Touvron et al.) | 13B | 43.0 |
| Llama3.1 (Dubey et al.) | 70B | 62.2 |

(a) **Choice of LLM**: In the LLoVi framework, Mistral outperforms LLama2 even at a smaller scale.

| Text encoder | Acc. |
|---|---|
| Sentence-T5-XL (Reimers and Gurevych) | 56.4 |
| CLIP-L/14 (Radford et al.) | 57.8 |

(b) **Text encoder**: CLIP outperforms Sentence-T5 (trained with setntence objective) for similarity-based pruning.

| VQA classifier | Acc. |
|---|---|
| Generative | 57.8 |
| Log-likelihood | 60.8 |

(c) **VQA classifier**: Log-likelihood classifier performs better on close-ended VQA.

| #Iter | #Ch | Read | Acc. |
|---|---|---|---|
| 1 | [2] | 1 | 57.0 |
| 1 | [4] | 1 | 60.8 |
| 3 | [4,3,2] | 1 | 58.4 |
| 3 | [4,3,2] | 2 | 59.4 |
| 3 | [4,3,2] | 3 | 61.2 |

(d) **Repository setup**: Having more iterations (#Iter) with finer chunks (#Ch) in writing, and multiple scales in reading improve performance.

| Model | Acc. |
|---|---|
| LangRepo (ours) | 60.8 |
| + tstmp | 60.4 |
| + occ | 61.4 |
| + tstmp + occ | 58.2 |

(e) **Metadata in repository**: Timestamps (tstmp) do not help in this setup, yet #occurrences (occ) help by weighing entries.

| Model | Params | Latency per video (s) | | |
|---|---|---|---|---|
| | | q/v = 1 | q/v = 2 | q/v = 5 |
| LLoVi (Zhang et al.) | 7B | 22.11 | 44.34 | 108.75 |
| LangRepo | 7B | 30.98 | 37.46 | 56.90 |
| LLoVi (Zhang et al.) | 12B | 50.06 | 99.84 | 249.95 |
| LangRepo | 12B | 85.09 | 94.90 | 124.33 |

(f) **Efficiency in a multi-query setup**: Despite being initially expensive, re-using our concise representation on videos with multiple queries is more efficient (measured on a single A5000 GPU).

| Captions | Acc. |
|---|---|
| BLIP-2 (Li et al.) | 55.4 |
| LLaVA-1.5 (Liu et al.) | 58.4 |
| LaViLa (Zhao et al.) | 60.8 |
| Oracle | 69.2 |

(g) **Captioner**: Clip-level captions (*e.g.* LaViLa) performs better than frame-level ones. A gap to oracle annotated captions exists.

| Streaming setup | Acc. |
|---|---|
| LLoVi (Zhang et al.) | 50.8 |
| Chunk-based LLoVi | 57.8 |
| LangRepo (ours) | 60.8 |

(h) **Video input**: Feeding short captions chunk-by-chunk to the LLM is empirically-better than feeding all the captions at-once.

| Model | 0.5× | 1× | 2× |
|---|---|---|---|
| Mistral (Jiang et al.) | 49.8 | 48.8 | 46.8 |
| LLoVi (Zhang et al.) | 57.2 | 55.4 | 53.6 |
| LangRepo | 56.4 | 57.8 | 56.4 |

(i) **Input length**: Both Mistral and LLoVi drops performance with increasing input length, whereas LangRepo retains a more-stable performance.

Table 3: **Ablating design decisions:** We evaluate different design decisions of our framework on EgoSchema (Mangalam et al., 2024) 500-video subset for zero-shot video VQA.

in Table 3f. LangRepo can be initially expensive, as it requires multiple write-read operations (yet, each processing smaller context-lengths). However, once repository is created, it can be re-used more-efficiently in a setup with multiple-queries for a given video (*i.e.*, the initial cost will be amortized). This is especially relevant in practical scenarios, where users may have multiple queries per video (*e.g.* 8.76 q/v in NExT-QA (Xiao et al., 2021) and 3.76 q/v in IntentQA (Li et al., 2023a)).

**Captioner quality:** In Table 3g, we evaluate the quality of captions consumed by LangRepo. By default, we use short-clip captions from LaViLa (Zhao et al., 2023c), which outperform frame-level captions (BLIP-2 (Li et al., 2023b), LLaVA-1.5 (Liu et al., 2023)). Oracle captions from Ego4D show the performance upper-bound w.r.t. the input (*i.e.*, quality of captions).

**Input format and length:** We consider different ways of consuming long video data, either as a whole or as chunks (see Table 3h). Among these options, processing as chunks enables preserving more fine-grained details in LLM outputs. Our repository setup provides further improvement showing its effectiveness over the baseline with the same chunk-based processing. Finally, we re-visit the experiment on how the input length affects the effectiveness of LLMs, presented in Table 1. In Table 3i, we show that LangRepo provide more-stable performance with increasing input lengths, in contrast to baselines.

## 6 Conclusion

In this paper, we introduced a Language Repository (LangRepo), which reads and writes textual information corresponding to video chunks, as a concise, multi-scale and interpretable language representation, together with additional metadata. Both our `write-to-repo` and `read-from-repo` operations are text-based and implemented as calls to a backbone LLM. Our empirical results show the superior performance of LangRepo on multiple long-video reasoning benchmarks at its respective scale, while also being (1) less-prone to performance drops due to increasing input lengths, and (2) interpretable, enabling easier human intervention as needed.

## Acknowledgements

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grants funded by the Korean government [24ZR1100, A Study of Hyper-Connected Thinking Internet Technology by autonomous connecting, controlling and evolving ways]. This work was also supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [No. RS-2024-00336738, Development of Complex Task Planning Technologies for Autonomous Agents, 100%]. The authors thank the Robotics Lab at SBU for helpful discussions, and especially Xiang Li, for managing the lab servers and promptly responding to all the infrastructure-related issues.

## Limitations

Despite the effectiveness of LangRepo— as validated by benchmark experiments and thorough ablations, there exists a few limitations of our study which we discuss below.

- First, we note that we are unable to test our approach with all available open-source LLMs due to the rapid pace of development and compute limitations. Yet, we carefully select state-of-the-art model backbones in our main experiments (*e.g.* Mistral (Jiang et al., 2023), Mixtral (Jiang et al., 2024)) and further show generalization to other models/scales in our ablations (*e.g.* LLama2 (Touvron et al., 2023), LLama3.1 (Dubey et al., 2024), GPT-4).
- We also highlight that since our study is focused on a concise and interpretable (*i.e.*, all-textual) representation that applies to two-stage VQA pipelines (*i.e.*, captioner + question-answering LLM), we do not explore single-stage VQA pipelines (*i.e.*, multi-modal LLM such as LLaVA (Liu et al., 2024)) within the scope of this paper.
- Our default redundancy detection technique relies on text embeddings (*e.g.* CLIP (Radford et al., 2021)). Although this decision is validated based on ablations with *(i)* alternate approaches (*e.g.* LLM-based), *(ii)* different encoders (*e.g.* Sentence-T5 (Reimers and Gurevych, 2019)), and *(iii)* hyperparameter search on reduction rate, we note that a very few redundancies may exist in our repository entries.
- As any LLM based approach, LangRepo is sensitive to prompting. We carefully select our prompts, being faithful to prior methods that we compare with (*e.g.* LLoVi (Zhang et al., 2023a)). We also include an extended discussion on such sensitivity in the supplementary, particularly w.r.t. the classifier used in VQA.
- Finally, we note that since our approach is zero-shot, any limitations or biases in pretrained models may still exist in the outputs of LangRepo.

## Reproducibility Statement

Our experiments utilize open-source vision modules, including ResNet and Clippy, with publicly available code and pretrained weights, alongside the proprietary GPT-4o model. Since all experiments are conducted in a zero-shot setting, we do not modify any pretrained weights. Evaluations are performed on publicly available long-video benchmarks, ensuring transparency and comparability. We provide detailed steps and prompts necessary to reproduce our results. Finally, we commit to releasing our code alongside the paper to facilitate further research and reproducibility.

## References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. 2021. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846.

Ivana Balažević, Yuge Shi, Pinelopi Papalampidi, Rahma Chaabouni, Skanda Koppula, and Olivier J Hénaff. 2024. Memory consolidation enables long-context video understanding. *arXiv preprint arXiv:2402.05861*.

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

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2022. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*.

Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference* *on Computer Vision and Pattern Recognition*, pages 6299–6308.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097.

Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Bin Lin, Zhenyu Tang, et al. 2024. Sharegpt4video: Improving video understanding and generation with better captions. *arXiv preprint arXiv:2406.04325*.

Yangyu Chen, Shuhui Wang, Weigang Zhang, and Qingming Huang. 2018. Less is more: Picking informative frames for video captioning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 358–373.

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

Rohan Choudhury, Koichiro Niinuma, Kris M Kitani, and László A Jeni. 2023. Zero-shot video question answering with procedural programs. *arXiv preprint arXiv:2312.00937*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Brendan Duke, Abdalla Ahmed, Christian Wolf, Parham Aarabi, and Graham W Taylor. 2021. Sstvos: Sparse spatiotemporal transformers for video object segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5912–5921.

Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. 2021. Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6824–6835.

Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. 2024. Videoagent: A memory-augmented multimodal agent for video understanding. *arXiv preprint arXiv:2403.11481*.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research*, 23(1):5232–5270.

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

Emilio Ferrara. 2023. Should chatgpt be biased? challenges and risks of bias in large language models. *arXiv preprint arXiv:2304.03738*.

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. 2023. Model tells you what to discard: Adaptive kv cache compression for llms. *arXiv preprint arXiv:2310.01801*.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. 2023. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. 2022. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. 2016. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232.

Tanmay Gupta and Aniruddha Kembhavi. 2023. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14953–14962.

DeLesley Hutchins, Imanol Schlag, Yuhuai Wu, Ethan Dyer, and Behnam Neyshabur. 2022. Block-recurrent transformers. *Advances in Neural Information Processing Systems*, 35:33248–33261.

Md Mohaiminul Islam, Ngan Ho, Xitong Yang, Tushar Nagarajan, Lorenzo Torresani, and Gedas Bertasius. 2024. Video recap: Recursive captioning of hour-long videos. *arXiv preprint arXiv:2402.13250*.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Kumara Kahatapitiya, Anurag Arnab, Arsha Nagrani, and Michael S Ryoo. 2023. Victr: Video-conditioned text representations for activity recognition. *arXiv preprint arXiv:2304.02560*.

Kumara Kahatapitiya and Michael S Ryoo. 2021. Coarse-fine networks for temporal activity detection in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8385–8394.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. *arXiv preprint arXiv:1805.04623*.

Wonkyun Kim, Changin Choi, Wonseok Lee, and Wonjong Rhee. 2024. An image grid can be worth a video: Zero-shot video question answering using a vlm. *arXiv preprint arXiv:2403.18406*.

Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. 2011. Hmdb: a large video database for human motion recognition. In *2011 International conference on computer vision*, pages 2556–2563. IEEE.

Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. 2023. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*.

Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, Daniel Haziza, Luca Wehrstedt, Jeremy Reizenstein, and Grigory Sizov. 2022. xformers: A modular and hackable transformer modelling library. `https://github.com/facebookresearch/xformers`.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. 2024. Same task, more tokens: the impact of input length on the reasoning performance of large language models. *arXiv preprint arXiv:2402.14848*.

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

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

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. 2023c. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. 2024a. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206.

Sheng Li, Fengxiang He, Bo Du, Lefei Zhang, Yonghao Xu, and Dacheng Tao. 2019. Fast spatio-temporal residual network for video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10522–10531.

Xiang Li, Cristina Mata, Jongwoo Park, Kumara Kahatapitiya, Yoo Sung Jang, Jinghuan Shang, Kanchana Ranasinghe, Ryan Burgert, Mu Cai, Yong Jae Lee, et al. 2024b. Llara: Supercharging robot learning data for vision-language policy. *arXiv preprint arXiv:2406.20095*.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. 2023. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE.

Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2021. Towards understanding and mitigating social biases in language models. In *International Conference on Machine Learning*, pages 6565–6576. PMLR.

Ruotong Liao, Max Erler, Huiyu Wang, Guangyao Zhai, Gengyuan Zhang, Yunpu Ma, and Volker Tresp. 2024. Videoinsta: Zero-shot long video understanding via informative spatial-temporal reasoning with llms. *arXiv preprint arXiv:2409.20365*.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. 2023. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023. Improved baselines with visual instruction tuning.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. 2022. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3202–3211.

Ziyu Ma, Chenhui Gou, Hengcan Shi, Bin Sun, Shutao Li, Hamid Rezatofighi, and Jianfei Cai. 2024. Drvideo: Document retrieval based long video understanding. *arXiv preprint arXiv:2406.12846*.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. 2024. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36.

Sachit Menon and Carl Vondrick. 2022. Visual classification via description from large language models. *arXiv preprint arXiv:2210.07183*.

Juhong Min, Shyamal Buch, Arsha Nagrani, Minsu Cho, and Cordelia Schmid. 2024. Morevqa: Exploring modular reasoning models for video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13235–13245.

Liliane Momeni, Mathilde Caron, Arsha Nagrani, Andrew Zisserman, and Cordelia Schmid. 2023. Verbs in action: Improving verb understanding in video-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15579–15591.

Jongwoo Park, Kanchana Ranasinghe, Kumara Kahatapitiya, Wonjeong Ryoo, Donghyun Kim, and Michael S Ryoo. 2024. Too many frames, not all useful: Efficient strategies for long-form video qa. *arXiv preprint arXiv:2406.09396*.

AJ Piergiovanni and Michael Ryoo. 2019. Temporal gaussian mixture layer for videos. In *International Conference on Machine learning*, pages 5152–5161. PMLR.

AJ Piergiovanni and Michael S Ryoo. 2018. Learning latent super-events to detect multiple activities in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5304–5313.

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

Kanchana Ranasinghe, Xiang Li, Kumara Kahatapitiya, and Michael S Ryoo. 2024. Understanding long videos in one multimodal language model pass. *arXiv preprint arXiv:2403.16998*.

Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Erix Xing, Ming-Hsuan Yang, and Fahad S Khan. 2023. Glamm: Pixel grounding large multimodal model. *arXiv preprint arXiv:2311.03356*.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Joshua Robinson, Christopher Rytting, and David Wingate. 2023. Leveraging large language models for multiple choice question answering. *ICLR*.

Michael S Ryoo, Keerthana Gopalakrishnan, Kumara Kahatapitiya, Ted Xiao, Kanishka Rao, Austin Stone, Yao Lu, Julian Ibarz, and Anurag Arnab. 2023. Token turing machines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19070–19081.

Michael S Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. 2021. Tokenlearner: What can 8 learned tokens do for images and videos? *arXiv preprint arXiv:2106.11297*.

Chuyi Shang, Amos You, Sanjay Subramanian, Trevor Darrell, and Roei Herzig. 2024. Traveler: A multi-lmm agent framework for video question-answering. *arXiv preprint arXiv:2404.01476*.

Jinghuan Shang, Kumara Kahatapitiya, Xiang Li, and Michael S Ryoo. 2022. Starformer: Transformer with state-action-reward representations for visual reinforcement learning. In *European Conference on Computer Vision*, pages 462–479. Springer.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. 2016. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 510–526. Springer.

Chandan Singh, Jeevana Priya Inala, Michel Galley, Rich Caruana, and Jianfeng Gao. 2024. Rethinking interpretability in the era of large language models. *arXiv preprint arXiv:2402.01761*.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. 2024. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. Vipergpt: Visual inference via python execution for reasoning. *arXiv preprint arXiv:2303.08128*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jiawei Wang, Liping Yuan, and Yuchen Zhang. 2024a. Tarsier: Recipes for training and evaluating large video description models. *arXiv preprint arXiv:2407.00634*.

Shijie Wang, Qi Zhao, Minh Quan Do, Nakul Agarwal, Kwonjoon Lee, and Chen Sun. 2023. Vamos: Versatile action models for video understanding. *arXiv preprint arXiv:2311.13627*.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. 2024b. Videoagent: Long-form video understanding with large language model as agent. *arXiv preprint arXiv:2403.10517*.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803.

Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, et al. 2024c. Internvideo2: Scaling video foundation models for multimodal video understanding. *arXiv preprint arXiv:2403.15377*.

Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. 2022a. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*.

Ying Wang, Yanlai Yang, and Mengye Ren. 2024d. Lifelongmemory: Leveraging llms for answering queries in long-form egocentric videos. *arXiv preprint arXiv:2312.05269*.

Zhenhailong Wang, Manling Li, Ruochen Xu, Luowei Zhou, Jie Lei, Xudong Lin, Shuohang Wang, Ziyi Yang, Chenguang Zhu, Derek Hoiem, et al. 2022b. Language models with image descriptors are strong few-shot video-language learners. *Advances in Neural Information Processing Systems*, 35:8483–8497.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. 2024e. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. *arXiv preprint arXiv:2405.19209*.

Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. 2019. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 284–293.

Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. 2022. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13587–13597.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. 2024. Longvideobench: A benchmark for long-context interleaved video-language understanding. *arXiv preprint arXiv:2407.15754*.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. 2021. 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.

Junbin Xiao, Angela Yao, Yicong Li, and Tat Seng Chua. 2023. Can i trust your answer? visually grounded video question answering. *arXiv preprint arXiv:2309.01327*.

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, et al. 2023. Effective long-context scaling of foundation models. *arXiv preprint arXiv:2309.16039*.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296.

Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. 2024. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*.

Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. 2022. Zero-shot video question answering via frozen bidirectional language models. *Advances in Neural Information Processing Systems*, 35:124–141.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. 2023. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*.

Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. 2024. Self-chained image-language model for video localization and question answering. *Advances in Neural Information Processing Systems*, 36.

Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, et al. 2022. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv preprint arXiv:2204.00598*.

Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. 2023a. A simple llm framework for long-range video question-answering. *arXiv preprint arXiv:2312.17235*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023b. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2023a. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. 2023b. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*.

Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. 2023c. Learning video representations from large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6586–6597.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024a. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024b. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

# A Appendix

## A.1 Extended ablation and discussion

**Similarity-based pruning:** We notice that the short captions generated by the VLLM captioner can be highly-redundant, as it has a limited temporal span. Such excess details can adversely affect the performance (see Table 1), while also wasting the LLM context. This motivates us to prune redundancies. We consider prompting the LLM directly to identify and rephrase redundant information. However, the outputs in this setup can be noisy and lack of any structure that is useful for parsing. In other words, although redundancies get pruned, there is limited controllability and inability of identifying what gets pruned. Hence, we decide to delegate the function of identifying redundancies to a separate module: a similarity-based grouping with the help of text embeddings (*e.g.* CLIP (Radford et al., 2021)). This gives more control on what to prune and how much to prune, while generating outputs that can be parsed to extract other useful metadata (*e.g.* timestamps). In Table A.1c, we show that this CLIP-based approach outperforms LLM-based alternative. We also ablate the reduction rate as a hyperparameter (*i.e.*, percentage of captions detected as redundant), which needs to be balanced to avoid over- or under-detection of redundancy.

**Processing videos as chunks:** Our decision to consume longer videos as chunks is motivated by prior work (Wu et al., 2022; Ryoo et al., 2023). It allows us to not lose short-term details, while also keeping track of long-term dependencies via multi-scale processing. Additionally, although not explored in our scope, such a setup integrates well with temporally-fine-grained prediction tasks, where an LLM needs to make multiple predictions over time.

**Choice of metadata:** To avoid the loss of important details during pruning, we maintain additional metadata in our `LangRepo`. Since captions across time can be grouped together in a single repo description, we save their timestamps as a separate field. This can help with temporal reasoning questions. We also update an occurrence counter, which shows the number of captions grouped within a single description. This can act as a weight, to help in cases such as counting or identifying repetitive events. In Table 3e, we see that the occurrence counter does improve the performance on EgoSchema. We implicitly see the benefit of timestamps in `LangRepo` for time-sensitive question answering (*e.g.* Temporal questions in NExT-QA as in Table A.2, Before/After questions in IntentQA as in Table A.3, and grounded VQA in NExT-GQA as in Table A.4).

**Two-stage VQA pipeline:** `LangRepo` is a two-stage VQA pipeline that relies purely on text information to perform visual question-answering. Such pipelines (Zhang et al., 2023a; Islam et al., 2024; Ranasinghe et al., 2024) are usually comprised of two separate components that focus on either (1) converting visual-information to text, or (2) question-answering. Different from single-stage VQA pipelines, *i.e.*, multi-modal question-answering LLMs (Liu et al., 2024; Wang et al., 2024a,c), that only consist of latent representations (Balažević et al., 2024), two-stage pipelines generate an intermediate language representation with useful properties (*e.g.* interpretability for human observers, a more-natural form of structure for LLM-based processing). Among such, closest to ours are LLoVi (Zhang et al., 2023a) (in-terms of its summarization), Video ReCap (Islam et al., 2024) (in-terms of its multi-scale descriptions), and MVU (Ranasinghe et al., 2024) (in-terms of its multi-modal information as text). Different from these, the novelty of `LangRepo` is on removing redundancies in visual information given as text, across varying scales (based on iterative refinement), improving the effectiveness and context utilization of question-answering LLMs.

**Classifier for close-ended VQA:** The multiple-choice question-answering setup usually considers a generative classifier. Meaning, an LLM is prompted to generate the correct answer option among multiple-choices, directly as next-token prediction. Another approach used in NLP literature is log-likelihood based classification (see Cloze prompting in (Robinson et al., 2023)). Here, the LLM is prompted separately for each of the multiple choices with a template such as "`Question: Answer-option`". The choice that maximizes the log-likelihood of predicted tokens (*i.e.*, tokens corresponding to `Answer-option`) is selected as the correct answer. This is a more-natural setup for close-ended VQA since it avoids hallucination. Among these classifiers, we find the latter to be better-performing, and use it in `LangRepo` as well as other closely-related baselines (*e.g.* LLoVi, Mistral) in our comparisons (see Table A.1b). It even helps smaller backbones to outperform much larger ones (*e.g.* see Table A.1e: `LangRepo` with

| Dataset | Captions per-video | | | | | |
|---|---|---|---|---|---|---|
| | 0.5× | | 1× | | 2× | |
| | #tokens | Acc. | #tokens | Acc. | #tokens | Acc. |
| EgoSchema | ∼0.9k | 49.8 | ∼1.8k | 48.8 | ∼3.6k | 46.8 |
| NExT-QA | ∼0.45k | 48.2 | ∼0.9k | 48.2 | ∼1.8k | 46.9 |
| IntentQA | ∼0.45k | 47.1 | ∼0.9k | 46.9 | ∼1.8k | 45.2 |

(a) **Input length**: We extend Table 1 by including the average #tokens per-video. Despite comfortably fitting into the context length of LLM (*i.e*., Mistral-7B (Jiang et al.)), the performance degrades with longer inputs.

| Model (7B) | VQA classifier | |
|---|---|---|
| | Gen. | LL |
| Mistral | 47.2 | 48.8 (+1.6) |
| LLoVi | 50.2 | 50.8 (+0.6) |
| LangRepo | 58.8 | 60.8 (+2.0) |

(b) **VQA Classifier**: Log-likelihood classifier is consistently better than a generative one across various models.

| Grouping | Red. | Acc. |
|---|---|---|
| CLIP-based | 10% | 55.4 |
| | 25% | 57.8 |
| | 50% | 56.2 |
| LLM-based | 25% | 52.6 |

(c) **Detecting redundancy** with CLIP embeddings is cleaner, yet balancing reduction rate is important.

| Model | Red. removal | Params | Acc. |
|---|---|---|---|
| LifelongMemory | Caption Digest | 8B | 60.4 |
| | | 175B | 64.0 |
| LangRepo | Group/Rephrase | 7B | 60.8 |
| | | 12B | 66.2 |

(d) **Redundancy removal** with Group/Rephrase in LangRepo is better than Caption Digest in LifelongMemory (Wang et al.).

| w/ open-source LLMs | | | | w/ proprietary LLMs | |
|---|---|---|---|---|---|
| Model (7B) | Acc. (Gen.) | Model (70B) | Acc. (LL) | Model (1.8T) | Acc. (Gen.) |
| Mistral | 47.2 | LLama3.1 | - | GPT-4 | 59.0 |
| LLoVi | 50.2 | LLoVi | 62.2 | LLoVi | 61.2 |
| LangRepo | 58.8 | LangRepo | 67.0 | LangRepo | 64.6 |

(e) **Model scales**: Utility of LangRepo is visible not only in relatively-small open-source LLMs (*e.g*. Mistral-7B (Jiang et al.)), but also in large and proprietary LLMs (*e.g*. LLama3.1-70B (Dubey et al.), GPT-4).

Table A.1: **Additional Ablation experiments:** We evaluate different design decisions of our framework on zero-shot video VQA (unless otherwise-stated, on EgoSchema (Mangalam et al., 2024) 500-video subset).

LLama3.1-70B and LL-classifier gives 67.0% on EgoSchema vs. the same with GPT-4 and Gen-classifier at 64.6%). However, we find that the LL-classifier is also more-sensitive to the prompt template. We direct the reader to the next subsection (A.2) for more details.

**Soft degradation of LLM performance:** In this paper, we propose an approach that effectively utilizes the context of question-answering LLMs. It can handle both *(i)* soft performance degradations due to longer inputs that do no exceed context limit, and *(ii)* context truncation— which is an extreme case. We formally show the observations of the more-generic case of *(i)* above w.r.t. the number of captions (in Table 1), and further extend it to include token counts in Table A.1a. We highlight that the Mistral-7B (Jiang et al., 2023) backbone used here, can definitely handle the full dense captions without any context overflow/truncation in all configurations of this experiment. For instance, EgoSchema dense captions at 2× setting contains ∼3.6k tokens— that is reduced to ∼1.4k tokens by LangRepo— which can be handled comfortably within the context length (8k) of this model. This validates that, we will not be triggering a context truncation (even after considering prompt tokens), but rather the observations in this study is due to a soft-trigger of information decay. This behavior is shown to be true even for much-larger LLMs (*e.g.* Gemini) in concurrent work (Levy et al., 2024), which is attributed to the attention mechanism being overwhelmed with increasingly-longer inputs.

## A.2 Sensitivity of prompting for VQA

Given the close-ended answer formulation in our VQA setup, we can consider two different classifiers to make the prediction: (1) a Generative classifier, which directly generates the answer choice, or (2) a Log-likelihood classifier, which select the most-probable choice. Although the latter is less-prone to hallucinations (*i.e*., prediction is explicitly constrained to answer choices), it can also be sensitive to how we prompt— as discussed below.

**Generative classifier:** Here, we directly prompt the LLM to generate the correct answer, conditioned on the descriptions generated by LangRepo, the question and the answer options (inspired by (Zhang et al., 2023a)). To make sure that the output can be parsed, we provide additional guiding instructions and any syntax specific to the LLM (Mistral (Jiang et al., 2023)). This also discourages any hallucinations. On all benchmarks, we use the common prompt given below.

“[INST] «SYS» You are a helpful expert in first person view video analysis. «/SYS» Please provide a single-letter answer (A, B, C, D, E) to the following multiple-choice question, and your answer must be one of the letters (A, B, C, D, or E). You must not provide any other response or explanation. You are given some language descriptions of a first person view video. The video is ${duration} seconds long. Here are the descriptions: ${description}.\n You are going to answer a multiple choice question based on the descriptions, and your answer should be a single letter chosen from the choices.\n Here is the question: ${question}.\n Here are the choices.\n A: ${optionA}\n B: ${optionB}\n C: ${optionC}\n D: ${optionD}\n E: ${optionE}\n [/INST]"

**Log-likelihood classifier:** Inspired by (Robinson et al., 2023; Ranasinghe et al., 2024), in this setup, we prompt the LLM with each answer option separately, and select the highest-probable answer. The probability is computed only on the tokens of the answer option, conditioned on the input sequence. In our experiments, we notice that the effectiveness of this method is sensitive to the prompt. This is due to the question-answer formats in the dataset considered. For instance, EgoSchema (Mangalam et al., 2024) consists of full-sentence answers, whereas NExT-QA (Xiao et al., 2021) consists of answer phrases. Hence, the latter benefits from additional guidance from formatting within the prompt template. More specifically, on EgoSchema (Mangalam et al., 2024), our prompt has the following format.

"${description} ${question} ${answer_option}"

Here, the probability is computed only on `${answer_option}`. However, on the benchmarks based on NExT-QA (Xiao et al., 2021) data, our prompt has a more-structured format as below.

"${description} Based on the description above, answer the following question: ${question}? Select one of these choices as the answer:\n A: ${optionA}\n B: ${optionB}\n C: ${optionC}\n D: ${optionD}\n E: ${optionE}\n The correct answer is, ${option_id}: ${answer_option}"

Here, the probability is computed only on `${option_id}: ${answer_option}`. We observe that neither prompt template works as effective when interchanged.

### A.3 Qualitative examples of repository entries

We present qualitative examples from EgoSchema (Mangalam et al., 2024) dataset to better clarify the operations in LangRepo. In Fig. 4, we show the format of repository entries. Here, non-redundant captions from the input get directly written to the repo. In contrast, any redundant captions—grouped based on similarity— get rephrased as concise descriptions (1 per-group). Each repository description may come with additional metadata such as timestamps and #occurrences to avoid the loss of meaningful information due to pruning. In Fig. A.1, we further elaborate on multiple scales within the repository, which are generated by iteratively processing increasingly-longer chunks (created by `re-chunk` operation). During reading, we can decide to summarize information at various temporal scales to generate output descriptions useful for VQA.

### A.4 Additional benchmark results

**Detailed results on NExT-QA and IntentQA:** In Table A.2, we extend the benchmark evaluation on NExT-QA (Xiao et al., 2021) to include its validation splits (Causal, Temporal and Descriptive) to provide a better semantic understanding of model performance. Similarly, in Table A.3, we extend the benchmark evaluation on IntentQA (Li et al., 2023a) to include its test splits splits (Why?, How? and Before/After). In both benchmarks, we observe that LangRepo outperforms the competition on respective splits, showing the generalization of our language representation for various semantic reasoning tasks.

**Grounded VQA:** We consider NExT-GQA (Xiao et al., 2023), a visually-grounded VQA dataset with 10.5K temporal grounding annotations, where we perform zero-shot inference similar to (Zhang et al., 2023a) on its test split. We report multiple metrics including Intersection-over-Prediction (IoP) that measures the overlap w.r.t. the predicted window, Intersection-over-Union (IoU) that measures the overlap w.r.t. the union of ground-truth and predicted windows, and Acc@GQA that measures the accuracy of correctly-grounded predictions. In Table A.4, we compare the performance of LangRepo with state-of-the-art models on NExT-GQA (Xiao et al., 2023). We follow the same grounding setup as in (Zhang et al., 2023a). Our method achieves a strong performance at its scale, outperforming baseline Mistral LLM (Jiang et al., 2023) by $+2.0\%$ and LLoVi (12B) (Zhang et al., 2023a) by $+0.9\%$ on Acc@GQA metric.

**Very-short video VQA:** We consider MSRVTT-QA (Xu et al., 2016) as a very-short video VQA benchmark for LangRepo evaluation. It is based on MSR-VTT dataset with $\sim$3k clips that range from 10-30s of duration (with an average of 15s), and consists of open-ended questions. Here, we can not rely on log-likelihood classifier to select an

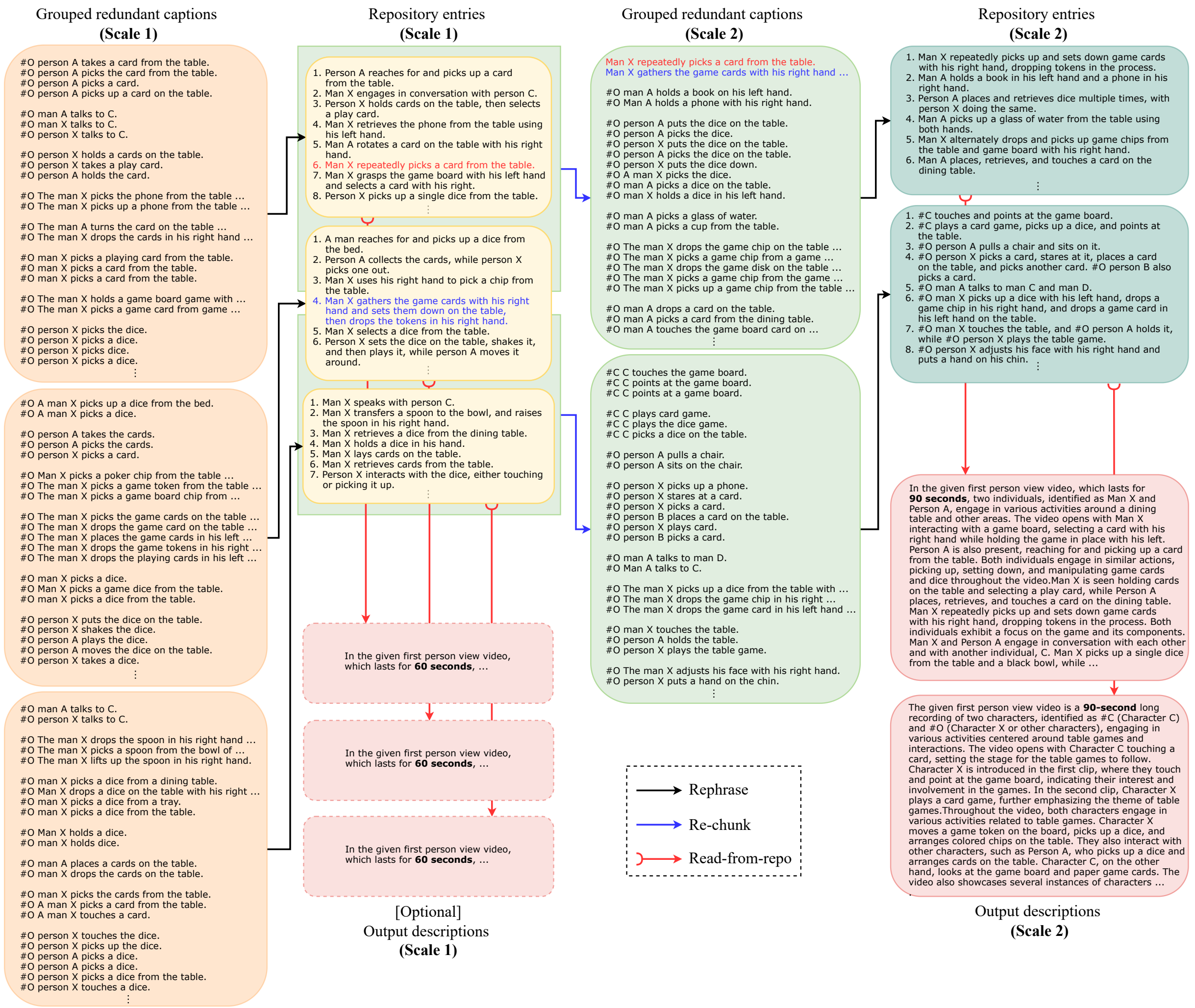



Figure A.1: **A qualitative example of iterative writing and multi-scale reading in LangRepo:** Here, we present an example with 2-scales, given captions of a 180s long video. In scale-1, we consider 3 chunks of 60s each, and in scale-2, we `re-chunk` them into 2 chunks of 90s each. We only show the redundant captions that go through pruning, and also, omit any metadata (*e.g.* timestamps) within the repository. In each scale, captions grouped based on similarity get rephrased concisely. To generate inputs of the subsequent scale, we simply order previous repository descriptions in time, and split (*i.e.*, `re-chunk`) into fewer (and, longer) chunks. When reading, each entry in each scale is summarized separately to create output descriptions of various temporal spans. In general, we always consider the last-scale descriptions to be mandatory, but any prior-scale to be optional. Yet, we observe multiple scales to be beneficial (see Table 3d). Best-viewed with zoom-in.

answer, but rater generate an open-ended answer and compare it with the ground-truth using LLM-as-a-judge (Zheng et al., 2024b), as in prior work. It involves an additional step of querying an LLM to evaluate the correctness of a predicted answer, together with a confidence score (within the range of 1-5). In Table A.5, we compare the performance of LangRepo with similar-sized models (7B). We rely on LLaVA-1.5-7B (Liu et al., 2023) captions (at 4fps) to answer questions in a two-stage VQA pipeline. Here, a two-stage pipeline refers to a *"captioner + question-answering LLM"* setup as in LangRepo, in contrast to a single-stage pipeline (*i.e.*, *multi-modal question-answering LLM*). We see that LangRepo achieves a competitive performance compared to other baselines, while outperforming LLoVi (Zhang et al., 2023a) in the same experimental setup. This validates that LangRepo generalizes to the extreme case of very-short video QA, and our redundancy removal technique is useful even for shorter videos.

**Very-long video VQA:** We consider recent LongVideoBench (Wu et al., 2024) as a very-long video VQA benchmark. It consists of ∼3.7k video clips corresponding to various domains, that range from 8s-1hr of duration, and annotated with multiple-choice questions. In Table A.6, we perform evaluations on its validation set, and report av-

| Model | Params | Causal | Temporal | Descriptive | All |
|---|---|---|---|---|---|
| *with proprietary LLMs* | | | | | |
| ViperGPT (Surís et al., 2023) | 175B | - | - | - | 60.0 |
| ProViQ (Choudhury et al., 2023) | 175B | - | - | - | 64.6 |
| MoReVQA (Min et al., 2024) | 340B | 70.2 | 64.6 | - | 69.2 |
| LVNet (Park et al., 2024) | <1.8T | 75.0 | 65.5 | 81.5 | 72.9 |
| IG-VLM (Kim et al., 2024) | 1.8T | 69.8 | 63.6 | 74.7 | 68.6 |
| LLoVi (Zhang et al., 2023a) | 1.8T | 69.5 | 61.0 | 75.6 | 67.7 |
| TraveLER (Shang et al., 2024) | 1.8T | 70.0 | 60.5 | 78.2 | 68.2 |
| VideoAgent-[S] (Wang et al., 2024b) | 1.8T | 72.7 | 64.5 | 81.1 | 71.3 |
| VideoTree (Wang et al., 2024e) | 1.8T | 75.2 | 67.0 | 81.3 | 73.5 |
| *with open-source LLMs* | | | | | |
| VFC (Momeni et al., 2023) | 164M | 45.4 | 51.6 | 64.1 | 51.5 |
| InternVideo (Wang et al., 2022a) | 478M | 43.4 | 48.0 | 65.1 | 49.1 |
| SeViLA (Yu et al., 2024) | 4B | 61.3 | 61.5 | 75.6 | 63.6 |
| Mistral (Jiang et al., 2023) | 7B | 51.0 | 48.1 | 57.4 | 51.1 |
| LLoVi (Zhang et al., 2023a) | 7B | 55.6 | 47.9 | 63.2 | 54.3 |
| LLoVi (Zhang et al., 2023a) | 12B | 60.2 | 51.2 | 66.0 | 58.2 |
| Tarsier (Wang et al., 2024a) | 34B | - | - | - | 79.2 |
| LangRepo (ours) | 7B | 57.8 | 45.7 | 61.9 | 54.6 |
| LangRepo (ours) | 12B | 64.4 | 51.4 | 69.1 | 60.9 |

Table A.2: **Extended results on NExT-QA (Xiao et al., 2021):** We compare LangRepo against state-of-the-art zero-shot methods on NExT-QA validation set, highlighting standard splits: causal, temporal and descriptive. Our method shows strong performance across all splits at its scale. Open-source multi-modal LLMs with video-caption pretraining are de-emphasized for fair comparison.

| Model | Params | Why? | How? | Before/After | All |
|---|---|---|---|---|---|
| *with proprietary LLMs* | | | | | |
| LVNet (Park et al., 2024) | <1.8T | 75.0 | 74.4 | 62.1 | 71.7 |
| LLoVi (Zhang et al., 2023a) | 1.8T | 68.4 | 67.4 | 51.1 | 64.0 |
| IG-VLM (Kim et al., 2024) | 1.8T | - | - | - | 64.2 |
| VideoTree (Wang et al., 2024e) | 1.8T | - | - | - | 66.9 |
| *with open-source LLMs* | | | | | |
| SeViLA (Yu et al., 2024) | 4B | - | - | - | 60.9 |
| Mistral(Jiang et al., 2023) | 7B | 52.7 | 55.4 | 41.5 | 50.4 |
| LLoVi (Zhang et al., 2023a) | 7B | 57.9 | 55.4 | 42.3 | 53.6 |
| LLoVi (Zhang et al., 2023a) | 12B | 59.7 | 62.7 | 45.1 | 56.6 |
| LangRepo (ours) | 7B | 56.9 | 60.2 | 42.1 | 53.8 |
| LangRepo (ours) | 12B | 62.8 | 62.4 | 47.8 | 59.1 |

Table A.3: **Extended results on IntentQA (Li et al., 2023a):** We compare LangRepo against state-of-the-art zero-shot methods on IntentQA test set, highlighting standard splits: why?, how? and before/after. We focus on the zero-shot setting. Our method shows strong performance across all splits at its scale. Open-source multi-modal LLMs with video-caption pretraining are de-emphasized for fair comparison.

| Model | Params | mIoP | IoP@0.5 | mIoU | IoU@0.5 | Acc@GQA |
|---|---|---|---|---|---|---|
| *with proprietary LLMs* | | | | | | |
| MoReVQA (Min et al., 2024) | 340B | 37.8 | 37.6 | 19.7 | 15.4 | 39.6 |
| LLoVi (Zhang et al., 2023a) | 1.8T | 37.3 | 36.9 | 20.0 | 15.3 | 24.3 |
| *with open-source LLMs* | | | | | | |
| Mistral (Jiang et al., 2023) | 7B | 20.4 | 20.2 | 8.7 | 5.9 | 9.2 |
| LLoVi (Zhang et al., 2023a) | 7B | 20.7 | 20.5 | 8.7 | 6.0 | 11.2 |
| LLoVi (Zhang et al., 2023a) | 12B | 31.4 | 28.8 | 18.4 | 12.0 | 16.2 |
| LangRepo (ours) | 7B | 20.3 | 20.0 | 8.7 | 6.0 | 11.2 |
| LangRepo (ours) | 12B | 31.3 | 28.7 | 18.5 | 12.2 | 17.1 |

Table A.4: **Grounded VQA results on NExT-GQA (Xiao et al., 2023):** We compare LangRepo against state-of-the-art zero-shot methods on NExT-GQA test set. Our method shows strong performance across at its scale.

| Model (7B) | Acc. | Confidence |
|---|---|---|
| *single-stage* | | |
| MovieChat (Song et al., 2024) | 52.7 | 2.6 |
| VideoChat2 (Li et al., 2024a) | 54.1 | 3.3 |
| IG-VLM (Kim et al., 2024) | 63.7 | 3.5 |
| *two-stage* | | |
| VideoChat (Li et al., 2023c) | 45.0 | 2.5 |
| LLoVi (Zhang et al., 2023a) | 58.6 | 2.9 |
| LangRepo (ours) | 59.2 | 3.0 |

Table A.5: **Very-short video VQA results:** We compare LangRepo against state-of-the-art zero-shot methods at the same scale (7B), on open-ended question answering on MSRVTT-QA (Xu et al., 2016) (using LLM-as-a-judge). Our method shows competitive performance at its scale. Open-source multi-modal LLMs with video-caption pretraining are de-emphasized for fair comparison.

| Model | Params | Acc. |
|---|---|---|
| *single-stage* | | |
| VideoChat2 (Li et al., 2024a) | 7B | 36.0 |
| mPLUG-Owl2 (Ye et al., 2023) | 7B | 39.1 |
| VideoLLaVA (Lin et al., 2023) | 7B | 39.1 |
| ShareGPT4Video (Chen et al., 2024) | 8B | 39.7 |
| PLLaVA (Xu et al., 2024) | 7B | 40.2 |
| *two-stage* | | |
| Mistral (Jiang et al., 2023) | 7B | 37.4 |
| LangRepo (ours) | 7B | 38.2 |

Table A.6: **Very-long video VQA results:** We compare LangRepo against state-of-the-art zero-shot methods at a similar scale, on LongVideoBench (Wu et al., 2024) validation set. Our method shows a competitive performance at its scale. Open-source multi-modal LLMs with video-caption pretraining are de-emphasized for fair comparison.

erage performance across all duration splits. Here, we run our framework on captions extracted with LLaVA-1.5-7B (Liu et al., 2023) in a two-stage pipeline, evaluating its performance against similar-scaled models. We observe that LangRepo shows a competitive performance, while outperforming Mistral (Jiang et al., 2023) based on the same captioner. Even in a benchmark that better-supports single-stage VQA pipelines based on multi-modal LLMs, such a performance from LangRepo validates its effectiveness in the extreme case of very-long video QA.