World Model on Million-Length Video And Language With Blockwise RingAttention

Anonymous authors
Paper under double-blind review

Abstract

Enabling long-context understanding remains a key challenge in scaling existing sequence models – a crucial component in developing generally intelligent models that can process and operate over long temporal horizons that potentially consist of millions of tokens. In this paper, we aim to address these challenges by providing a comprehensive exploration of the full development process for producing 1M context language models and video-language models, setting new benchmarks in language retrieval and new capabilities in long video understanding. We detail our long context data curation process, progressive context extension from 4K to 1M tokens, and present an efficient open-source implementation for scalable training on long sequences. Additionally, we open-source a family of 7B parameter models capable of processing long text documents and videos exceeding 1M tokens.

1 Introduction

Enabling long-context understanding remains a key challenge in scaling existing sequence models—a crucial step toward developing generally intelligent models that can process and operate over extended temporal horizons, potentially involving millions of tokens. Current modeling approaches are predominantly limited to processing short sequences, whether in the form of language, images, or video clips (Brown et al., 2020; Touvron et al., 2023a;b; OpenAI, 2023; Brooks et al., 2024; Team et al., 2023). As a result, these models fall short when tasked with understanding complex, long-form language and visual contexts.

However, training models to process sequences that exceed millions of tokens is a significant challenge due to the high memory and computational costs, as well as the lack of long-context data. In this work, we address these challenges by leveraging Blockwise RingAttention (Liu et al., 2024; Liu and Abbeel, 2023), a technique that scales context size without approximations or overheads, enabling efficient training on long sequences. We curate an extensive dataset of long-form videos and books from public sources, covering a wide variety of activities and narrative structures. To address the scarcity of long-form conversational datasets, we developed a model-based question-answering technique, where a short-context model generates training data from books, significantly enhancing the model's chat

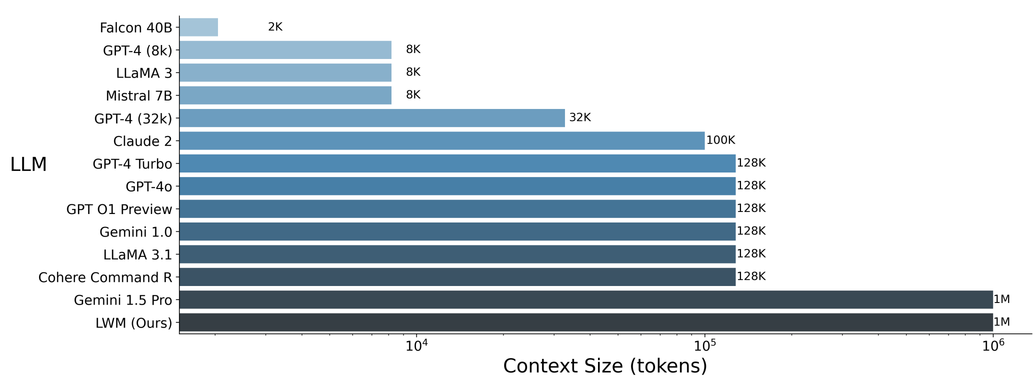


Figure 1 Comparison of context size in state-of-the-art LLMs. Our model and concurrent work Gemini 1.5 both achieve a 1M context size, significantly outperforming other LLMs.

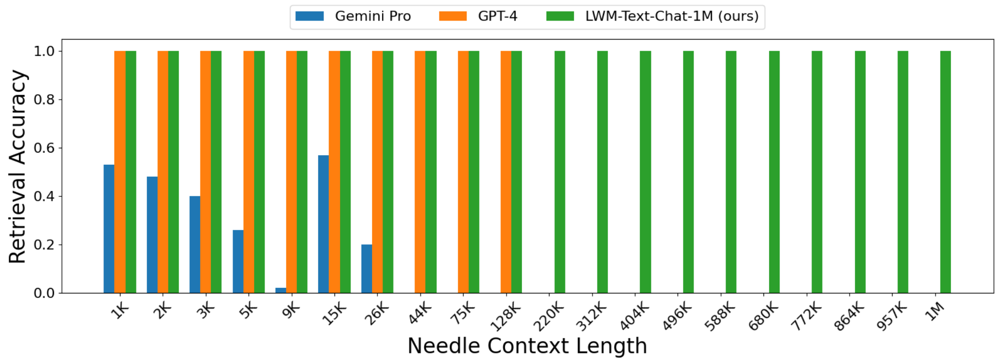


Figure 2 Retrieval comparisons against Gemini Pro and GPT-4. Needle retrieval comparisons against Gemini Pro and GPT-4 for each respective max context length – 32K and 128K. Our model performs competitively while being able to extend to 8x longer context length. Note that in order to show fine-grained results, the x-axis is log-scale from 0-128K, and linear-scale from 128K-1M.

capabilities over long sequences. To mitigate computational costs, we gradually extended context size from an initial 4K tokens to 1M tokens, achieving a cost-effective and scalable approach for long-context modeling.

Following this, we further train our long-context language model to incorporate visual modalities, such as image and video. Contrary to existing popular vision-language models (Liu et al., 2023a; OpenAI, 2023; Chen et al., 2023a), we opt to additionally optimize next-token prediction losses for image and video (generation) with a VQGAN (Esser et al., 2021) encoder. We encountered various challenges training on mixed modalities (video, image, text). To balance their unique characteristics - sequential information, visual detail, and linguistic content - we implement an efficient masked sequence packing strategy, as well as introduce careful loss balancing to retain short context accuracy. This approach handles varying sequence lengths more effectively than standard methods. We also optimized the ratio of image, video, and text inputs in each batch, proposing an empirically effective balance for cross-modality learning. Since our model aims to model both textual and visual projections of the world through a large context window, drawing inspiration from prior work on world models (Brooks et al., 2024; Ha and Schmidhuber, 2018), we name our work as Large World Model (LWM).

Our contributions are threefold: (a) we train one of the largest context size transformers to date on long text documents and videos and achieved competitive results on long video understanding and long context fact retrieval. (b) We discover a range of challenges associated with training on long sequences and propose solutions for them: masked sequence packing to effectively train with different sequence lengths and synthetic model-generating question-answering for effective attention. (c) We provide an open-source and optimized implementation for training with millions of tokens in context, as well as a family of Llama-based 1M context models capable of processing long documents (`LWM-Text`, `LWM-Text-Chat`) and videos (`LWM`, `LWM-Chat`) of 1M tokens.

2 Method overview

We train a large autoregressive transformer model with a large context window of up to one million tokens, building upon Llama2 7B (Touvron et al., 2023b). To achieve this goal, we implement a two-stage training strategy. In Stage I (Section 3), we extend the context to 1M tokens using book-length texts. This is followed by Stage II (Section 4), where we conduct joint training on diverse long multimodal sequences, incorporating text-image data, text-video data, and book-length texts. Our model architecture is the standard autoregressive transformer design, as illustrated in Figure 3. For a comprehensive overview of our training stages and the datasets employed, please refer to Figure 4.

3 Stage I: Learning Long-Context Language Models

This stage aims at first developing `LWM-Text` and `LWM-Text-Chat`, a set of long-context language models learned by training on progressively increasing sequence length data, and modifying positional

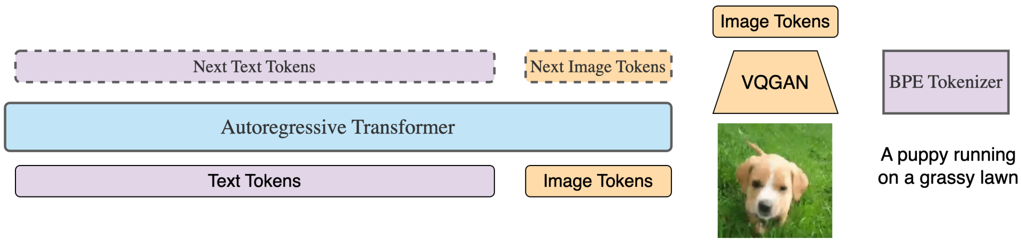


Figure 3 Model Architecture. The LWM model is an autoregressive transformer trained on sequences of multimodal tokens. Each video frame is tokenized into 256 tokens using VQGAN, while text is processed using a Byte-Pair Encoding (BPE) tokenizer. These tokens—both image and text—are combined and input into the transformer to autoregressively predict the next token. The model can handle various input-output modalities, including text, image, video, and text-video pairs. To distinguish between images and text, special tokens <vision> and </vision> are used for image and video frames, with <eof> and <eov> marking the end of these sequences. For simplicity, delimiters are not shown in the figure.

encoding parameters to account for longer sequence lengths (see Section 3.1). In Section 3.2, we show how to construct model-generated question-answering data for enabling long sequence conversations.

3.1 Progressive Training Towards Long Context

Learning long-range dependencies over sequences of millions of tokens requires (1) memory efficient training to scale to such long sequences, as well as a need to (2) compute efficient training to extend the context of our base language model. We outline our approach to these challenges, detailing our methods for training on long sequences, designs for efficiency and stability, and experimental setup.

Training on long sequences has become prohibitively expensive due to memory constraints imposed by the quadratic complexity of attention weight computations. To address these computational limitations, we leverage recent advancements in scaling context window size, particularly Blockwise RingAttention (Liu et al., 2024). This approach theoretically allows for an infinite context, bounded only by available devices. We further enhance performance by fusing it with FlashAttention (Dao et al., 2022) using Pallas (Bradbury et al., 2018) to optimize performance compared with using XLA compiler. Notably, with enough tokens per device—already a given—the communication cost during sequence parallelism is fully overlapped by computation, resulting in no additional overhead.

For better efficiency, we adopt a training approach inspired by prior research on extending context (Jin et al., 2023a), where our model is trained on progressively longer sequence lengths, starting from 32K tokens and ending at 1M tokens in increasing powers of two. Intuitively, this allows the model to save compute by first learning shorter-range dependencies before moving onto longer sequences. For extending positional embeddings to longer contexts, we adopt a simple, scaled-up version of the approach explored in Rozière et al. (2023), where the θ parameter for RoPE (Su et al., 2024) is scaled in proportion to the context length. We found this approach to be stable for extending positional embeddings with larger context lengths due to its simplicity, requiring the tuning of only a single hyperparameter. Specifically, we scale the θ parameter for RoPE alongside increases in context window sizes – the values are shown in Table 6. The progressive training of growing context sizes is shown in Figure 4.

We initialize from LLaMA-2 7B (Touvron et al., 2023b) as base language model and progressively increase the effective context length of the model across 5 stages: 32K, 128K, 256K, 512K, and 1M. For each stage, we train on different filtered versions of the Books3 dataset from The Pile (Gao et al., 2020). Table 6 details information about each training stage, such as the number of tokens, total time, and the Books3 dataset filtering constraints. Each successive run is initialized from the prior sequence length.

3.2 Model-Generated Question-Answering For Effective Context

We construct a simple question-answering dataset to develop long-context chat capabilities. First, we split documents from the Books3 dataset into fixed chunks of 1,000 tokens, feed each chunk into our short-context language model, and prompt it to generate a question-answer pair based on the

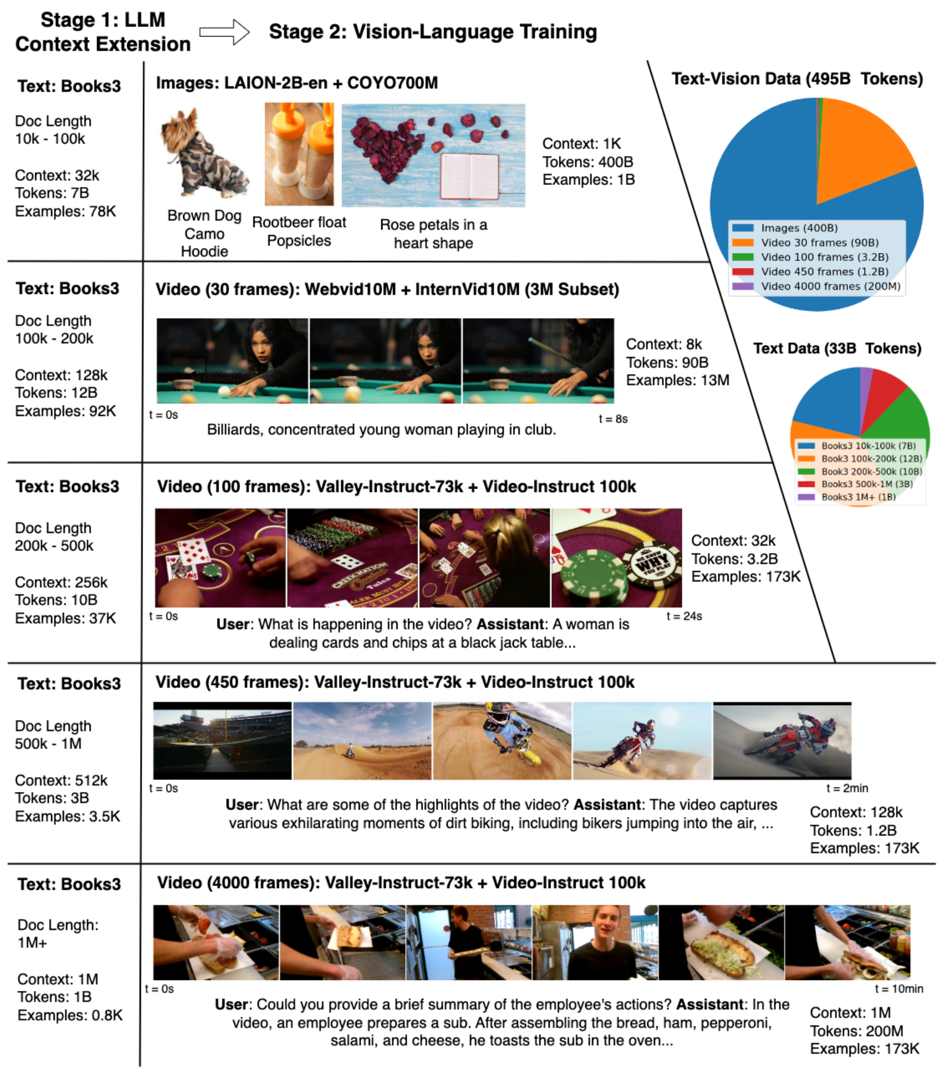


Figure 4 Curated dataset and training process with progressively increasing data length and complexity. The diagram outlines a two-stage training process. Stage 1 extends text-based understanding using books datasets of increasing document lengths and token counts. Stage 2 integrates vision-language training. Pie charts display token distribution, showing that images and short-frame videos dominate visual data, while mid-length text examples lead in the text corpus.

content. To create longer examples (e.g., 32K tokens), we concatenate adjacent chunks and append the relevant question-answer pairs toward the end of the sequence in a chat format. The key intuition is that the model must learn to focus on any part of the context to answer the questions, as the relevant information can appear anywhere within the sequence.

For chat fine-tuning, we train each model on a mix of the UltraChat conversation dataset (Ding et al., 2023) and our custom question-answering dataset, using approximately a 7:3 ratio. We found it crucial to pre-pack the UltraChat data to the training sequence length and keep these examples separate from our question-answering data. This separation is necessary because UltraChat data generally contains a much higher proportion of loss tokens (due to densely packed, short questions in chat), whereas our question-answering data has long questions in chat thus a significantly lower percentage of loss tokens per sequence ($< 1\%$). This difference arises from the long documents in the given context of our question-answering data, which are not included in loss calculations. Table 7 provides further training details for each run. Notably, we do not employ progressive training for any of the chat models; instead, we initialize them from their respective pretrained models at the same context length.

Summary: Stage I progressively increase sequence lengths using our curated dataset: starting with 32K tokens and gradually scaling up to 1M tokens. Model-generated question-answering data aids in learning effective long context.

3.3 Language Evaluation Results

3.3.1 Short Context Tasks

Table 1 presents a comparative analysis between the Llama2-7B model with a 4K context and its context-expanded counterparts, ranging from 32K to 1M. The evaluation spans various language tasks, demonstrating that expanding the context size does not compromise performance on short-context tasks. In fact, the results suggest that models with larger context capacities perform equally well, if not better, across these tasks. This evidence indicates the absence of negative effects from context expansion, highlighting the models' capability to adapt to different task requirements without losing efficiency in shorter contexts.

Table 1 Performance evaluation across language tasks, comparing Llama-2 7B (4K context window) and context-expanded variants of LWM-Text (32K to 1M). The results demonstrate that increasing context length does not significantly degrade performance on tasks with shorter contexts.

		LWM-Text				
Task / Metric	**Llama-2 7B**	**32k**	**128k**	**256k**	**512k**	**1M**
arc_challenge/acc	0.40	0.43	0.45	0.44	0.44	0.43
arc_challenge/acc_norm	0.43	0.47	0.47	0.46	0.46	0.46
hellaswag/acc	0.57	0.57	0.57	0.57	0.56	0.57
hellaswag/acc_norm	0.77	0.76	0.76	0.76	0.75	0.75
mmlu	0.39	0.4	0.41	0.41	0.36	0.35
openbookqa/acc	0.32	0.33	0.31	0.32	0.33	0.30
openbookqa/acc_norm	0.44	0.44	0.44	0.43	0.41	0.41

3.3.2 Retrieval Task: Single Information

We evaluate on the popular Needle In A Haystack task (gkamradt, 2023) – more specifically an version (ArizeAI, 2023) that finds and retrieves random numbers assigned to randomized cities from the context. Figure 2 shows that we can scale to far larger contexts compared to the current best available LLMs. Figure 11 in Appendix shows nearly perfect retrieval accuracy over the entire context of our 1M context model. Appendix C shows more single needle retrieval results for our other shorter context length models.

3.3.3 Retrieval Task: Multiple Information

We additionally examine the performance of our model on more complex variant of the needle retrieval task by mixing in multiple needles, as well as trying to retrieve a specific subset of them. Figure 5 shows multi-needle retrieval results under different settings. Our model generalizes well when retrieving a single needle from multiple needles in context, with slight degradation when asked to retrieve more than one needle. Table 2 shows multi-needle comparisons, where our model is able to perform competitively or better than GPT-4 at retrieving one needle, or slightly lower performance when retrieving more than one needle. Furthermore, our model is also able to perform well and extend to longer context lengths of up to 1M tokens and far outperforms any recent shorter context baselines applies to longer sequence lengths through positional extrapolation techniques.. However, we note that we see degradation in accuracy while increasing the difficulty of the needle retrieval task, suggesting that there is still more room to improve on the 1M context utilization of our model. We believe that our released model will provide a foundation for future work on developing longer context models, as well as encourage more challenging benchmarks that contain difficult long-range tasks that require higher levels of synthesis, rather than pure fact retrieval.

Table 2 Multi Needle in a Haystack. * denotes models **after** the completion of this paper.

Context Length	**Model**	$N=2, R=2$	$N=4, R=1$	$N=4, R=2$
32K	Gemini Pro (02/23)	0.34	0.44	0.6
	GPT-4-1106	0.97	0.95	0.9
	Llama-3.1-8B-Instruct*	0.87	0.95	0.93
	Qwen2.5-7B-Instruct*	**1.0**	**1.0**	**0.97**
	Mistral-7B-Instruct-v0.3*	0.98	0.85	0.83
	LWM-Text-1M (Ours)	0.84	0.97	0.84
128K	Gemini Pro (02/23)	-	-	-
	GPT-4-1106	0.92	0.8	0.82
	Llama-3.1-8B-Instruct*	**0.98**	0.91	0.87
	Qwen2.5-7B-Instruct*	**0.98**	0.80	**0.90**
	Mistral-7B-Instruct-v0.3*	0.85	0.75	0.68
	LWM-Text-1M (Ours)	0.83	**0.98**	0.83
1M	Gemini Pro (02/23)	-	-	-
	GPT-4-1106	-	-	-
	Llama-3.1-8B-Instruct*	0.27	0.32	0.18
	Qwen2.5-7B-Instruct*	0.0	0.0	0.0
	Mistral-7B-Instruct-v0.3*	0.05	0.13	0.10
	LWM-Text-1M (Ours)	**0.67**	**0.84**	**0.69**

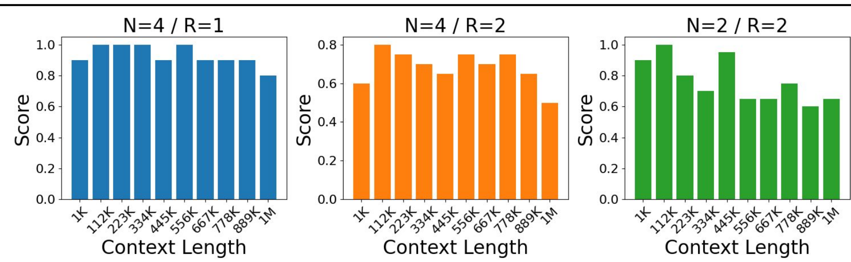


Figure 5 Multiple needles retrieval task with LWM-1M. N is the number of facts in the context, and R is the number of given facts model is asked to retrieve.

3.3.4 Evaluation on LOFT

Table 3 Evaluations on some benchmarks in the LOFT dataset.

Setting: 512K Context	LWM (512K)	GPT-4o (128K)	Claude 3 Opus (200K)
Quora	**0.38**	0.23	0.37
NQ	**0.37**	0.22	0.37
HotPotQA	**0.72**	0.21	0.32

We further evaluate our model on a coverage of the LOFT (Lee et al., 2024) dataset collection, we provides a more natural set of benchmarks that examine capabilities for long-context models in the context of document retrieval, and RAG. The benchmark includes tasks such as duplication detection (Quora [1]), document retrieval (HotpotQA (Yang et al., 2018)), and retrieval-based question-answering (NQ). Each dataset contains a corpus of 1000s of documents, and the model is asked to retrieve a set of document ids pertaining to its specific task (Quora, HotpotQA). For RAG (NQ dataset), the model is asked to answer the question using the given context. Table 3 shows evaluations results on 512K context length against various language model baselines.

Takeaway: Long context capability enables LWM to outperform state-of-the-art text models at multiple benchmarks. This demonstrates the effectiveness of our methods for enabling long context.

4 Stage II: Extending to Long-Context Vision-Language

Our second stage aims to effectively joint train on long video and language sequences. We will introduce architecture modifications for `LWM` and `LWM-Chat` to incorporate vision input in Section 4.1.

[1] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

Training on varying sequence lengths is discussed in Section 4.2. The evaluation results are shown in Section 4.3. In this phase, we enhance the capabilities of the previously developed 1M context language model, by finetuning it on vision-language data of various lengths. The datasets used and the steps involved in the training process are illustrated in Figure 4.

4.1 Architectural Modifications For Vision

We use the pretrained VQGAN (Esser et al., 2021) from aMUSEd (Patil et al., 2024) that tokenizes 256×256 input images to 16×16 discrete tokens. Videos are tokenized by applying the VQGAN per-frame, and concatenating the codes together. In order to distinguish between modalities when generating, as well as knowing when to switch, we introduce mechanisms to mark the end of text generation / beginning of vision generation, and vice-versa. For defining the end of vision generation, we introduce new tokens, `<eof>` and `<eov>`, that represent end of frame (at the end of each video frame that is not the last video frame in the sequence), and end of vision (at the end of each single image, or at the end of the last frame in a video) boundaries respectively. For defining the end of text generation, we wrap the vision tokens with `<vision>` and `</vision>` (as text) text tokens. The model is trained with interleaved concatenations of vision and text tokens, and predicted autoregressively (see Figure 3).

4.2 Training Steps

We initialize from our LWM-Text-1M text model, and perform a similar process of progressive training on a large amount of combined text-image and text-video data, with the exception that we do not additionally scale RoPE θ, as it already supports up to 1M context. Table 8 shows details for each training stage, where the model is initialized from the prior shorter sequence length stage. For each stage, we train on the following data:

- `LWM-1K`: We train on large set of text-image dataset comprising of a mix of LAION-2B-en (Schuhmann et al., 2022) and COYO-700M (Byeon et al., 2022). The datasets were filtered to only include images with at least 256 resolution – in total roughly 1B text-image pairs. During training, we concatenate the text-image pairs and randomly swap the order of the modalities to model both text-image generation, unconditional image generation, and image captioning. We pack text-image pairs to sequences of 1K tokens.
- `LWM-8K`: We train on a text-video dataset mix of WebVid10M (Bain et al., 2021) and 3M InternVid10M (Wang et al., 2023) examples. Similar to prior works (Ho et al., 2022a;b; Villegas et al., 2022), we jointly train on both images and video with a 50-50 ratio of each modality. We pack images to sequences of 8K tokens, and 30 frame videos at 4FPS. Similar to image training, we randomly swap the order of modalities for each text-video pair.
- `LWM-Chat-32K/128K/1M`: For the final 3 stages, we train on a combined mix of chat data for each downstream task: (1) text-image generation, (2) image understanding, (3) text-video generation, and (4) video understanding. We construct a simple version of text-image and text-video chat data by sampling random subsets of the pretraining data augmented with chat format. For image understanding, we using the image chat instruct data from ShareGPT4V (Chen et al., 2023a). Lastly, for the video understanding chat data, we use a combined mix of Valley-Instruct-73K (Luo et al., 2023) and Video-ChatGPT-100K instruct data (Maaz et al., 2023). For all short context data (image generation, image understanding, video generation), we pack sequences to the training context length. During packing, we found it crucial to mask out the attention so that each text-vision pair only attends to itself, as well as re-weighting losses to make computation identical to training in a non-packed + padding training regime. For video understanding data, we uniformly sample a max number of frames to fit the training context length of the model if the video is too long. During training, We allocate 25% of each batch to each of the 4 downstream tasks.

For the first two stages of training (LWM-1K and LWM-8K), we additionally mix 16% of the batch to be pure text data from OpenLLaMA (Geng and Liu, 2023), as we found it beneficial to preserve language capabilities while training on vision data.

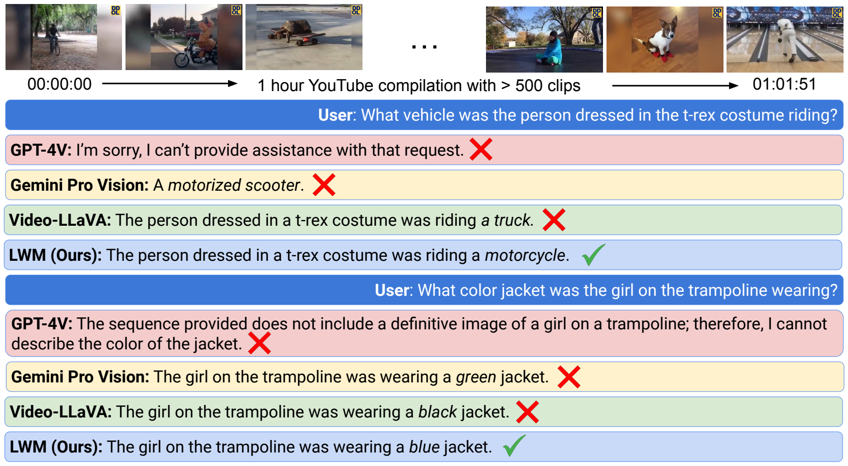


Figure 6 **LWM excels in answering questions about a 1-hour YouTube video**. This figure compares LWM-Chat-1M with proprietary models like Gemini Pro Vision and GPT-4V, along with open-source models. The test involves answering questions based on an hour-long YouTube compilation containing over 500 video clips. LWM demonstrates superior performance in providing accurate answers requiring comprehension of extended video content.

Table 4 Long Video-MME Benchmark. * denotes models **after** the completion of this paper.

Method	Parameters	Frames	Medium (4min-15min)	Long (30min-60min)
Gemini 1.5 Pro*	Unknown	$\leq$ 1800	74.3	67.4
GPT-4o*	Unknown	384	70.3	65.3
LLaVA-Video*	72B	64	68.9	61.5
VideoLLaMA 2*	72B	32	59.9	57.6
Long-LLaVA*	7B	64	51.4	45.4
Video-LLaVA	7B	8	38.1	36.2
LWM-1M	7B	$\leq$ 1800	63.7	60.8

Summary: Stage II training incorporates image and video. Building on Stage I, it gradually increases sequence lengths of vision and text input. Importantly, we found our masked sequence packing and mixing synthetic and chat data crucial to retain short context performance during our progressive training. Appendix B shows ablations when not using our training method on instruction-following and text-image understanding benchmarks.

4.3 Vision-Language Evaluation Results

4.3.1 Long Video Understanding

Although vision-language model (Lin et al., 2023; OpenAI, 2023; Team et al., 2023) can ingest long videos, this is commonly done by performing large temporal subsampling of video frames due to limited context length. For example, Video-LLaVA (Lin et al., 2023) is restricted to uniformly sampling 8 frames from a video, no matter how long the original video may be. As such, models may lose more fine-grained temporal information that is important for accurately answering any questions about the video. In contrast, our model is trained on long sequences of 1M tokens, and as a result, can simultaneously attend thousands of frames of videos to retrieve fine-grained information over short time intervals. Table 4 shows long video evaluations on the Video-MME (Fu et al., 2024) benchmark, demonstrating our model as the best performing model among its size class. Figure 6 shows an example of our model correctly answering questions about a long, 1-hour YouTube compilation consisting of more than 500 individual clips. Our baseline methods, on the other hand, generally have difficulty answering the questions due to a limited number of frames. More results are shown in Figure 18 and Appendix F.

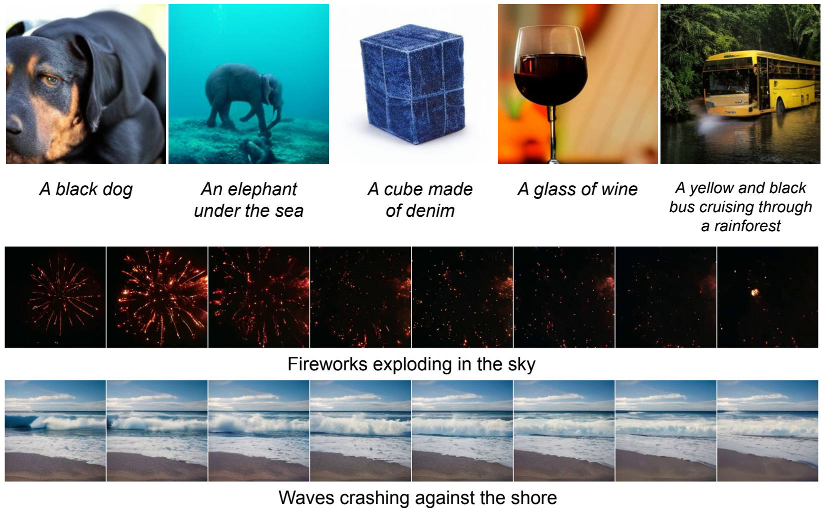


Figure 7 LWM's ability to generate both static images and dynamic videos from text is shown. The top row illustrates image, while the bottom rows show video.

4.3.2 Image Understanding and Short Video Understanding

We evaluate LWM on standard benchmarks for image and short video understanding, with results presented in Table 5. Our model performs comparably to baselines but falls short of state-of-the-art (SOTA) models. This performance gap is not unexpected, given that SOTA models leverage vision backbones that have undergone extensive CLIP training (Radford et al., 2021). In contrast, LWM utilizes discrete tokens from an off-the-shelf model (Patil et al., 2024). Discrete tokens result in greater information loss, particularly for OCR-like textual data, compared to continuous CLIP embeddings. Moreover, our model learns text-image alignment from scratch, while CLIP-based models benefit from large-scale pretraining. This work primarily focuses on long-context methodology, and we defer additional training to future work due to computational constraints. A straightforward approach to improving benchmark scores would be to incorporate CLIP embeddings as additional input. Despite not achieving SOTA scores on these short video benchmarks, we believe LWM provides valuable insights for future long-context language and video understanding and generation. The model's performance could be enhanced through additional training and minor modifications. We include qualitative image understanding examples in Appendix E and qualitative video understanding examples in Appendix F.

4.3.3 Image and Video Generation

Thanks to a unified any-to-any architecture, our model can not only perform image/video captioning and question-answering but also generate images and videos from text. Figure 7 demonstrates examples of these capabilities. For autoregressive sampling, we employ classifier-free guidance (Ho and Salimans, 2022) on the logits, similar to previous works (Yu et al., 2022; Gafni et al., 2022). In the unconditional branch, we initialize each sequence with `<bos><vision>`. For additional image and video generation examples, please refer to Appendices H and I, respectively.

Takeaway: LWM excels in long video understanding by processing significantly more frames than previous state-of-the-arts, resulting in better understanding. Moreover, its long-context enabled unified any-to-any architecture allows for versatile image and video and text understanding and generation.

Table 5 Image Understanding Benchmarks (left) and Video Understanding Benchmarks (right)

Method	Visual Token	VQAv2	GQA	SQA
MiniGPT-4	CLIP	-	30.8	25.4
Otter	CLIP	-	38.1	27.2
InstructBLIP	CLIP	-	49.2	60.5
LLaVA-1.5	CLIP	78.5	62.0	66.8
LWM (ours)	VQGAN	55.8	44.8	47.7

Method	MSVD	MSRVTT	TGIF
VideoChat	56.3	45	34.4
LLaMA-Adapte	54.9	43.8	-
Video-LLaMA	51.6	29.6	-
Video-ChatGPT	64.9	49.3	51.4
LWM (ours)	55.9	44.1	40.9

5 RELATED WORKS

Our research builds upon existing efforts to extend the context windows of language models, enabling them to process more tokens (Chen et al., 2023b; Tworkowski et al., 2023; Liu et al., 2023c). These approaches often employ innovative extrapolation techniques to expand pretrained positional encodings, followed by model finetuning on longer context data. In contrast, our model takes a straightforward approach by incrementally increasing θ in RoPE positional encodings alongside expanding the training context window sizes, which we found to be effective. Additionally, there have been investigations into architectures that avoid modeling pairwise interactions, such as sparse attention and sliding window techniques (Child et al., 2019; Beltagy et al., 2020). Prior research has explored sequence parallelization (Li et al., 2021; Korthikanti et al., 2022, inter alia), though it is not optimized for blockwise transformers or compatible with memory-efficient attention, both of which are critical for large context training. Our work further leverages large context transformer techniques (Liu et al., 2024; Liu and Abbeel, 2023) to capture exact pairwise interactions in extended sequences for enhanced performance. Load-balancing strategies, such as skipping causal masked computation (Brandon et al., 2023; Li et al., 2023) offer room for further optimization. Concurrent developments like Gemini 1.5 (Reid et al., 2024) reach 1M tokens context size in language and video.

Additionally, our approach relates closely to advances in instruction tuning (Taori et al., 2023; Chiang et al., 2023; Geng et al., 2023), which focus on finetuning models with conversational data to boost their performance across diverse language tasks. We aim to extend these capabilities to the domain of long-sequence understanding in both video and language tasks. To achieve this, we extend the model's context size by training on comprehensive datasets, including books and long videos, and finetune on model-generated question-answering datasets to enhance its ability to handle extended conversational sequences.

Furthermore, our research draws from work on integrating vision capabilities into language models (Liu et al., 2023b; Lin et al., 2023; Awadalla et al., 2023; Zhang et al., 2023; Jin et al., 2023b; Aiello et al., 2023). These efforts frequently utilize continuous embeddings (Radford et al., 2021; Li et al., 2022) to encode visual information into embeddings for inputting into language models. While these approaches benefit from CLIP's cross-modal understanding to encode textual information from images, their ability to predict text from visual input is limited, as is their capacity to learn from diverse visual-language formats. In contrast, our autoregressive model, which processes "tokens in, tokens out," allows greater flexibility in modeling various formats, including image-text, text-image, text-video, video-text, and pure formats like video, image, or text. Our method is compatible with these prior works, making it an interesting future direction to combine continuous embeddings as input with discrete tokens and a long-context autoregressive model.

6 CONCLUSION

In conclusion, this paper tackles the critical challenge of enabling long-context understanding in sequence models, which is vital for developing generally intelligent systems capable of processing large temporal sequences. By exploring the development of 1M context language and video-language models, the work sets new benchmarks in language retrieval and long video understanding. We outline approaches to data curation and progressive context extension, accompanied by an efficient open-source implementation for scalable training on long sequences. Moreover, we open-source a family of 7B parameter models capable of handling over 1M tokens in text and video.

Limitations. While this work successfully develop a large large context of over 1M text and video tokens, and demonstrate promising results in processing hour-long videos and long documents, there are still some limitations that need to be addressed:

- Improved tokenization and embedding. This work uses a vanilla image tokenizer for images and frame-by-frame tokenization for videos. Future work could explore video tokenization that takes time redundancy into account, as well as including continuous embeddings as input to enrich image understanding.
- Limited scale. Our models use more tokens per parameter than Chinchilla's recommendation, but being much smaller than current large language models (100B+ parameters), our findings may not directly apply to them. Extrapolating to larger scales should be done cautiously, as different scaling behaviors could emerge at those larger sizes.

REFERENCES

Emanuele Aiello, Lili Yu, Yixin Nie, Armen Aghajanyan, and Barlas Oguz. Jointly training large autoregressive multimodal models. *arXiv preprint arXiv:2309.15564*, 2023.

ArizeAI. Needle in a haystack - pressure testing llms. https://github.com/Arize-ai/LLMTest_NeedleInAHaystack, 2023.

Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023.

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1728–1738, 2021.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

William Brandon, Aniruddha Nrusimha, Kevin Qian, Zachary Ankner, Tian Jin, Zhiye Song, and Jonathan Ragan-Kelley. Striped attention: Faster ring attention for causal transformers. *arXiv preprint arXiv:2311.09431*, 2023.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

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

Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. https://github.com/kakaobrain/coyo-dataset, 2022.

Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793*, 2023a.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023b.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*, 2023.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

Facebook. Fully Sharded Data Parallel: faster AI training with fewer GPUs — engineering.fb.com. https://engineering.fb.com/2021/07/15/open-source/fsdp/, 2023. [Accessed 16-May-2023].

Chaoyou Fu, Yuhan Dai, Yongdong 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.

Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *European Conference on Computer Vision*, pages 89–106. Springer, 2022.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Xinyang Geng and Hao Liu. Openllama: An open reproduction of llama. *URL: https://github.com/openlm-research/open_llama*, 2023.

Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. Koala: A dialogue model for academic research. *Blog post, April*, 1, 2023.

gkamradt. Needle in a haystack - pressure testing llms. https://github.com/gkamradt/LLMTest_NeedleInAHaystack/tree/main, 2023. [Online; accessed 7-Feb-2024].

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022a.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022b.

Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Chia-Yuan Chang, and Xia Hu. Growlength: Accelerating llms pretraining by progressively growing training length. *arXiv preprint arXiv:2310.00576*, 2023a.

Yang Jin, Kun Xu, Liwei Chen, Chao Liao, Jianchao Tan, Bin Chen, Chenyi Lei, An Liu, Chengru Song, Xiaoqiang Lei, et al. Unified language-vision pretraining with dynamic discrete visual tokenization. *arXiv preprint arXiv:2309.04669*, 2023b.

Vijay Korthikanti, Jared Casper, Sangkug Lym, Lawrence McAfee, Michael Andersch, Mohammad Shoeybi, and Bryan Catanzaro. Reducing activation recomputation in large transformer models. *arXiv preprint arXiv:2205.05198*, 2022.

Jinhyuk Lee, Anthony Chen, Zhuyun Dai, Dheeru Dua, Devendra Singh Sachan, Michael Boratko, Yi Luan, Sébastien MR Arnold, Vincent Perot, Siddharth Dalmia, et al. Can long-context language models subsume retrieval, rag, sql, and more? *arXiv preprint arXiv:2406.13121*, 2024.

Dacheng Li, Rulin Shao, Anze Xie, Eric P Xing, Joseph E Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. Lightseq: Sequence level parallelism for distributed training of long context transformers. *arXiv preprint arXiv:2310.03294*, 2023.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022.

Shenggui Li, Fuzhao Xue, Yongbin Li, and Yang You. Sequence parallelism: Making 4d parallelism possible. *arXiv preprint arXiv:2105.13120*, 2021.

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.

Hao Liu and Pieter Abbeel. Blockwise parallel transformer for large context models. *Advances in neural information processing systems*, 2023.

Hao Liu, Matei Zaharia, and Pieter Abbeel. Ring attention with blockwise transformers for near-infinite context. *International Conference on Learning Representations(ICLR)*, 2024.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023a.

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

Xiaoran Liu, Hang Yan, Shuo Zhang, Chenxin An, Xipeng Qiu, and Dahua Lin. Scaling laws of rope-based extrapolation. *arXiv preprint arXiv:2310.05209*, 2023c.

Ruipu Luo, Ziwang Zhao, Min Yang, Junwei Dong, Minghui Qiu, Pengcheng Lu, Tao Wang, and Zhongyu Wei. Valley: Video assistant with large language model enhanced ability. *arXiv preprint arXiv:2306.07207*, 2023.

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.

OpenAI. Gpt-4 technical report, 2023.

Suraj Patil, William Berman, Robin Rombach, and Patrick von Platen. amused: An open muse reproduction. *arXiv preprint arXiv:2401.01808*, 2024.

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

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.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7, 2023.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

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

Szymon Tworkowski, Konrad Staniszewski, Mikołaj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Miłoś. Focused transformer: Contrastive training for context scaling. *arXiv preprint arXiv:2307.03170*, 2023.

Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*, 2022.

Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*, 2023.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022.

Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

A Further Details

Model Flops Utilization. We trained our models using TPUv4-1024, which is approximately equivalent to 450 A100s, with a batch size of 8M using FSDP (Facebook, 2023) and BlockwiseRingAttention (Liu et al., 2024) for large contexts. Figure 8 shows the model FLOPS utilization (MFU) for each training stage. Blue color bars show language training and orange color bars show vision-language training. Our training achieves good MFUs even for very large context sizes.

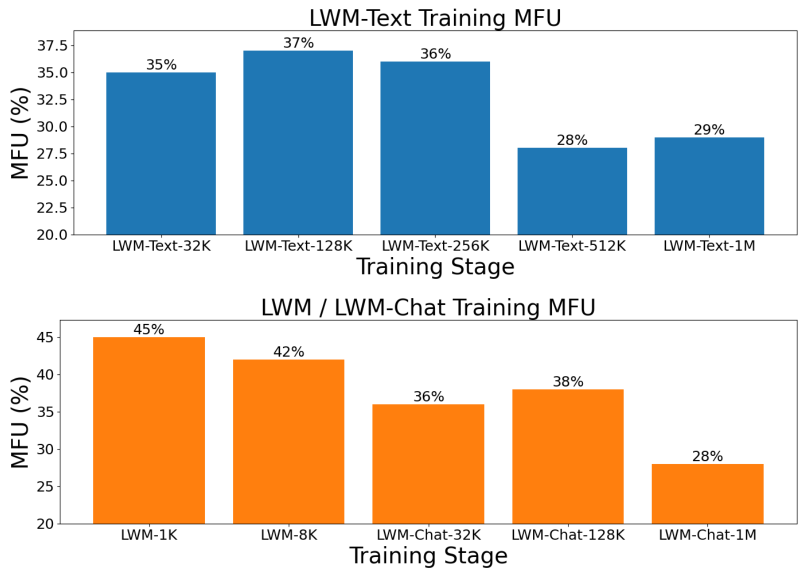


Figure 8 High MFU training across sequence lengths. Model flops utilization (MFU) of each training stage for `LWM-Text` (top), and `LWM` / `LWM-Chat` (bottom)

Training Loss Curves. Figure 9 and Figure 10 show the training loss curves for each stage of training the language and vision-language models respectively.

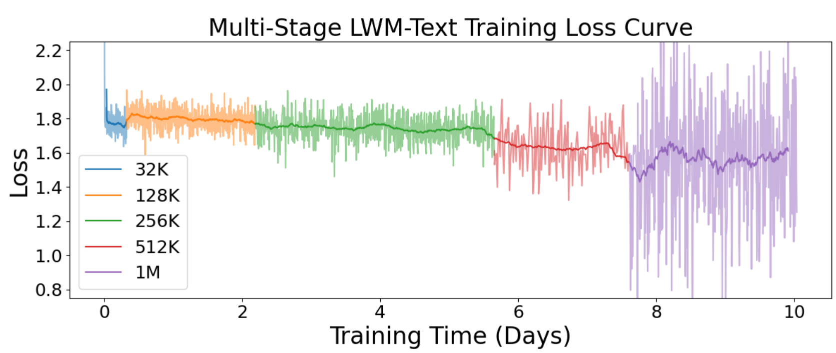


Figure 9 Training progress over multiple days for `LWM-Text`. Train loss curve for each training stage for `LWM-Text` models.

Training Hyperparameters. See Appendix **??**

Scaling Inference. We additionally scale our inference code to support million-length sequences by implementing RingAttention for decoding. Inference for such long sequences requires a minimum of v4-128 with a TPU mesh sharding of 32 tensor parallelism, and 4 sequence parallelism (ring dimension). We perform inference in pure single precision, where additional improvements can be made through techniques in scalability such as quantization.

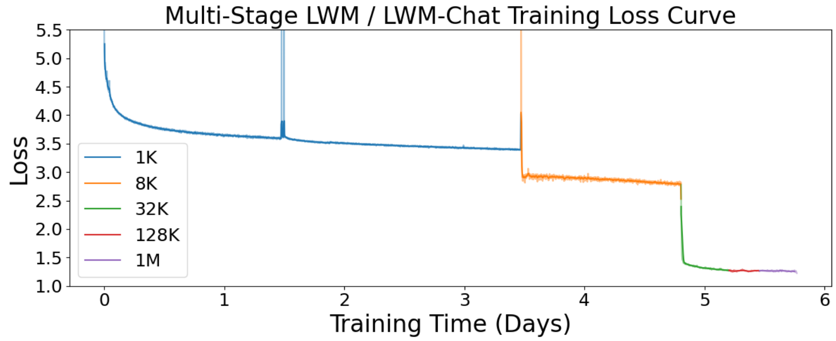


Figure 10 `Training progress over multiple days for LWM`. Train loss curve for each training stage for LWM and LWM-Chat models. Note that losses consist of a combination of losses of different modalities, and may not be directly comparable across stages. The sharp peak in the middle of 1K training is due to newly incporating EOF and EOV tokens into the vision codebook.

Table 6 LWM-Text Training Stages

	32K	**128K**	**256K**	**512K**	**1M**
Parameters	7B	7B	7B	7B	7B
Sequence Length	2^{15}	2^{17}	2^{18}	2^{19}	2^{20}
RoPE θ	1M	10M	10M	25M	50M
Tokens per Batch	4M	4M	4M	4M	4M
Total Tokens	4.8B	12B	12B	3B	1.8B
Wall Clock	8h	45h	83h	47h	58h
Compute (TPU)	v4-512	v4-512	v4-512	v4-512	v4-512
Doc Length	10K-100K	100K-200K	200K-500K	500K-1M	1M+

Table 7 LWM-Text-Chat Training Details

	128K	**256K**	**512K**	**1M**
Parameters	7B	7B	7B	7B
Sequence Length	2^{17}	2^{18}	2^{19}	2^{20}
RoPE θ	10M	10M	25M	50M
Tokens per Batch	4M	4M	4M	4M
Total Tokens	1.2B	1.2B	1.2B	1.2B
Wall Clock	6h	10h	20h	40h
Compute (TPU)	v4-512	v4-512	v4-512	v4-512

Table 8 LWM and LWM-Chat Training Stages

	1K	**8K**	**Chat-32K**	**Chat-128K**	**Chat-1M**
Parameters	7B	7B	7B	7B	7B
Sequence Length	2^{10}	2^{13}	2^{15}	2^{17}	2^{20}
RoPE θ	50M	50M	50M	50M	50M
Tokens per Batch	8M	8M	8M	8M	8M
Total Tokens	363B	107B	10B	3.5B	0.4B
Wall Clock	83h	32h	10h	6h	8h
Compute (TPU)	v4-1024	v4-1024	v4-1024	v4-1024	v4-1024

B Ablation Studies

B.1 Masked Sequence Packing

As mentioned in Section 4.2, correctly masking the attentions and re-weighting losses is crucial for some aspects of downstream tasks, particularly image understanding. Table 9 shows a comparison of our model with and without packing corrections. Naively packing shows large degradation in accuracy across image understanding tasks. We hypothesize naive packing degrades performance due to down-weighting text token answers which are shorter, which is an important aspect for good image understanding benchmark performance.

Table 9 Ablation study comparing standard independent packing and our masked sequence packing mechanisms across three tasks. Results show that masked sequence packing significantly improves performance across all tasks.

	VQAv2	SQA	POPE
Standard independent packing	48.3	34.8	62.5
Masked sequence packing (Ours)	**55.8**	**47.7**	**75.2**

B.2 Mixing Synthetic And Chat Data

We additionally evaluate the our model on MT-Bench (Zheng et al., 2023) to test its conversation ability. Table 10 shows the MT-Bench scores of for each of our models. Table 11 illustrates the relationship between the mix of chat and fact retrieval tasks and the performance on MT-Bench score and Needle Retrieval accuracy. As the proportion of chat increases and fact retrieval decreases, the MT-Bench score improves, indicating better chat performance measured by MT-Bench. Conversely, Needle Retrieval accuracy decreases, suggesting a trade-off where increasing chat interaction capabilities may reduce the system's precision in retrieving specific information or 'needles' from input context. Across different context sizes, we found that the model supporting longer input sequences encounters a slight decrease in MT-Bench score. We hypothesize that this is because we chose to train with fewer examples on longer sequence training and can be improved by simply training on more data. In addition, this trade-off may be resolved by acquiring higher quality long-context chat data that is closer to the chat distribution of the UltraChat dataset.

Table 10 Results on MT-Bench across different context sizes. Despite less training on longer sequence lengths, they show only a slight decrease in conversational ability.

Model	MT-Bench
LWM-Text-Chat-128k	4.62
LWM-Text-Chat-256k	5
LWM-Text-Chat-512k	4.83
LWM-Text-Chat-1M	4.19

Table 11 Relationship between the mix of chat and fact retrieval tasks and the performance on MT-Bench score and Needle Retrieval accuracy.

Chat / QA Mix	MT-Bench	Needle Acc
0% / 100%	2.42	100%
40% / 60%	4.14	100%
70% / 30%	4.62	96%
90% / 10%	5.1	55%
100% / 0%	5.8	31%

C More Single-Needle Retrieval Results

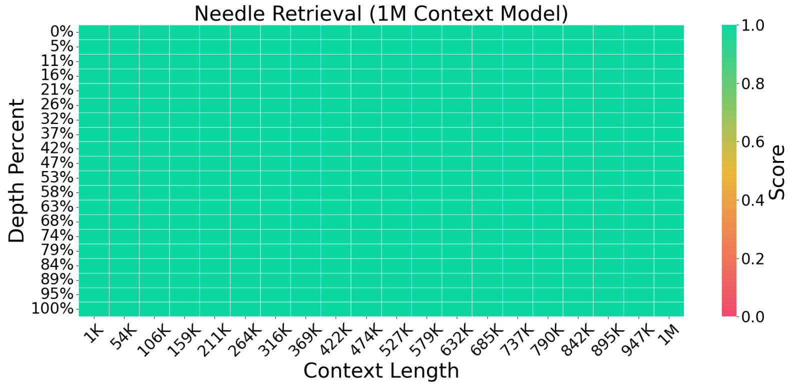


Figure 11 Needle retrieval task using the LWM-Text-Chat-1M model. The model demonstrates near-perfect retrieval accuracy across various positions within the 1M context window, as reflected by consistently high scores at different depth percentages and context lengths.

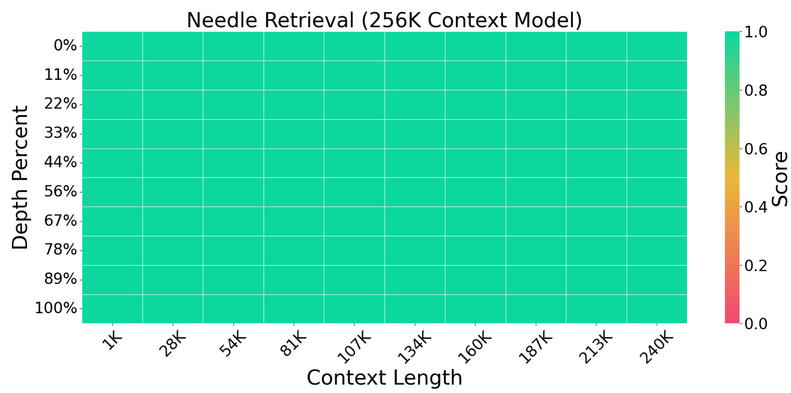


Figure 12 Single needle retrieval accuracy for the LWM-Text-Chat-256K model. The model achieves near-perfect retrieval performance across various positions in the 256K context window, as shown by consistently high scores across all depth percentages and context lengths.

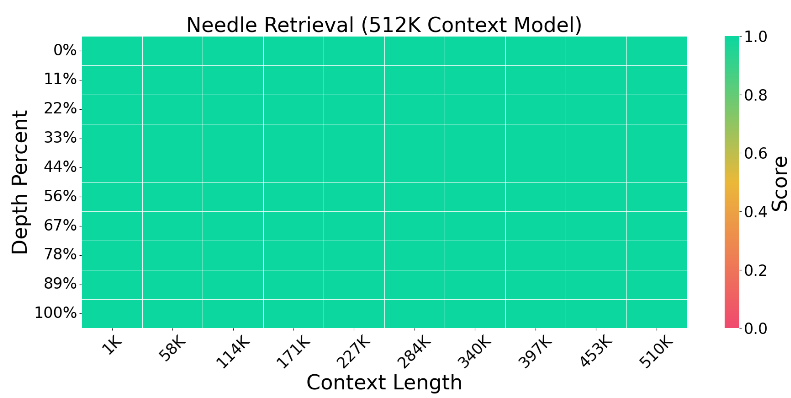


Figure 13 Single needle retrieval accuracy for the LWM-Text-Chat-512K model. The model demonstrates near-perfect retrieval performance across different positions within the 512K context window, as indicated by consistently high scores at varying depth percentages and context lengths.

D Details Of The Multi-needle Evaluations

We built on top of the original Needle in a Haystack problem (see Section 3.3.2 for details), which was to retrieve a single magic number randomly inserted into some long context. In this case, magic numbers are associated with random cities ("The magic number for San Francisco is 2521233" → "What is the magic number for San Francisco?"). We extend the task by introducing two variables N and R, where N is the number of needles (magic numbers + different cities) randomly inserted into the context, and R is the random subset of magic numbers asked to retrieve ("What are the magic numbers for San Francisco, . . . , and Bangkok?"). Correctness is computed by extracting out the numbers retrieved for each cities and checked with string matching.

E More Image Understanding Examples

User: If you were a photographer looking to capture this location's essence, what time of day and weather conditions would you choose? Describe the reasons behind your choice.

Assistant: I would choose the time of day to be during sunset. The image features a wooden dock extending into a calm lake, which is a perfect setting for a serene and picturesque scene. The sunset casts a warm, golden light on the lake, enhancing the tranquility of the scene. Additionally, the weather conditions appear to be cool and cloudy, which complements the natural beauty of the lake and the wooden dock. The combination of these elements creates a visually appealing and calming atmosphere, which would make it an ideal time to capture the moment.

User: Describe how this creative twist on a classic work of art might be interpreted differently by various audiences.

Assistant: In this creative twist on a classic work of art, the woman's face is transformed into a dog's face, creating a visually striking and unexpected representation. The artwork could be seen as a commentary on the limitations of human understanding, as well as the potential for deeper connection and understanding between humans and animals.

Figure 14 Question answering based on image input using LWM. The assistant provides detailed responses to questions about capturing the essence of a serene lakeside scene and interpreting a creative twist on a classic artwork featuring a dog.

F MORE VIDEO UNDERSTANDING EXAMPLES

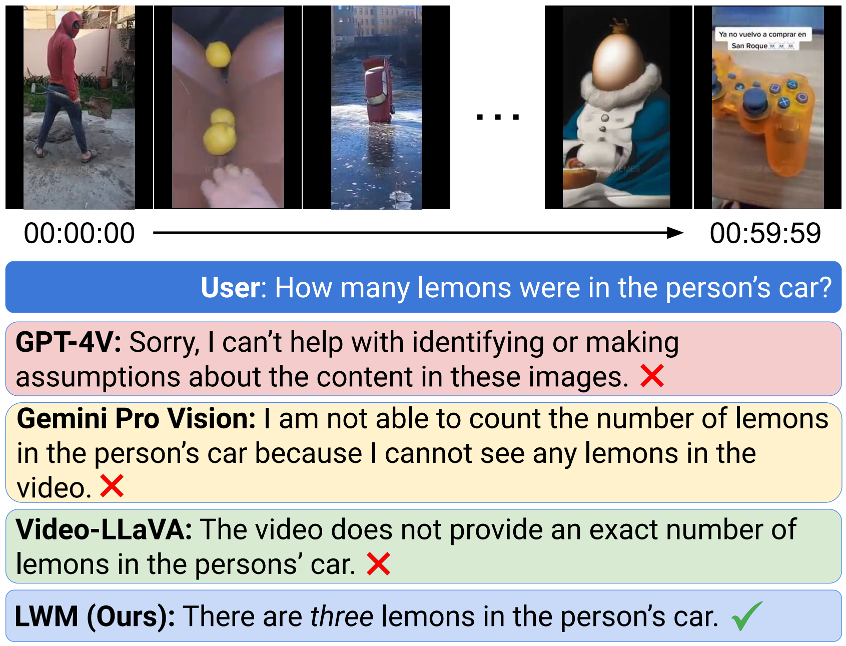


Figure 15 LWM demonstrates video understanding over a 1-hour video. The figure shows a comparison of different AI systems responding to a question about the number of lemons in a person's car. While GPT-4V, Gemini Pro Vision, and Video-LLAVA fail to provide a correct response, LWM accurately identifies that there are three lemons in the car.

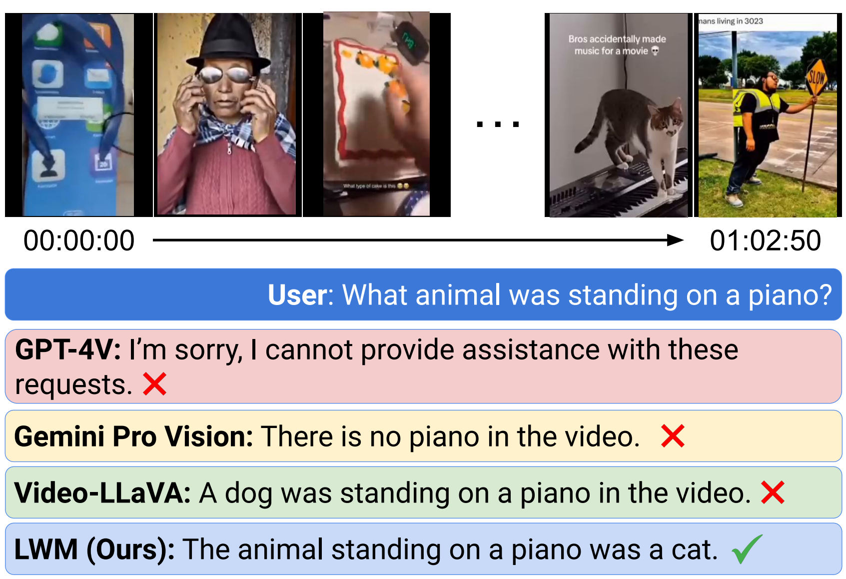
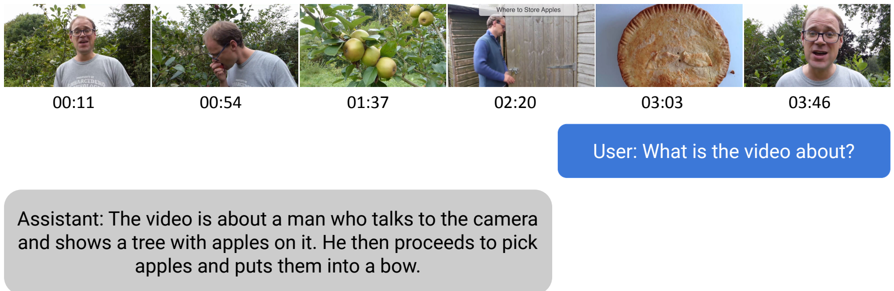
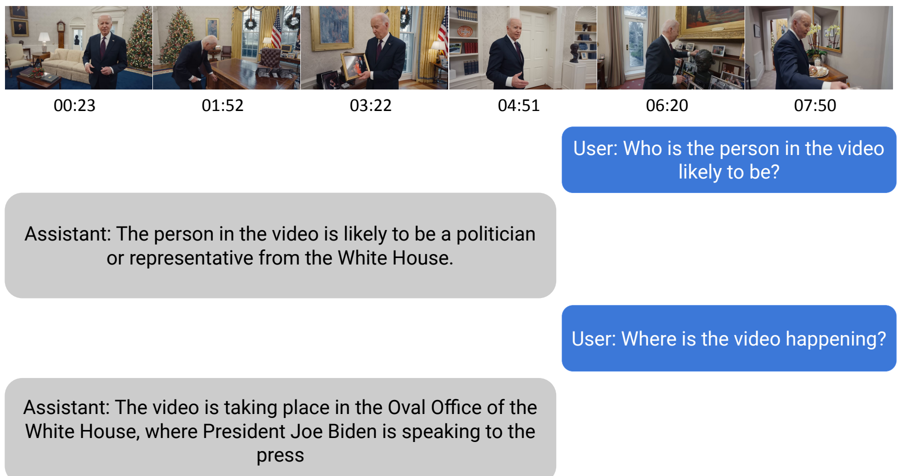


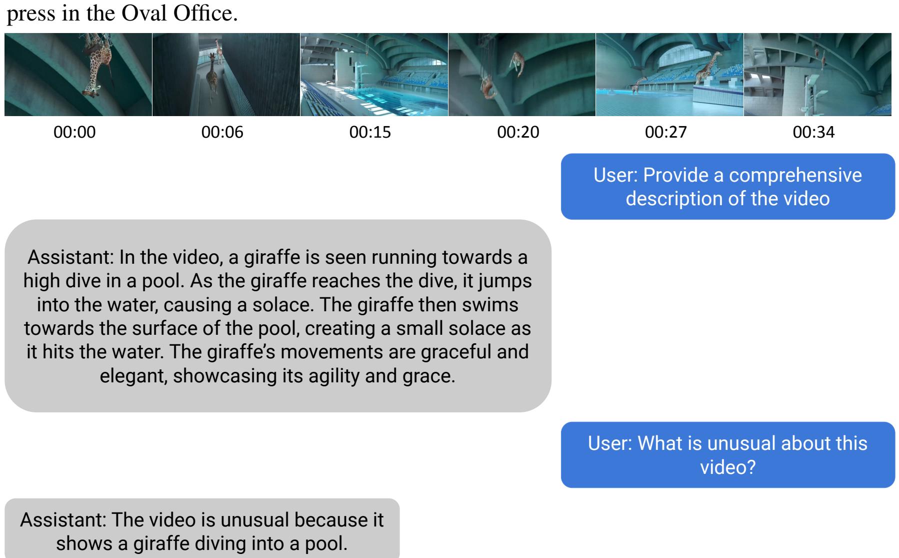
Figure 16 LWM demonstrates video understanding over a 1-hour video. The figure compares AI systems' responses to the question 'What animal was standing on a piano?' While GPT-4V, Gemini Pro Vision, and Video-LLAVA provide incorrect or incomplete answers, LWM correctly identifies that the animal standing on the piano was a cat.



(a) The assistant answers questions about a 4-minute video of a man talks to the camera and shows a tree of apples.

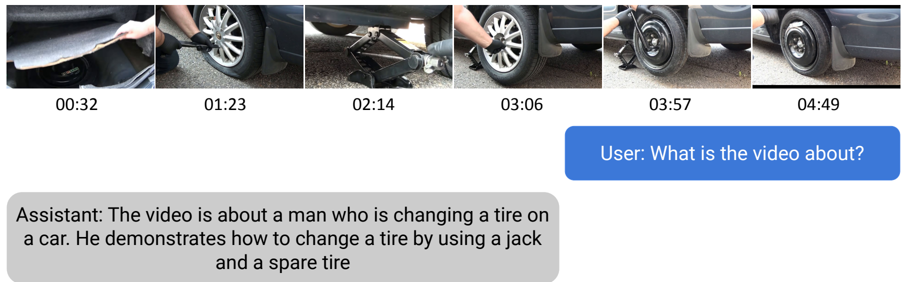


(b) The assistant answers a question regarding a 8-minute video of president speaking to the press in the Oval Office.

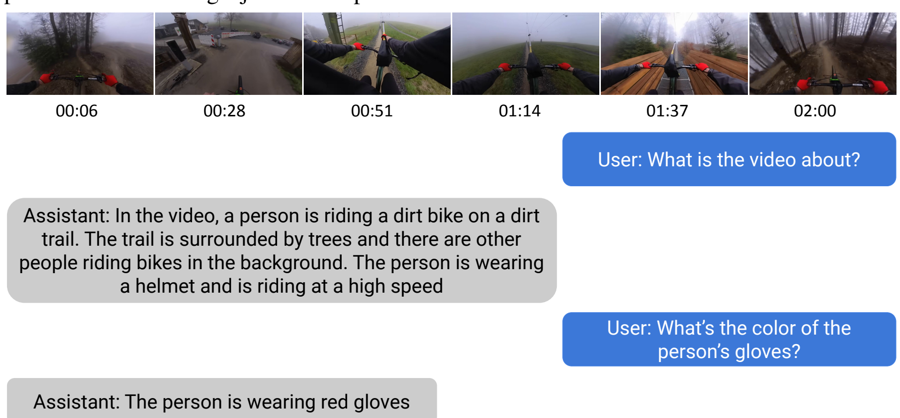


(c) The assistant answers a question about a 30-second video of a giraffe.

Figure 17 Answering questions about videos using LWM. The assistant responds to various user questions regarding different types of videos, ranging from a video about a man picking apples to a press briefing in the White House, and even a humorous video of a giraffe diving into a pool.



(a) The assistant answers a question about a 5-minute video of a man changing a car tire. The process involves using a jack and a spare tire.



(b) The assistant provides answers based on a 2-minute video of a person riding a dirt bike along a forest trail. The rider wears a helmet and red gloves, traveling at high speed.

Figure 18 The system (LWM) successfully answers questions about video content.

G DETAILS OF QUALITATIVE VIDEO UNDERSTANDING EVALUATION

For qualitative evaluation of our videos, we source various videos from YouTube that cover a range of topics, such as ego-centric camera, how to videos, interviews, and animations. We evaluate all videos at 1FPS, and sample uniformly a max number of frames for videos that are longer than what our video can support at 1 FPS. Videos are additionally resized and center cropped to 256×256 resolution before inputting into the model.

H More Image Generation Examples

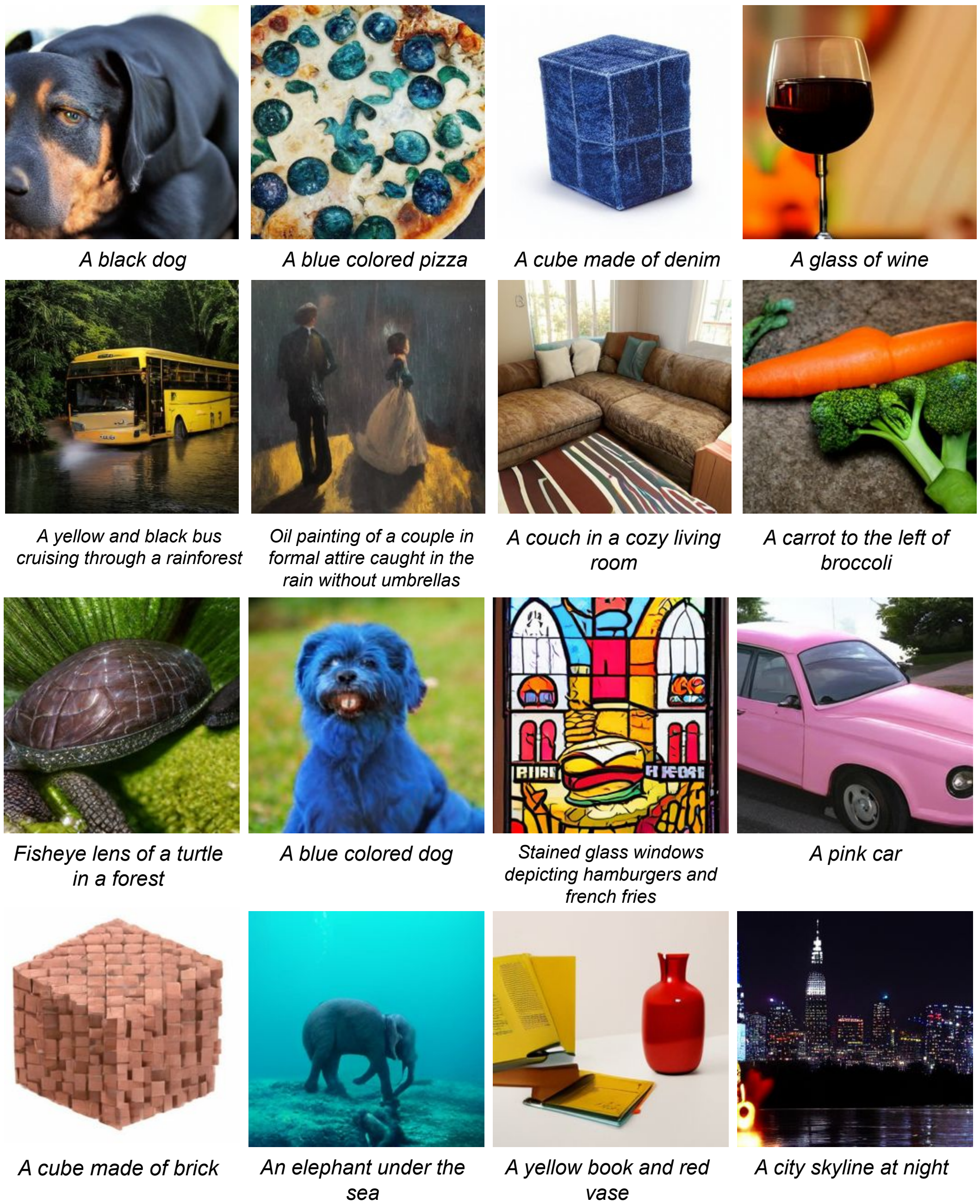


Figure 19 Images generation using LWM, showcasing various scenes and objects.

I MORE VIDEO GENERATION EXAMPLES

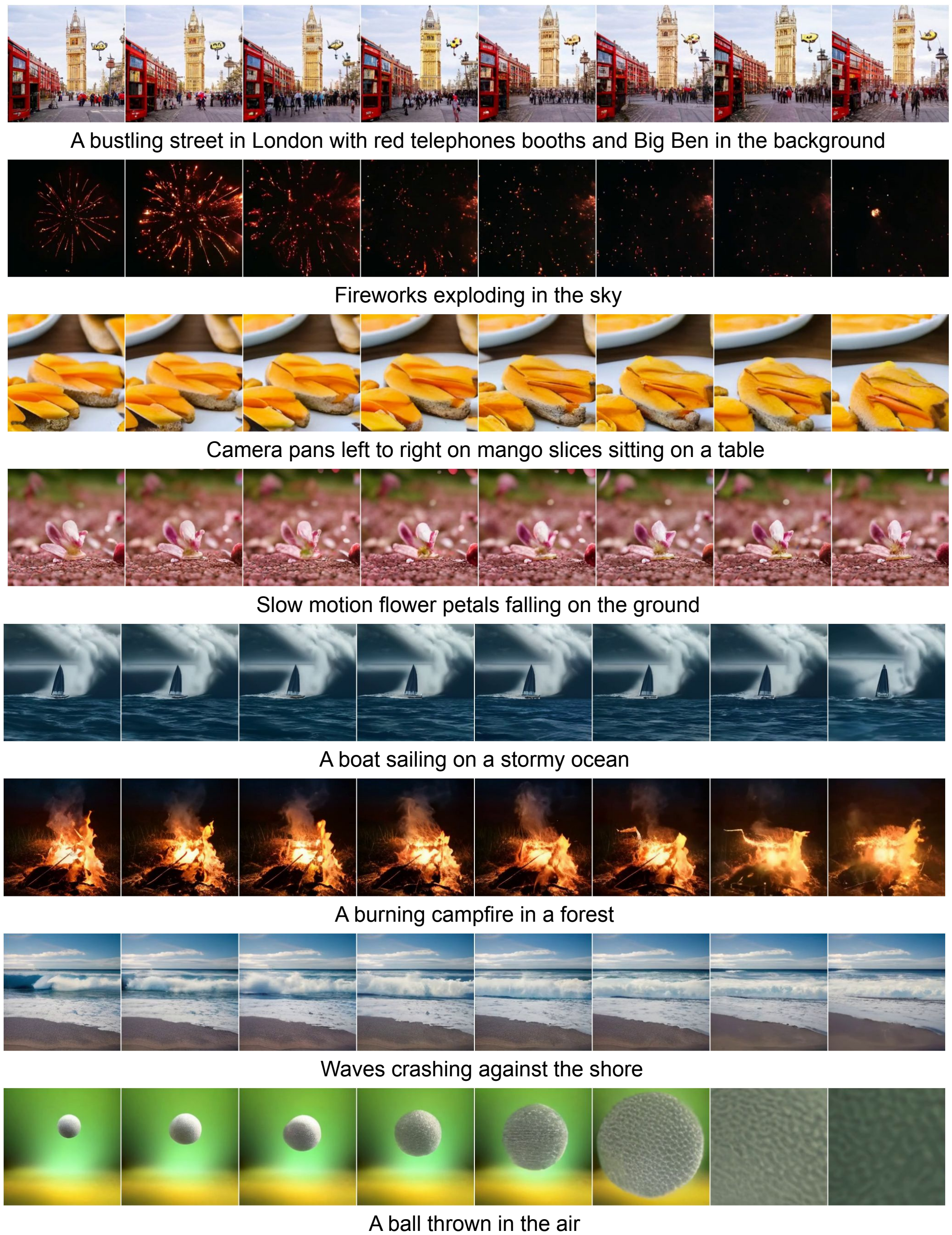


Figure 20 Video sequences generated using LWM, showing various scenes.

J TRAINING HYPERPARAMETERS

Table 12 LWM-Text Training Stages

	32K	**128K**	**256K**	**512K**	**1M**
Parameters	7B	7B	7B	7B	7B
Initialize From	LLaMA-2 7B	Text-32K	Text-128K	Text-256K	Text-512K
Precision	float32	float32	float32	float32	float32
Sequence Length	2^{15}	2^{17}	2^{18}	2^{19}	2^{20}
RoPE θ	1M	10M	10M	25M	50M
Tokens per Batch	4M	4M	4M	4M	4M
Total Tokens	4.8B	12B	12B	3B	1.8B
Total Steps	1200	3000	3000	720	450
LR Schedule	Constant	Constant	Constant	Constant	Constant
LR Warmup Steps	100	200	200	50	25
LR	4×10^{-5}	4×10^{-5}	4×10^{-5}	4×10^{-5}	4×10^{-5}
Compute (TPU)	v4-512	v4-512	v4-512	v4-512	v4-512
Mesh Sharding	1,-1,4,1	1,-1,8,1	1,-1,16,1	1,-1,16,2	1,-1,16,4

Table 13 LWM-Text-Chat Training Details

	128K	**256K**	**512K**	**1M**
Parameters	7B	7B	7B	7B
Initialize From	Text-128K	Text-256K	Text-512K	Text-1M
Precision	float32	float32	float32	float32
Sequence Length	2^{17}	2^{18}	2^{19}	2^{20}
RoPE θ	10M	10M	25M	50M
Tokens per Batch	4M	4M	4M	4M
Total Tokens	1.2B	1.2B	1.2B	1.2B
Total Steps	300	300	300	300
LR Schedule	Constant	Constant	Constant	Constant
LR Warmup Steps	25	25	25	25
LR	4×10^{-5}	4×10^{-5}	4×10^{-5}	4×10^{-5}
Compute (TPU)	v4-512	v4-512	v4-512	v4-512
Mesh Sharding	1,-1,4,1	1,-1,8,1	1,-1,16,1	1,-1,16,2

Table 14 LWM / LWM-Chat Training Stages

	1K	**8K**	**32K**	**128K**	**1M**
Parameters	7B	7B	7B	7B	7B
Initialize From	Text-1M	1K	8K	32K	128K
Precision	float32	float32	float32	float32	float32
Sequence Length	2^{10}	2^{13}	2^{15}	2^{17}	2^{20}
RoPE θ	50M	50M	50M	50M	50M
Tokens per Batch	8M	8M	8M	8M	8M
Total Tokens	363B	107B	10B	3.5B	0.4B
Total Steps	45000	14000	1200	450	50
LR Schedule	Cosine	Cosine	Cosine	Cosine	Cosine
LR Warmup Steps	1000	500	100	50	5
Max LR	6×10^{-4}	6×10^{-4}	8×10^{-5}	8×10^{-5}	8×10^{-5}
Min LR	6×10^{-5}	6×10^{-5}	8×10^{-5}	8×10^{-5}	8×10^{-5}
Compute (TPU)	v4-1024	v4-1024	v4-1024	v4-1024	v4-1024
Mesh Sharding	1,-1,1,1	1,-1,1,1	1.-1.4,1	1.-1.8,1	1,-1,16,4