ViLL-E: Video LLM Embeddings for Retrieval

Rohit Gupta and Jayakrishnan Unnikrishnan and Fan Fei and Sheng Liu
{rohitgpt, jayunn, feiff, shenlu}@amazon.com

Son Tran
sontran@amazon.com

Mubarak Shah
shah@crcv.ucf.edu

Abstract

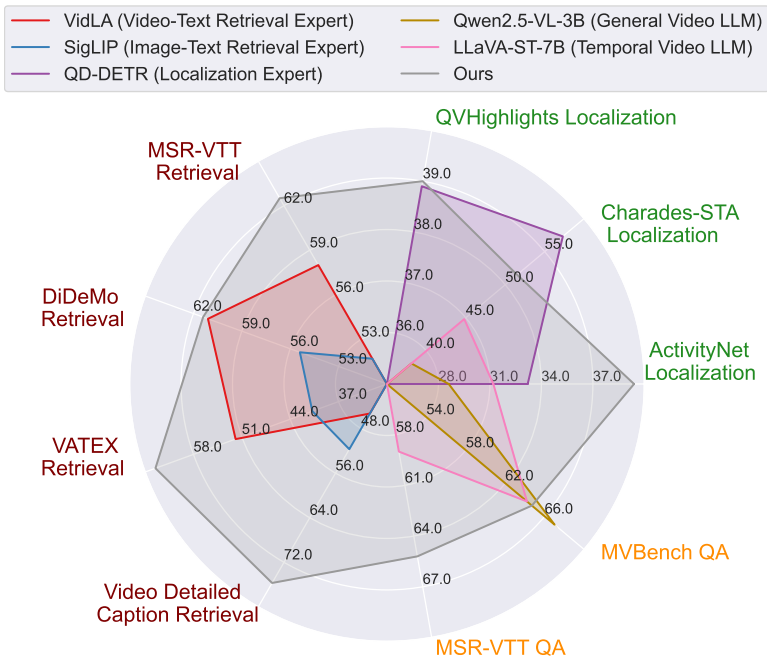
Video Large Language Models (VideoLLMs) excel at video understanding tasks where outputs are textual, such as Video Question Answering and Video Captioning. However, they underperform specialized embedding-based models in Retrieval tasks, such as Text-to-Video Retrieval and Moment Retrieval. We introduce ViLL-E (Video-LLM-Embed), a unified VideoLLM architecture endowed with a novel embedding generation mechanism that allows the model to "think longer" for complex videos and stop early for easy ones. We train this model with a three-stage training methodology combining generative and contrastive learning: initial large-scale pre-training with video-caption pairs; followed by continual training on a smaller, detailed-caption dataset; and concluding with task-specific fine-tuning on a novel multi-task dataset covering Video QA, Temporal Localization, Video Retrieval, and Video-Text Matching. Our model significantly improves temporal localization (on avg. 7% over other VideoLLMs) and video retrieval (up to 4% over dual encoder models), achieving performance comparable to state-of-the-art specialized embedding models while remaining competitive on VideoQA tasks. Furthermore, our joint contrastive-generative training unlocks new zero-shot capabilities, significantly outperforming state-of-the-art methods in composed video retrieval (+5% over SotA) and retrieval from long text (+2% over SotA).

1 Introduction

Large Language Models (LLMs) have demonstrated significant capabilities in a "single model multi-task approach" setting, effectively solving various natural language understanding tasks. Inspired by the success of LLMs [7, 41, 13, 34], numerous studies have integrated additional modalities like images [19, 26, 3] and audio [38, 56] into these models. These Multimodal LLMs models excel at most vision tasks that can be formulated as text *generation problems*. Of specific interest are Video Large Language Models (VideoLLMs) [29, 24, 57, 21], which incorporate videos into LLMs by encoding video frames and aligning visual features with LLM feature spaces via projection layers, enabling video understanding through tasks like captioning and QA.

However, this generative approach has limitations. Many video tasks, like Text-to-Video Retrieval (T2V), require embedding-based matching rather than text generation. Such tasks typically use specialized models with aligned video-text embeddings, fine-tuned per dataset. Currently these specialized methods lead performance on standard benchmarks. Even VideoLLM variants that are specialized to attempt temporal reasoning (e.g. LLaVA-ST [18]) trail expert detectors such as QD-DETR [30] by double-digit margins on ActivityNet [15] and Charades-STA [10]. Figure 1 underscores the point: every top performer in video retrieval, temporal localisation or cross-modal search (VidLA [37], QD-DETR [30], SigLIP [53]) achieves its performance with embeddings, while Video-LLMs dominate the Video-QA column. Also, recent NLP research has demonstrated that LLMs can be transformed into strong retrieval models with contrastive finetuning on surprisingly little data (see, e.g., GRIT [31] or E5 [43]). This raises the question: Instead of maintaining two separate stacks, can a single model handle both generative tasks and extract discriminative video/text embeddings, addressing the retrieval and localization gaps?

Motivated by these observations, we propose an approach that combines generative and embedding tasks during Video-LLM training, aiming for strong performance on generative tasks like VideoQA, while also enabling embedding generation for tasks such as video retrieval—previously unsupported by VideoLLMs. Our approach builds on PaliGemma [6], a standard multi-modal LLM



Method	Architecture LLM	Embeddings
VidLA	✗	✓
SigLIP	✗	✓
QD-DETR	✗	✓
Qwen2.5-VL	✓	✗
LLaVA-ST	✓	✗
Ours	✓	✓

Method	Capabilities Ret.	Loc.	QA	Cap.
VidLA	✓	✗	✗	✗
SigLIP	✓	✗	✗	✗
QD-DETR	✗	✓	✗	✗
Qwen2.5-VL	✗	◆	✓	✓
LLaVA-ST	✗	◆	✓	✓
Ours	✓	✓	✓	✓

Figure 1: **Left & Bottom Right**: VideoLLMs lag expert models on some retrieval-based tasks, e.g. Temporal Localization (in green [17, 10, 15]), and are incapable of others, e.g. Text-to-Video Retrieval (in red [49, 1, 44]). A key difference between existing VideoLLMs and state of the art expert models in these tasks is the use of embeddings(**Top Right**). Our approach, **ViLL-E** (VideoLLM-Embed, *pronounced willy*) equips VideoLLMs with embedding generation. ViLL-E matches SotA on retrieval tasks while remaining competitive on generation tasks such as VideoQA (in orange [47, 22]). ◆ indicates limited capabilities.

and adds a learnable attentive-pooling "embedding head" after the LLM decoder. The embedding head is designed to adapt to video complexity during inference. This allows the model to "think longer" for hard videos but return quickly for easy ones. We propose three-stage training, which begins with large-scale pre-training on video-caption pairs using joint supervision: next-word prediction for generation and a multi-modal contrastive loss for embedding. In the second stage, we continue pre-training on a smaller dataset of videos with high-quality generated captions. The final stage involves instruction finetuning on a multi-task dataset covering VideoQA, captioning, localization, and retrieval. Our proposed model, ViLL-E (pronounced willy), is competitive with VideoLLMs on VideoQA and outperforms them on localization. Our embeddings narrow the gap with non-LLM localization expert methods and enable SoTA performance on video retrieval.

In summary, the key contributions of our work are:

- We present ViLL-E, an unified solution to diverse video-understanding tasks, which is the first VideoLLM that can generate text responses or video & text embeddings with a single model.
- An effective three-stage training pipeline: (1) large-scale joint captioning + contrastive learning, (2) continued pre-training on high quality captions, and (3) multitask fine-tuning.
- Our single model ViLL-E, outperforms VideoLLMs by more than 7% across three moment retrieval benchmarks, rivals them on VideoQA, and is competitive with expert retrieval models.
- ViLL-E also solves new tasks in zero-shot setting, including long-caption retrieval, compositional video search, and two-stage retrieve-and-match pipelines using a single model.

2 Related Work

Video Expert Models for Retrieval and Localization: Prior works on text-video and video-text retrieval [50, 28, 11, 37, 45] typically extend dual-encoder image-text models such as CLIP [33] or BLIP [20] to video by aggregating the CLIP vision encoder outputs from multiple frames with architecture adaptations to model temporal relations [5, 8]. For retrieval they use dense vector search in the embedding space while a few other methods use an additional reranking step with cross-modality fusion layers [23, 9]. Prior works on temporal localization such as Moment-DETR [17], QD-DETR [30], TR-DETR [39] and UnLoc [51] build on ideas from retrieval. Typically Video-Text Fusion is also employed to score the match between a given text and video segment for compatibility. Specific unique

ideas include additional modules like time span prediction in QD-DETR and learnable moment queries in Moment-DETR. Vid2Seq [52] utilizes dense captioning trained on large video-text datasets instead of retrieval for localization. These expert models are close to the state of the art frontier for their respective tasks, however they are specialized for one task, and are typically finetuned individually for each dataset.

Video LLMs: Many recent works have extended ImageLLMs such as LLaVA [26] to videos. VideoLLaVA [24] is a straightforward extension that utilizes LanguageBind [59] encoder instead of CLIP because its trained on both images and videos. Some works have tried to extend a BLIP2 style architecture to Videos, namely VideoChat [21] and Video LLaMA [57]. In addition, some works have focused on creation of strong IFT datasets specific to video tasks, VideoChatGPT [29] created IFT data using ActivityNet dense captions and semi-automated process with questions generated by GPT3.5 using dense text captions and open vocabulary attributes. In VideoChat2 [21], another diverse IFT dataset was created by combining many (30+) datasets together. These models excel at video question answering and other text generation tasks, however they often struggle at temporal localization tasks and don't generate embeddings, which are necessary for video retrieval.

Video LLMs with Localization IFT: Recent works have sought to instill temporal awareness and localization skills in video LLMs. Broadly, the architectures of these approaches can be categorized into LLAVA-like or BLIP2-like [19] depending on whether or not they use a Q-Former for token reduction before input to LLM. TimeChat [36] uses sliding window Q-Formers as a projection layer, VTimeLLM [12] uses MLP, HawkEye [46] uses simple Q-Former and Momentor [32] proposes a parallel temporal perception module as part of its projection layers. LLaVA-ST [18] inserts special spatio-temporal tokens into the language stream, and aligns them with vision features through a language-aligned position embedding. To impart temporal awareness to the models, these works use IFT datasets with specialized tasks such as dense captioning, timestamp generation for localization, repetition counting and more. On localization tasks, these models outperform general VideoLLMs, however they are unable to outperform expert models which do not use LLMs.

LLMs for Embedding generation: Some works in the NLP domain [43, 16] have demonstrated that generative LLMs can be finetuned with a limited amount of data (relative to the scale of generative pre-training) to generate strong embeddings for a variety of downstream tasks including retrieval. GRIT [31] has demonstrated that a pretrained LLM can be jointly finetuned to improve on both generative instruction following and embedding generation, maintaining strong performance on both sets of tasks. Some recent concurrent works such as VLM2Vec [14], GME [58] and MM-Embed [25] have sought to use LLMs for multimodal embeddings, but are limited to images.

Our approach unifies text-generative and embedding-generative capabilities into a single VideoLLM model, which is competitive with task-specific and dataset-specific expert models on temporal localization and retrieval tasks that require embedding generation, as well as with Video LLMs on text generation like QA and captioning tasks.

3 Method

In this section, we present the architecture and training methodology of our VideoLLM, designed to leverage multi-modal video data to perform both generative and embedding tasks. Our approach involves a multi-stage and multi-task process that includes continued multi-modal pre-training to improve video language alignment, and subsequent multi-task finetuning.

3.1 Model Architecture

Our model (see Figure 2) features an LLM backbone, shared by both its generative and embedding tasks, integrated with a vision encoder module designed to effectively process video inputs. We initialize the model's weights using a pretrained multimodal LLM (PaliGemma-3B [6]). Following PrefixLM (Raffel et al., 2020) [34], visual tokens and input prompts are processed with bidirectional attention, whereas causal attention is applied to the suffix autoregressively generated by the model. For text generation tasks (e.g. VideoQA), our model follows the same approach as PaliGemma.

3.2 Embedding Head Design

We introduce a learnable embedding head to support retrieval-oriented tasks with four design goals: (1) dedicated modeling capacity separate from generative tasks, (2) a bottleneck that yields dense embeddings without overly restricting representation power, (3) adaptability to variable token lengths for videos of different complexity, and

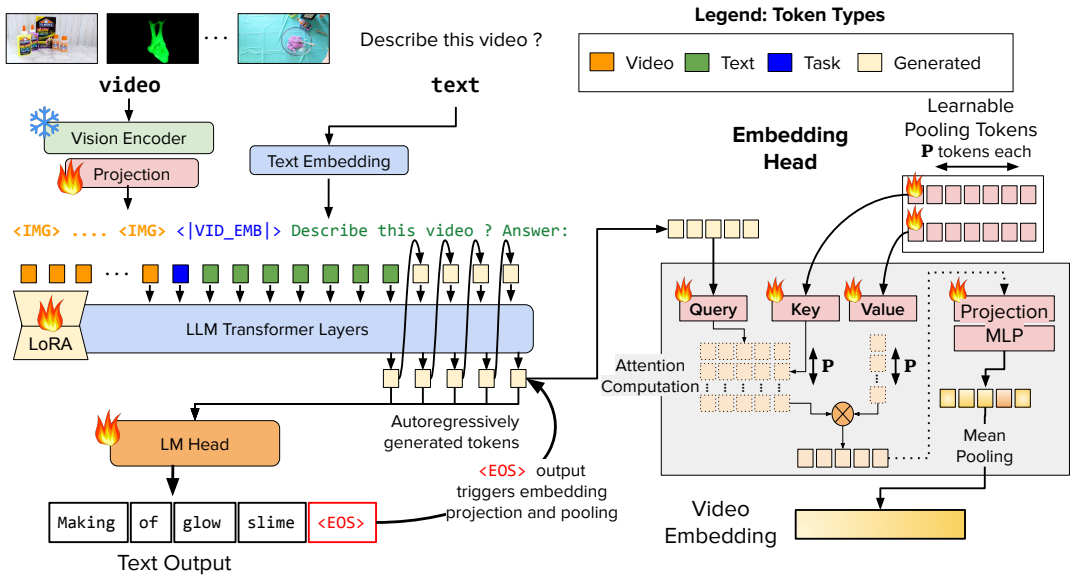


Figure 2: Our model ViLL-E is a multi-modal LLM which has been equipped with additional embedding generation capabilities. Individual video frames are first encoded by a pretrained vision encoder, with extracted features projected into embedding vectors. These visual embeddings are combined with textual embeddings of an input prompt and jointly processed autoregressively through a pre-trained Multi-Modal LLM's transformer layers. After the model autoregressively generates an end-of-sequence (`|EOS|`) token, all the generated tokens are collected and passed to our embedding head. The embedding head employs an attention-based learnable pooling mechanism to extract and aggregate relevant information across the tokens; The generated tokens are further passed through projection layers and mean pooled to yield a compact, informative video embedding. The ability to dynamically generate different number of tokens to feed the embedding head allows the model to handle videos of different complexity levels and is key for its success at varied retrieval tasks.

(4) parameter efficiency. We evaluated several approaches: **1. Attention-Free pooling**: Pools penultimate layer LLM tokens into a fixed-size embedding, lightweight but dependent on heuristic pooling (cf. Sentence-BERT [35]). **2. Self-Attention**: Adds an extra transformer layer, offering some task-specific capacity but redundant with prior LLM layers and lacking a bottleneck. **3. Q-Former**: Learned queries cross-attend to LLM outputs to produce fixed-length embeddings (e.g., BLIP-2 [19]); effective but inflexible for variable content length, as output size always equals number of learnable queries. **4. 'K-Former'**: Learns only keys, enabling variable-length outputs. **5. Our 'KV-Former'**: Uses LLM tokens as queries with P learnable key and value "pooling tokens," each. Enabling variable-length outputs while acting as a bottleneck akin to dictionary learning.

Our embedding head (on the right in Fig. 2) receives tokens from the final LLM transformer layer and employs a learnable attentive pooling mechanism. Specifically, a transformer-based self-attention layer adaptively weighs tokens, using the LLM's tokens as queries and P learnable keys and values ("pooling tokens"). After this attention operation, the weighted representations are projected into the embedding space through a multi-layer perceptron (MLP) and subsequently mean-pooled.

For the video embedding generation, our model uniquely leverages an end-of-sequence (`<EOS>`)-triggered embedding generation mechanism. This mechanism allows the model to utilize variable inference steps tailored to each video's complexity - more steps for complex videos needing deeper analysis, fewer for simple ones. This adaptive computation approach substantially improves performance over existing fixed-step LLM-based embedding methods, offering both efficiency and improved representational quality. The proposed design is flexible and can be leveraged for various tasks, e.g. text and video embeddings for temporal localization, composed video retrieval etc. For text embedding generation, we rely on standard inference similar to prior works [31], with entire text embeddings being generated in a single step.

3.3 Training Objectives

We follow a three-stage training strategy: (1) pre-training on a large, diverse, and noisy dataset to jointly improve video captioning and embedding capabilities; (2) continued pre-training on a smaller, higher-quality dataset; and (3) multi-task finetuning to unlock the model's performance across diverse

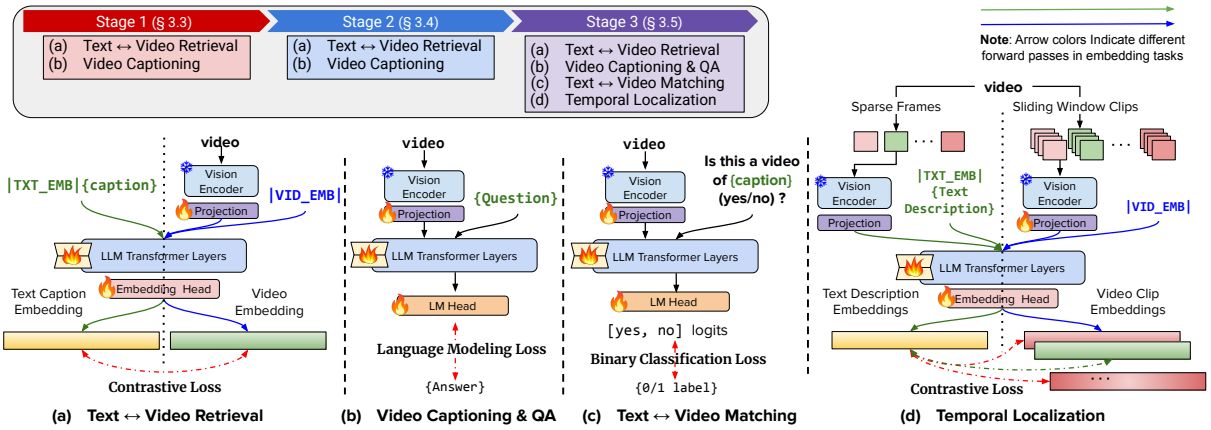


Figure 3: Three stages of training. **Stage 1 (§ 3.4)**, the large scale joint contrastive and generative pre-training, which utilizes (a) Video-Text retrieval task and (b) Video captioning on the large scale `Shutterstock` video dataset. **Stage 2 (§ 3.5)**, continues a similar joint pre-training approach (retrieval + captioning tasks as shown in (a) and (b)) on a smaller high quality dataset created using `Claude-3-Sonnet`. **Stage 3 (§ 3.6)**, our multi-task fine-tuning stage, in addition to the prior tasks (retrieval + captioning) integrates (b) Video QA, (c) Video-Text matching and (d) Temporal Localization to improve all-around retrieval capabilities.

downstream tasks. To ensure parameter efficiency during fine-tuning, we adopt Low-Rank Adapters (LoRA). The vision projection module and embedding head are fully trained to maximize representation learning. Our model is designed to effectively balance multimodal generative capabilities with state-of-the-art performance on embedding-oriented retrieval tasks. Across our training stages, a total of four different tasks are used, which are described below.

Text–Video Retrieval (Fig. 3a). This task aligns video and text embeddings using CLIP-like [33] contrastive loss, encouraging each paired example to outrank in-batch negatives.

Video Captioning & Question–Answering (Fig. 3b). The goal of captioning is for the model to generate accurate descriptions of videos. The model learns to predict each token in the caption, given the previous tokens and the video. For QA, the model generates answers to questions about the video; the question serves as part of the prompt, guiding the model's attention to answer accurately. We use the next-token prediction loss.

Text–Video Matching (Fig. 3c). As a complement to the retrieval task we also formulate a matching task where the model is trained to identify if a given video and caption form a matching pair. Binary decisions are supervised by labels.

Temporal Localization (Fig. 3d). Localization requires the model to match specific video segments to text annotations while distinguishing similar but incorrect segments. Sliding window hard negative mining provides challenging non-matching segments with limited temporal overlap (IoU < 0.2), helping the model learn to differentiate similar content. We use a standard contrastive formulation with one positive and two hard negatives per query. *Complete notation and full equations are provided in the Appendix B.* The following subsections discuss our multi-stage training strategy in detail.

3.4 Training Stage 1: Generative-Contrastive Pre-Training

In stage 1, we utilize a large video-caption dataset and jointly train the network using generative and contrastive supervision. In the first forward pass, for each video caption pair, video input tokens are passed along with a prompt asking the model to describe the video. Captioning output for the generative part and the video embedding are generated simultaneously in this pass. This is followed by a second computationally lighter forward pass with text-only input to generate the text embedding. The next token prediction loss is applied to the caption generation task, and the CLIP multimodal contrastive loss is applied for the embedding generation task. The generative component aims to improve the model's ability to create coherent and contextually relevant textual descriptions of the video content, while the contrastive component enhances its discriminative power by teaching it to differentiate between relevant and irrelevant text-video pairs. This dual-objective training strategy ensures that the model not only learns to generate high-quality textual content but also develops a better alignment between modalities.

3.5 Training Stage 2: Continued Pre-Training

While our pre-training data has a diverse collection of videos, the captions tend to be short summaries

of the video content, often missing many details in the video. Since preserving important details from the video in our VideoLLM's feature space is beneficial for downstream tasks, we supplement our pre-training with a small intermediate pre-training phase using higher quality captions for a subset of videos from our pre-training set. The dataset for intermediate pre-training with high quality descriptive captions is discussed in detail in Section 4.

3.6 Training Stage 3: Multi-Task Finetuning

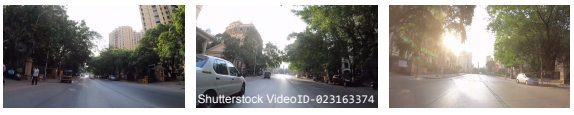
In this stage, a high-quality finetuning dataset is used combining supervision across four tasks: captioning or question answering (QA), retrieval, matching and temporal localization. The finetuning process integrates these tasks to ensure that the model develops well-rounded capabilities, enabling it to handle diverse real-world scenarios that involve multi-modal understanding and generation.

4 Training Data

In this section, we provide details about the datasets used in the different training stages of our model.

4.1 Stage 1: Large Scale Pre-Training Data

For pre-training, we use video–caption pairs from the licensed Shutterstock stock videos dataset[1], which contains 10M unique captions. Retaining one random clip per caption yields 10M video–caption pairs, equivalent to the WebVid-10M dataset [4] previously used in academic research.



Original Caption

Mumbai/India 2.21.2016 video of Driving On Mumbai Roads taken by point of view camera

High Quality `Claude-3-Sonnet` Generated Caption

This video appears to be a first-person perspective drive through a quiet urban street in India during early morning hours. The camera moves along a wide, tree-lined road flanked by impressive multi-story buildings with ornate architecture. As the journey progresses, the sun rises, casting a warm glow on the buildings and empty street. Few vehicles and pedestrians are seen, creating a serene atmosphere as the city slowly awakens.

Figure 4: Comparing original caption against `Claude-3-Sonnet` generated high-quality caption.

4.2 Stage 2: High Quality Intermediate Data

We use a re-captioned subset of the Shutterstock dataset, whose extensively labeled keywords we utilize for the balancing operation. We exclude keywords with < 30 occurrences. Starting with the most frequent remaining keywords, we add a maximum of 500 videos per keyword to the candidate pool. Further details are in Supplementary L. In order to generate this data within a limited budget, we generate captions for a diverse subset of 200,000 videos. We follow the insight from prior works (e.g., LLaVA [26], Meta-CLIP [48], Cambrian [40]) that under-sampling common concepts is important for large scale vision-language training to achieve better generalization on rare concepts.

We use the `Claude-3-Sonnet` model prompted with 20 frames from the video, each resized to 480 pixels height, along with the following text prompt: *"Provide detailed narrative caption for the video, covering its overall theme, visuals, characters, actions, scene, and key moments comprehensively"*. The original caption and generated high quality caption for a sample video are shown in Fig. 4.

4.3 Stage 3: Multi-Task Fine-Tuning Data

Our fine-tuning stage is analogous to Instruction FineTuning (IFT) in LLMs. We design our fine-tuning dataset to include multi-task supervision covering these tasks as well as video captioning. We select 100,000 samples from a small number of video-text datasets and obtain supervision from each as summarized in Table 2. All these datasets contribute to captioning, retrieval and matching supervision. **MSR-VTT** provides short video captions and question answering supervision. **ActivityNet** and **DiDeMo** consist of longer videos with distinct temporally localizable captions for chunks of the video. Hence, these datasets can provide localization supervision. **ActivityNet** has also previously been annotated with question answering data. We add a random subset of Shutterstock video-caption pairs (original captions) to the mix to maintain long tail diversity.

Dataset	#	Cap	QA	Loc
MSR-VTT	10K	✓	✓	✗
ActivityNet	30K	✓	✓	✓
DiDeMo	10K	✓	✗	✓
Shutterstock	50K	✓	✗	✗

Table 2: Multi-Task Dataset.

5 Results

First we demonstrate the unified capabilities of our model on 8 benchmark datasets. Next up, we demonstrate the unique capabilities of our model by solving novel tasks that are hard for any single model prior to this work. Finally, we show ablations demonstrating the effectiveness of each component in our modeling and training approach.

5.1 Retrieval Tasks

We evaluate our model on video retrieval, and moment retrieval/temporal localization tasks. These benchmarks together provide a holistic evaluation of video understanding capabilities. We use standard evaluation protocols following prior works [36, 37, 24].

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

Method	Text to Moment Retrieval			Text to Video Retrieval		
	ActivityNet R@1, IoU=0.5	Charades R@1, IoU=0.5	QVHighlights mAP	MSR-VTT R@1	DiDeMo R@1	VATEX R@1
Expert Models						
M-DETR NeurIPS 21	-	53.6[FT]	35.7[FT]	✗	✗	✗
QD-DETR CVPR 23	33.2[FT]	57.3[FT]	38.9[FT]	✗	✗	✗
TR-DETR AAAI 24	-	57.6[FT]	39.9[FT]	✗	✗	✗
SigLIP ICCV 23	✗	✗	✗	51.7[FT]	55.4[FT]	40.8
VidLA CVPR 24	✗	✗	✗	58.0[FT]	61.1[FT]	-
InternVideo2-CLIP-1B ECCV 24	✗	✗	✗	50.0	47.7	63.2
Video LLMs						
Momentor-7B ICML 24	23.0	27.5	7.6	✗	✗	✗
VTimeLLM-7B CVPR 24	27.8	27.5	-	✗	✗	✗
TimeChat-7B CVPR 24	16.3	32.2	21.7	✗	✗	✗
Qwen2.5-VL-3B arXiv	28.6	38.1	-	✗	✗	✗
LLaVA-ST-7B CVPR 25	31.2	44.8	-	✗	✗	✗
ViLL-E-2.5B (Ours)	**39.4**↑8.2	**51.5**↑6.7	39.0	**62.5**↑4.5	**61.4**	**63.5**

Table 1: **Results on retrieval tasks**. In moment retrieval, we outperform specialized Video LLMs by over 8%, and approach the performance of SotA dataset-specific expert models. In video retrieval we beat finetuned SotA models. [FT] models fine-tuned on train splits. ✗- model is incapable of task

Moment Retrieval: We test on ActivityNet-Captions, Charades-STA and QVHighlights. Contextualized text embeddings are extracted by passing `{sparse video frames} <|txt_embed|> {text}`. Video clips are extracted using a sliding window operation and clip embeddings are generated by passing `{clip frames} <|vid_embed|>`. Text and Video Clip embedding similarities are post-processed using standard methods to get the matching segment.

Video Retrieval: We evaluate on MSR-VTT, DiDeMo and VATEX. Video embeddings are extracted by passing following input to the model: `{video} <|vid_embed|>`, while text embeddings are extracted by passing `<|txt_embed|> {text}`. Best pair is matched using nearest neighbor retrieval.

In moment retrieval (Table 1), ViLL-E outperforms localization specialized video-LLMs by nearly 10%, and approaches or surpasses the per-dataset fine-tuned performance of non-LLM methods. In text-video retrieval, ViLL-E matches image (SigLIP) and video (VidLA) retrieval models.

5.2 QA Tasks

Among VideoLLMs specialized for temporal localization, our method achieves top performance in several categories (MSR, VCG, MVBench) indicated by green upward arrows showing improvements over previous methods. It slightly underperforms compared to LLaVA-ST on the MSVD dataset, indicated by a small red downward arrow. While SotA general VideoLLMs Qwen-2.5VL and LLaVA-Video outperform ViLL-E on MVBench & VideoMME, our approach remains competitive.

5.3 Additional Tasks

In this section we demonstrate ViLL-E's ability to solve some unique tasks that are made possible by having unified generative and embedding capabilities in one VideoLLM.

Method	MSR	VCG	MSVD	MVBench	VidMME
General Video LLMs					
Qwen-2.5VL	-	-	-	67.0	61.5
LLaVA-Video	-	3.5	-	58.6	63.3
Temporally-Specialized Video LLMs					
ST-LLM	63.2	3.15	74.6	54.9	37.9
LLaVA-ST	59.0	3.3	**75.9**	64.2	
Momentor	55.6	3.0	68.9	-	-
VTimeLLM	50.2	2.9	-	-	-
TimeChat	45.0	*2.3*	-	38.5	34.7
ViLL-E	**65.2**	**3.7**	75.2	**64.7**	45.0

Table 3: ViLL-E is competitive with SotA VideoLLMs on QA tasks in both simple VideoQA (MSR-VTT QA, VideoChatGPT Benchmark and MSVD-QA) and Comprehensive Benchmarks (MVBench and VideoMME).

Two Stage Retrieval using Re-Ranking: This task leverages the VideoLLM's reasoning capability to augment retrieval performance. In the first stage, Top-K candidate videos matching a caption are retrieved using generated embeddings. During second stage matching, each candidate video-caption pair is input to the VideoLLM and the matching loss is used to re-rank the Top-K candidate videos. The results in Table 4 ($K = 25$) show that our two stage matching process results in an improvement of about 2% in R@1 accuracy.

MSR-VTT	T → V		V → T	
Method	R@1	R@5	R@1	R@5
VidLA	58.0	**81.1**	56.1	80.5
Ours (1 Stage)	62.5	78.1	55.3	74.8
Ours (2 Stage)	**62.8**	80.1	**57.3**	**83.5**

Table 4: Two-stage retrieval + matching.

Composed Video Retrieval [42] is the task of retrieving a video matching a given query video along with a modifier text. The retrieved video

must closely match a hypothetical video containing the contents of the original video further modified by the instructions in the change text. This is a relatively new task in the field of video retrieval and forms a challenging zero-shot benchmark for our model. To solve this, we extract multi-modal query embeddings by passing `{source video frames} <|txt_embed|> {change text}`, and retrieve videos by matching with target video embeddings extracted as we do for retrieval. The results in Table 5 demonstrate that in the zero-shot setting our method surpasses recent baselines by more than 5%.

Method	R@1	R@5	R@10
COVR-BLIP AAAI 24	45.46	70.46	79.54
Thawakar et al. CVPR 24	47.52	72.18	82.37
Ours	**53.13**	**74.80**	**85.95**

Table 5: Zero-Shot Composed Video Retrieval.

Detailed Video Caption Retrieval: Dual encoder video-text retrieval models, like CLIP, are limited by short context windows (e.g., 77 tokens). Our LLM approach supports much longer texts:

Method	Short Caption **Avg.** = 27 words				Long Caption **Avg.** = 500 words			
	T → V		V → T		T → V		V → T	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
SigLIP ICCV23	62.5	83.2	65.4	85.1	51.7	72.7	60.4	81.0
VidLA CVPR24	61.2	**85.7**	**65.9**	85.0	45.3	65.2	55.9	72.7
LongCLIP ECCV24	63.0	85.4	61.5	83.6	73.5	91.7	74.5	92.8
ViLL-E	**64.3**	**85.7**	65.5	**85.4**	**75.7**	**92.4**	**75.1**	**93.3**

Table 6: Zero-Shot Video Detailed Caption Retrieval.

thousands of tokens, enabling effective zero-shot retrieval using detailed captions. Existing benchmarks (e.g., YouCook2) are either too domain-specific or overlap with our training data. We use the AuroraCap-VideoDetailCaption dataset [2], which includes both long and short captions for 1,027 videos. As shown in Table 6, our model benefits from longer captions, improving R@1 by 11.4% (text-to-video) and 9.6% (video-to-text). In contrast, models like SigLIP [53] and VidLA [37] are constrained by short text encoder contexts (64 tokens). LongCLIP [54], with a 248-token limit, improves on long captions but lags our approach.

5.4 Ablations

We ablate the design choices for key components of our approach. For computational efficiency, some experiments are performed without pre-training.

Supervision Types: (Table 7a) We find that combining generative and contrastive objectives during fine-tuning complements each other, resulting in improvement on both generative and embeddings tasks. Retrieval and localization performances drop moderately on removing matching loss, and significantly on removing contrastive training, indicating the importance of learning good embeddings.

G	C	M	MSR	VCG	MSR	DDM	VTX	ANet	QVH	Ch.
✓	✓	✓	65.1	3.7	62.8	61.5	63.7	39.4	38.9	51.7
✓	✓	×	63.9	3.7	60.3	60.2	62.2	39.1	38.5	51.5
✓	×	×	61.3	3.0	25.1	23.5	11.9	28.7	23.8	42.1
×	✓	×	45.5	2.1	54.7	53.1	55.2	29.3	30.3	48.8

(a) Supervision Type (During Finetuning)

Head	MSR	VCG	MSR	DDM	VTX	ANet	QVH	Ch.
Ours	55.9	3.1	49.3	45.5	45.3	32.3	32.6	49.3
×	40.8	2.8	30.7	29.1	32.1	19.8	20.4	30.7
Linear	42.9	2.8	32.4	29.3	33.2	23.6	25.4	42.7
MLP	51.7	3.0	43.5	39.8	39.5	30.2	29.7	47.1
Self-Att	52.5	3.2	43.8	40.9	40.9		28.2	45.7
Q-Former	52.1	3.0	47.5	43.2	42.8		27.9	44.2
K-Former	52.7	3.1	49.0	44.1	44.7		32.8	47.4

(b) Embedding Head Design

Pre	HQ	MSR	VCG	MSR	DDM	VTX	ANet	QVH	Ch.
✓	✓	65.1	3.7	62.8	61.5	63.7	39.4	38.9	51.7
✓	×	63.2	3.3	58.6	59.6	60.5	36.8	38.8	51.9
×	×	55.9	3.1	49.3	45.5	45.3	32.3	32.6	49.3

(c) Impact of Pre-Training

Table 7: **Ablation Experiments.**
Benchmarks (left to right). QA: MSR-VTT-QA, VideoChatGPT. **Retrieval**: MSR-VTT, DiDeMo, VATEX. **Localization**: ActivityNet, QVHighlights, Charades-STA.
Legend: (a) G = Generative, C = Contrastive, M = Matching. **(c)** *Pre.* = 10M scale; *HQ* = High-Quality 200k.

Embedding Head (Table 7b) We find that our attentive pooling head outperforms other potential approaches discussed previously in Section 3.2.

Pre-Training (Table 7c) The large scale pre-training stage is very important for retrieval tasks, whereas the second pre-training stage with longer higher quality captions helps particularly in longer video datasets and generative tasks.

Additional experiments and analysis of our adaptive embedding mechanism: including inference latency, per-duration-bin retrieval accuracy, and distribution of generated token counts, are provided in Appendix A (Tables 10 and 11).

6 Conclusion

We introduced ViLL-E, a unified VideoLLM that combines text generation and embedding-based retrieval within one model. Through a multi-stage generative–contrastive training strategy and an adaptive embedding head, ViLL-E achieves strong results across retrieval, localization, and QA, while enabling new zero-shot tasks like composed and long-caption retrieval. This work highlights the potential of merging generative and discriminative learning for unified multimodal understanding.

7 Limitations

Our model inherits some limitations of the base Paligemma LLM, e.g. it lacks general multi-turn conversation abilities. Secondly, because our training dataset is primarily in English, we expect that our model would lose some of Paligemma's multilingual capabilities. Our work serves as a proof of concept for the unification of generative and embedding capabilties in VideoLLMs, and any practical model would need to address the issues of multi-turn conversations and multi-linguality.

References

[1] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pages 5803–5812.

[2] Anonymous. 2024. Auroracap: Efficient, performant video detailed captioning and a new benchmark. In *Submitted to The Thirteenth International Conference on Learning Representations*. Under review.

[3] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

[4] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. 2021. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *IEEE International Conference on Computer Vision*.

[5] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. 2021. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4.

[6] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, and 1 others. 2024. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*.

[7] Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

[8] Dongsheng Chen, Chaofan Tao, Lu Hou, Lifeng Shang, Xin Jiang, and Qun Liu. 2022. Litevl: Efficient video-language learning with enhanced spatial-temporal modeling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7985–7997.

[9] Feng Cheng, Xizi Wang, Jie Lei, David Crandall, Mohit Bansal, and Gedas Bertasius. 2023. Vindlu: A recipe for effective video-and-language pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10739–10750.

[10] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, pages 5267–5275.

[11] Satya Krishna Gorti, Noël Vouitsis, Junwei Ma, Keyvan Golestan, Maksims Volkovs, Animesh Garg, and Guangwei Yu. 2022. X-pool: Cross-modal language-video attention for text-video retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5006–5015.

[12] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. 2024. Vtimellm: Empower llm to grasp video moments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14271–14280.

[13] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

[14] Ziyan Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. 2025. VLM2vec: Training vision-language models for massive multimodal embedding tasks. In *The Thirteenth International Conference on Learning Representations*.

[15] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715.

[16] Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*.

[17] Jie Lei, Tamara L Berg, and Mohit Bansal. 2021. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34:11846–11858.

[18] Hongyu Li, Jinyu Chen, Ziyu Wei, Shaofei Huang, Tianrui Hui, Jialin Gao, Xiaoming Wei, and Si Liu. 2025. Llava-st: A multimodal large language model for fine-grained spatial-temporal understanding. *arXiv preprint arXiv:2501.08282*.

[19] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

[20] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR.

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

[22] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, and 1 others. 2024. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206.

[23] Kunchang Li, Yali Wang, Yizhuo Li, Yi Wang, Yinan He, Limin Wang, and Yu Qiao. 2023. Unmasked teacher: Towards training-efficient video foundation models. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19891–19903.

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

[25] Sheng-Chieh Lin, Chankyu Lee, Mohammad Shoeybi, Jimmy Lin, Bryan Catanzaro, and Wei Ping. 2024. Mm-embed: Universal multimodal retrieval with multimodal llms. *arXiv preprint arXiv:2411.02571*.

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

[27] Xiaolong Liu, Qimeng Wang, Yao Hu, Xu Tang, Shiwei Zhang, Song Bai, and Xiang Bai. 2022. End-to-end temporal action detection with transformer. *IEEE Transactions on Image Processing*, 31:5427–5441.

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

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

[30] WonJun Moon, Sangeek Hyun, SangUk Park, Dongchan Park, and Jae-Pil Heo. 2023. Query-dependent video representation for moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23023–23033.

[31] Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2024. Generative representational instruction tuning. *arXiv preprint arXiv:2402.09906*.

[32] Long Qian, Juncheng Li, Yu Wu, Yaobo Ye, Hao Fei, Tat-Seng Chua, Yueting Zhuang, and Siliang Tang. 2024. Momentor: Advancing video large language model with fine-grained temporal reasoning. In *Forty-first International Conference on Machine Learning*.

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

[34] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

[35] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3980–3990. Association for Computational Linguistics.

[36] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. 2024. Timechat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14313–14323.

[37] Mamshad Nayeem Rizve, Fan Fei, Jayakrishnan Unnikrishnan, Son Tran, Benjamin Z Yao, Belinda Zeng, Mubarak Shah, and Trishul Chilimbi. 2024. Vidla: Video-language alignment at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14043–14055.

[38] Paul K Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, and 1 others. 2023. Audiopalm: A large language model that can speak and listen. *arXiv preprint arXiv:2306.12925*.

[39] Hao Sun, Mingyao Zhou, Wenjing Chen, and Wei Xie. 2024. Tr-detr: Task-reciprocal transformer for joint moment retrieval and highlight detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4998–5007.

[40] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. 2024. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Preprint*, arXiv:2406.16860.

[41] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

[42] Lucas Ventura, Antoine Yang, Cordelia Schmid, and Gül Varol. 2024. Covr: Learning composed video retrieval from web video captions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5270–5279.

[43] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

[44] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. 2019. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4581–4591.

[45] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, and 1 others. 2024. Internvideo2: Scaling foundation models for multimodal video understanding. In *European Conference on Computer Vision*, pages 396–416. Springer.

[46] Yueqian Wang, Xiaojun Meng, Jianxin Liang, Yuxuan Wang, Qun Liu, and Dongyan Zhao. 2024. Hawkeye: Training video-text llms for grounding text in videos. *arXiv preprint arXiv:2403.10228*.

[47] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. 2017. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM*

International Conference on Multimedia, MM '17, page 1645–1653, New York, NY, USA. Association for Computing Machinery.

[48] Hu Xu, Saining Xie, Xiaoqing Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. 2024. Demystifying CLIP data. In *The Twelfth International Conference on Learning Representations*.

[49] 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.

[50] Hongwei Xue, Yuchong Sun, Bei Liu, Jianlong Fu, Ruihua Song, Houqiang Li, and Jiebo Luo. 2023. Clip-vip: Adapting pre-trained image-text model to video-language alignment. In *The Eleventh International Conference on Learning Representations*.

[51] Shen Yan, Xuehan Xiong, Arsha Nagrani, Anurag Arnab, Zhonghao Wang, Weina Ge, David Ross, and Cordelia Schmid. 2023. Unloc: A unified framework for video localization tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13623–13633.

[52] Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. 2023. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10714–10726.

[53] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986.

[54] Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. 2025. Long-clip: Unlocking the long-text capability of clip. In *Computer Vision – ECCV 2024*, pages 310–325, Cham. Springer Nature Switzerland.

[55] Chen-Lin Zhang, Jianxin Wu, and Yin Li. 2022. Actionformer: Localizing moments of actions with transformers. In *European Conference on Computer Vision (ECCV)*.

[56] Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. 2023. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities. *arXiv preprint arXiv:2305.11000*.

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

[58] Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. 2024. Gme: Improving universal multimodal retrieval by multimodal llms. *arXiv preprint arXiv:2412.16855*.

[59] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, WANG HongFa, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, Cai Wan Zhang, Zhifeng Li, Wei Liu, and Li Yuan. 2024. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. In *The Twelfth International Conference on Learning Representations*.

Overview of Supplementary Material

A Additional Ablations

Number of Pooling Tokens (Table 8) We find that in practice, a relatively high value of (we use 256) is optimal (compare to Q-Former, where usually 32 queries are used). Going beyond 256 to higher values, gains seem minimal.

#	Pre.	VCG	MSR	VTX	QVH
Ours (256)	×	3.1	49.3	45.3	32.6
32	×	2.7	34.8	35.5	25.7
64	×	3.0	42.5	39.2	29.5
128	×	3.1	48.6	44.9	32.0
512	×	3.1	49.5	45.0	32.4

Table 8: Number of Pooling Tokens

Number of Embedding Tokens (Table 9) Our video embedding generation outputs a variable number of tokens (until <EOS>) to send to the embedding head. Whereas prior encoder approaches (e.g. GRIT [31]) generate a fixed number of tokens, to match this setting and measure the benefit of our dynamic approach we also train the model to use just 1 (or 5) token for embedding generation.

#	Pre.	MSR VCG	MSR DDM	VTX	ANet	QVH	Ch.
Ours	×	55.9 3.1	49.3 45.5	45.3	32.3	32.6	49.3
1	×	50.2 2.6	45.8 37.6	38.9	23.7	24.1	36.8
5	×	54.1 2.8	47.9 40.3	41.2	27.3	27.8	42.3

Table 9: Number of Embedding Tokens

Inference Latency vs. Number of Embedding Tokens (Table 10) We measure inference latency on an AWS `g6e.8xlarge` instance equipped with an NVIDIA L40S GPU. Our adaptive method adds only marginal overhead compared to a fixed 5-token baseline, while achieving substantially better accuracy (Table 9).

Tokens	**Latency (ms)**
Fixed – 1	238
Fixed – 5	257
Fixed – 10	326
Adaptive (Ours)	262

Table 10: Inference latency vs. number of embedding tokens for 12-frames (batch size 16) on NVIDIA L40S.

Adaptive Tokens Across Video Durations (Table 11) To further analyze the adaptive token mechanism, we break down the ActivityNet Captions dataset by video duration. This dataset has a wide and nearly uniform distribution of videos up to 5 minutes, providing balanced bins. Our adaptive method generalizes better to variable video lengths, significantly outperforming fixed-token approaches across all duration bins.

Method	**0–60s**	**60–120s**	**120–180s**	**180s+**	**Overall**
VidLA Baseline	65.7	66.5	65.0	63.5	65.2
Fixed Token = 1	66.2	64.7	63.5	62.2	64.1
Fixed Tokens = 5	66.6	67.4	66.4	65.4	66.5
Adaptive (Ours)	**67.9**	**68.5**	**67.8**	**67.1**	**67.9**

Table 11: ActivityNet Captions Text-to-Video Retrieval (R@1) broken down by video duration.

To verify that the model actually leverages variable-length generation, we measure the average number of tokens produced by the adaptive method per duration bin: **0–60s**: 8.6 tokens; **60–120s**: 10.1 tokens; **120–180s**: 11.5 tokens; **180s+**: 14.8 tokens. This confirms that the model naturally "thinks longer" for more complex, longer videos.

B Shared Notation

For clarity we reserve a single symbol for each recurring concept. An *entire video* is denoted by a capital letter, V, whereas a *temporal segment* cut from that video is represented in lowercase, v. Hard-negative segments for localization are denoted by v_{ij}^n, where j indexes the negatives for

sample i. Any natural-language sequence—be it a caption, a question, or an answer—is denoted by t; its k-th token is w_k, and the total length is L_{cap} for captions or L_{ans} for answers. The LLM maps raw inputs to embeddings, producing $\mathbf{e}_V = f(V, < m >)$, $\mathbf{e}_v = f(v, < m >)$ and $\mathbf{e}_t = f(t, < m >)$, $f(.)$ representing our model and $< m >$ represents the mode token, e.g. `|VID_EMB|` is the video embedding token. All contrastive objectives use the same similarity function $\text{sim}(\cdot, \cdot)$ (cosine by default) scaled by a temperature parameter τ. We use a mini-batch of size N.

Text–Video Retrieval (Fig. 3a). This task aligns video and text embeddings using CLIP-like [33] contrastive loss.

$$\mathcal{L}_{\text{ret}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp\big(\text{sim}(\mathbf{e}_V^i, \mathbf{e}_t^i)/\tau\big)}{\sum_{j=1}^{N}\exp\big(\text{sim}(\mathbf{e}_V^i, \mathbf{e}_t^j)/\tau\big)}.$$

Video Captioning & Question–Answering (Fig. 3b). The goal of captioning is for the model to generate accurate descriptions of videos. The model learns to predict each token in the caption, given the previous tokens and the video. Whereas for QA, the model generates answers to questions about the video. The question serves as part of the prompt, guiding the model's attention to answer accurately.The next-token prediction loss is used.

$$\mathcal{L}_{\text{cap}} = -\sum_{k=1}^{L_{\text{cap}}}\log p\big(w_k^{\text{cap}} \mid w_{<k}^{\text{cap}}, V\big),$$

$$\mathcal{L}_{\text{qa}} = -\sum_{k=1}^{L_{\text{ans}}}\log p\big(w_k^{\text{ans}} \mid w_{<k}^{\text{ans}}, V, t_{\text{ques}}\big).$$

Text–Video Matching (Fig. 3c). As a complement to the retrieval task we also formulate a matching task where the model is trained to identify if a given video and caption form a matching pair. Binary decisions are supervised by the label $y \in \{0, 1\}$.

$$\mathcal{L}_{\text{m}} = -\big(y\log p(\text{Y} \mid V, t) + (1-y)\log p(\text{N} \mid V, t)\big).$$

Temporal Localization (Fig. 3d). Localization requires the model to match specific video segments to text annotations while distinguishing similar but incorrect segments. Sliding window hard negative mining provides challenging non-matching segments with limited temporal overlap (IoU < 0.2), helping the model learn to differentiate closely related content. Let $s_i^+ \triangleq \text{sim}(\mathbf{e}_v^i, \mathbf{e}_t^i)$ (positive) and $s_{ij}^- \triangleq \text{sim}(\mathbf{e}_v^{i,j}, \mathbf{e}_t^i)$ (j-th hard negative for sample i, $j \in \mathcal{N}_i$; IoU < 0.2). The localization loss, $\mathcal{L}_{\text{l}}$, can be written as

$$\mathcal{L}_{\text{l}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp\big(s_i^+/\tau\big)}{\exp\big(s_i^+/\tau\big) + \sum_{j\in\mathcal{N}_i}\exp\big(s_{ij}^-/\tau\big)}. \quad (1)$$

C Implementation Details and Hyperparameters

We use PaliGemma as our primary MLLM backbone. PaliGemma is openly licensed (Apache 2.0) and is based on the Gemma-2B language model and uses SigLIP-SO-400M ViT as its vision encoder. The single layer embedding head has the same dimensions as a Gemma-2B layer (`hidden_dim` = 2048, `mlp_dim` = 16384, and `n_heads` = 8). All video frames are resized to 224 × 224 before feeding into the model. LoRA adapters of rank 128 were used for finetuning and are applied to all linear layers of the LLM transformer (i.e., excluding the vision encoder, embedding layers, and output heads). PaliGemma's `<unused0>` and `<unused1>` tokens are leveraged to initialize `<txt_embed>` and `<video_embed>`. A modified Huggingface `Trainer` is used for training, with `FSDP` mode used across GPUs to reduce $VRAM$ usage. Training hyperparameters for each stage are provided in Table 12. The key reason behind these choices is improving the efficiency of the large scale Stage 1 training. The learning rates are chosen on the basis of small training runs on the MSR-VTT dataset. Inference latency is benchmarked on an AWS `g6e.8xlarge` instance with an NVIDIA L40S GPU. For temporal localization, we use a sliding window of 10 seconds with a stride of 5 seconds (see Appendix G).

Multi-Task Finetuning Batch Creation: Following Table 2, each video in our dataset has 4 boolean flags indicating which training tasks are supported for the video. When a random batch of videos is sampled, the flags are sampled with them. The flags are used to filter a partial task specific minibatch, with one minibatch per task. Gradients are accumulated across all training tasks before updating the model weights.

Hyperparameter	**Stage 1/2**	**Stage 3**
Number of GPUs	32	8
Batch Size (per GPU)	12	8
Gradient Accumulation	2	4
Optimizer	Lion	AdamW
Base Learning Rate	1e-5	5e-6
Warmup Steps	2000	1000
Weight Decay*	0.00001	-
LoRA Rank	128	32
Training Steps	25,000	4,000
Max Token Length	2304/4096	4096

Table 12: Training hyperparameters across different stages. *- only applied to new embedding head.

D Broader Impacts

A key finding of our work is the ability to unify generative and embedding models, which could potentially reduce the amount of energy used and environmental impact from maintaining two different model pipelines for inference in practical applications. In terms of potential negative impact, our model is very general, and hence shares some of the potential for misuse inherent to generalist computer vision models.

E Privacy Safeguards

All videos are stored and used with the faces of humans blurred using the `deface` python package.

F Licenses

WebVid-10M (Shutterstock clips). A proprietary corpus governed by Shutterstock's Terms of Service; URLs and captions may be downloaded for *internal, non-commercial research* only, and redistribution is forbidden.

MSR-VTT. Released by Microsoft Research solely for academic research; no explicit open-source license is attached, so usage is restricted to the terms on the dataset website/publication.

ActivityNet. Distributed under the MIT License, permitting commercial and non-commercial use provided that copyright and license notices are retained.

DiDeMo. Available under the BSD 2-Clause license, a permissive license that allows modification and redistribution with minimal obligations.

VATEX. Shared under Creative Commons Attribution 4.0 International (CC BY 4.0); free use with attribution.

QVHighlights. Released with a Creative Commons Attribution–NonCommercial–ShareAlike 4.0 license (CC BY-NC-SA 4.0); commercial use is prohibited and derivatives must adopt the same license.

Charades & Charades-STA. Allen AI provides these datasets under a *non-commercial research* license; any for-profit use requires separate permission.

SigLIP-SO-400M visual encoder. Model weights and code are released under the Apache License 2.0.

PaliGemma-3B multimodal LLM. Weights are covered by Google's *Gemma Open Model License 1.0*; accompanying reference code is Apache 2.0.

Claude-3 Sonnet (caption generator). A proprietary model whose use is governed by Anthropic's Terms of Service.

G Localization Inference

We apply post-processing inspired from prior embedding-based localization approaches to merge segments.

We split the video into non-overlapping windows and compute a text-video similarity score for each segment. We take the highest-scoring window (with score s_{seed}) as the seed, then merge adjacent windows to the left and right as long as each neighbor's score exceeds a merge threshold (τ_{merge}) or is at least a fixed ratio (i.e., α) of the seed's score. Formally, a neighboring window i is merged if

$$s_i \geq \tau_{\text{merge}} \quad \text{or} \quad s_i \geq \alpha \cdot s_{\text{seed}}.$$

Expansion stops at the first neighbor on either side that fails both conditions. The final prediction is the union of all accepted windows. Center correction is applied to the start and end clips if they are not the seed clip.

Some prior works that we referred to in order to develop our post-processing strategy include [55, 27].

We ablate some of the choices related to our localization inference protocol here:

Window	Stride	IoU	R@1, IoU=0.5
5s	5s	0.3	39.0
5s	5s	0.4	39.4
5s	5s	0.5	35.7
10s	10s	0.4	39.2
10s	**5s**	**0.4**	**41.7**
15s	10s	0.4	39.9

Table 13: Ablation of localization inference protocol choices. Score on ActivityNet Dataset.

H Matching Based Two Stage Retrieval

We provide more details of how we enhance video-caption retrieval performance by combining retrieval techniques with advanced matching capabilities of a VideoLLM. Below, we describe the steps for each stage:

Stage 1: Embedding-Based Retrieval

1. **Embedding Generation:**
    - Text embeddings for the input caption are generated using ViLL-E in text embedding mode.

- Video embeddings for all videos in the database are pre-computed using ViLL-E in video embedding mode.

2. **Similarity Computation:**
   - The similarity between the caption embedding and each video embedding is computed, typically using cosine similarity.
3. **Top-K Retrieval:**
   - The K most similar video candidates are selected based on the similarity scores. This reduces the search space for the more computationally intensive second stage.

Stage 2: Matching-Based Re-Ranking

1. **Video-Caption Pair Processing:**
   - Each video in the Top-K set is paired with the input caption and fed into the VideoLLM.
   - A carefully designed prompt is used to guide the VideoLLM in determining whether the pair is a match.
2. **Prediction and Scoring:**
   - For each video-caption pair, the VideoLLM predicts a likelihood score for both `Yes` (match) and `No` (no match) tokens.
   - A matching score is computed from the logits of these predictions (e.g., softmax probabilities or logit differences).
3. **Re-ranking:**
   - The Top-K ($K = 20$) videos are re-ranked based on their matching scores, with higher scores indicating stronger matches.

Benefits of the Two-Stage Approach

- **Efficiency:** By leveraging computationally efficient embedding-based retrieval in the first stage, the system avoids evaluating all database videos
- **Accuracy:** The VideoLLM's reasoning capabilities in the second stage allow it to resolve ambiguities and refine rankings, leading to a significant improvement in retrieval accuracy (e.g., $\sim 2\%$ increase in `R@1` accuracy).
- **Scalability:** The division into two stages ensures that the system can scale to large databases without compromising on retrieval quality.

Prompt for Matching

The prompt provided to the VideoLLM for second-stage matching is structured as follows:

- **Input:** `<Video Tokens>` Caption: `[Retrieved first stage caption]`. Question: Does the above video match the caption? Answer "Yes" or "No".
- **Output:**
  - Token logits for `Yes` and `No`, which are further used to compute the matching score.

I Composed Video Retrieval

As illustrated in Fig. 6, inference in the composed-video-retrieval (CoVR) pipeline begins by encoding the source clip with a spatio-temporal vision encoder whose output tokens are linearly projected into the hidden dimension of the downstream large-language-model (LLM) transformer. A single text-edit token, `|TXT_EMB|`, embeds the user's natural-language instruction and is concatenated to the projected visual tokens, enabling early fusion of visual context and linguistic modification. The resulting sequence is processed by several frozen LLM transformer layers, whose final hidden states are pooled to yield a compact query representation that jointly reflects the appearance of the source clip and the requested transformation. In parallel, every clip in the retrieval gallery is independently passed through the same vision-encoder $\rightarrow$ projection $\rightarrow$ LLM steps—without the text token—to obtain target embeddings via a lightweight embedding head. At retrieval time, cosine (or dot-product) similarity is computed between the composed query embedding and all target embeddings, and the gallery video with maximal similarity is returned as the predicted target.

J Detailed Overview of Stage 1 Pretraining

During Stage 1 joint pre-training (Fig. 9) the model is optimized with both a causal-language-model (LM) objective and a video–text contrastive objective, each applied in a separate forward pass that nevertheless shares parameters. First, a batch

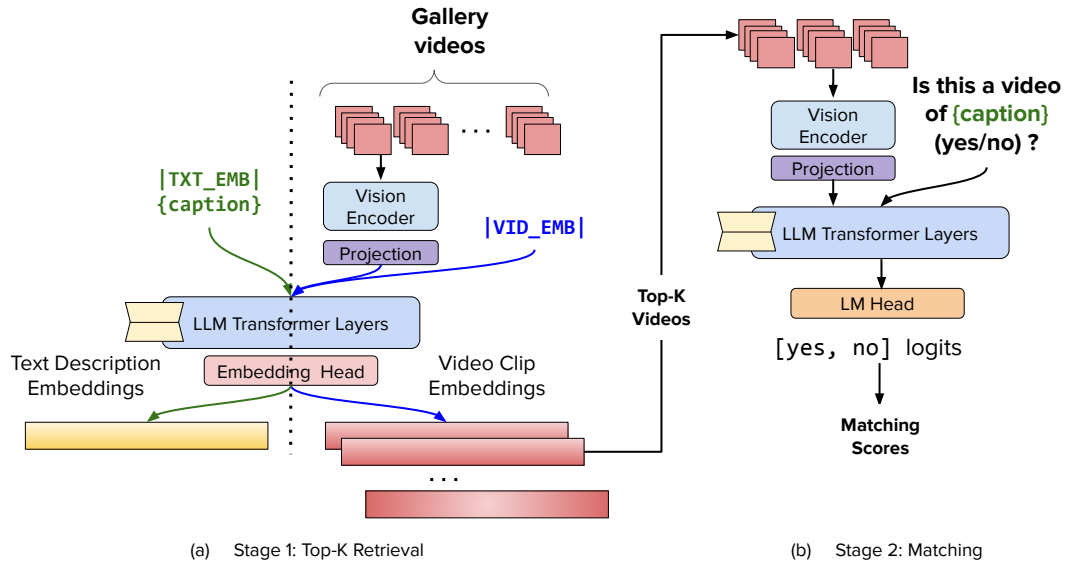


Figure 5: Two Step Retrieval Inference.

of videos is fed through a frozen spatio-temporal vision encoder; its output tokens are linearly projected into the hidden dimension of a large language model (LLM) that has been lightweight-fine-tuned via LoRA adapters (modules marked with fire symbol). These visual tokens are prepended to the textual prompt "<image> ... <image> Describe this video." and processed by the LLM. Two heads branch from the final hidden states: (i) a language-modeling head generates a caption autoregressively, whose cross-entropy with the ground-truth caption constitutes the language-modeling loss; and (ii) an embedding head pools the hidden states to yield a fixed-length video embedding.

In the second forward pass, only caption strings are supplied—again through the same LLM + LoRA stack, ensuring weight sharing between modalities—and their hidden states are mapped by the embedding head to caption embeddings. The embedding corresponding to the correct caption serves as the positive partner for the video embedding, while captions of the remaining videos in the batch act as negatives. A standard InfoNCE objective then enforces contrastive alignment between the video embedding and its positive caption while repelling mismatched pairs.

The two losses are summed, so the model simultaneously learns to generate fluent video descriptions and to discriminate between semantically consistent versus inconsistent video–text pairs, yielding representations that are effective for both captioning and downstream retrieval tasks.

K Detailed Overview of Stage 3: Temporal localization training

As depicted in Fig. 10, Stage 3 (Temporal Localization Task) fine-tunes the model for temporal localization by contrasting text queries with candidate video clips extracted from long-form footage**. Given an annotated clip-level caption (e.g., "playing with glow slime in the dark"), a small set of sparsely sampled frames from the entire video is first prepended to the caption prompt to furnish coarse visual context, after which the sequence is routed through the LoRA-adapted LLM and pooled by the embedding head to yield a contextualized text embedding. In parallel, the full video is partitioned with a fixed-stride sliding window; each windowed segment is passed through the same vision-encoder → projection → LLM → embedding-head stack to obtain video-clip embeddings. The clip whose temporal bounds coincide with the ground-truth annotation is designated the *positive* match, while all other windowed segments within the batch serve as *hard negatives*. A contrastive InfoNCE loss is then applied between the text embedding and the set of clip embeddings, driving the model to maximise similarity with the positive segment and to minimise similarity with temporally misaligned negatives. This procedure equips the shared embedding space with fine-grained temporal sensitivity, enabling the model to accurately localise textual events within extended, untrimmed videos without requiring explicit frame-level supervision.

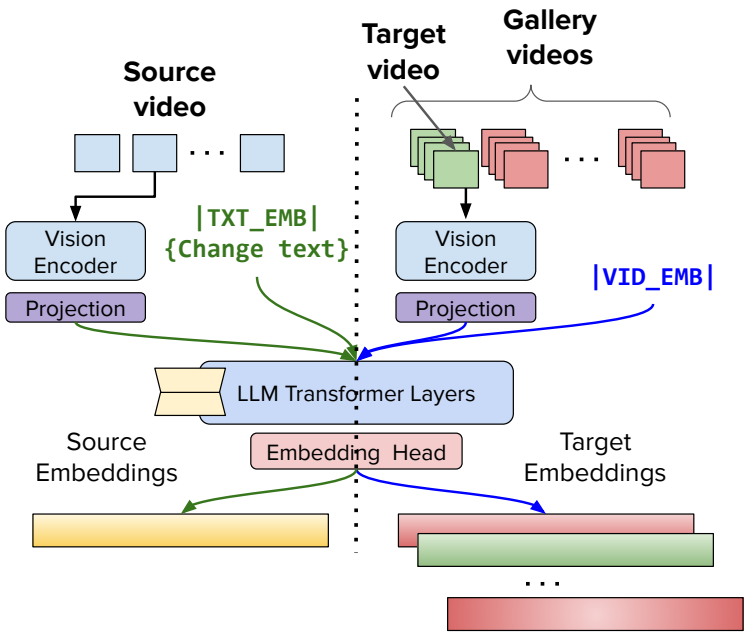


Figure 6: Composed Video Retrieval Inference.

L Intermediate Pre-training Data Balancing

The Shutterstock dataset is extensively labelled with keywords, which we use to select a balanced set of videos for our intermediate pre-training dataset. Keywords with fewer than 30 occurrences are excluded, and up to 500 videos per remaining keyword are added to the candidate pool, starting with the most frequent keywords, to ensure a balanced and representative selection. The original frequency distribution of keywords, and the final achieved distribution are shown in Fig. 7. Note that as each video has multiple keywords, this is not a proper probability distribution.

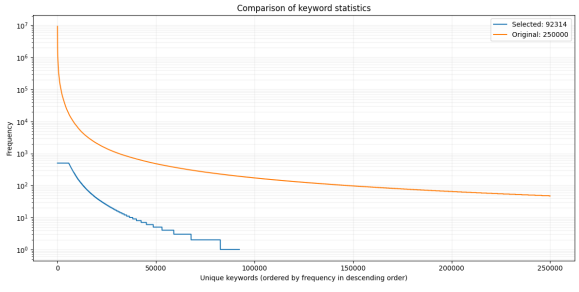


Figure 7: We ensure balance of concepts in our dataset by sub-sampling videos from original dataset by under-sampling common concepts. The y-axis is log-scale.

M High Quality Data Statistics

Our high quality recaptioning of a subset of shutterstock videos significantly increases the level of detail in the captions from having <10 words per caption on average to having 130+ words on average per caption.

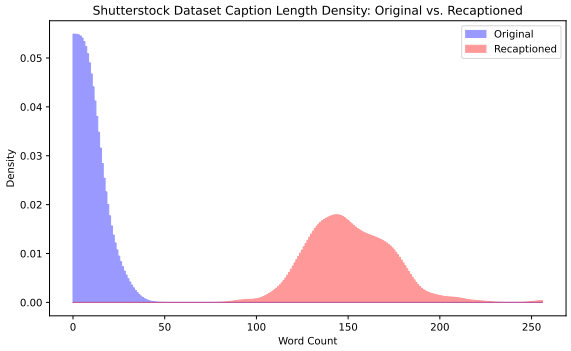


Figure 8: Our captions significantly increases the level of detail. Blue - original captions, Red - Generated.

N Qualitative Video Captioning Results

ViLL-E is able to generate rich video captions exceeding even the quality of some ground-truth captions in datasets. Some examples of this on the MSR-VTT dataset are provided in Figure 11. The figures compare ground-truth descriptions with those generated by the ViLL-E system, highlighting its ability to enrich scene interpretations. For a rocket launch, the ground-truth describes a simple ascent with smoke at the base, while ViLL-E adds detail about a vapor trail over a forested area. In a depiction of miniature donkeys, the ground-truth focuses on basic movement and noise, whereas ViLL-E elaborates on their interactions, feeding, and social behavior on a farm. Similarly, a scene

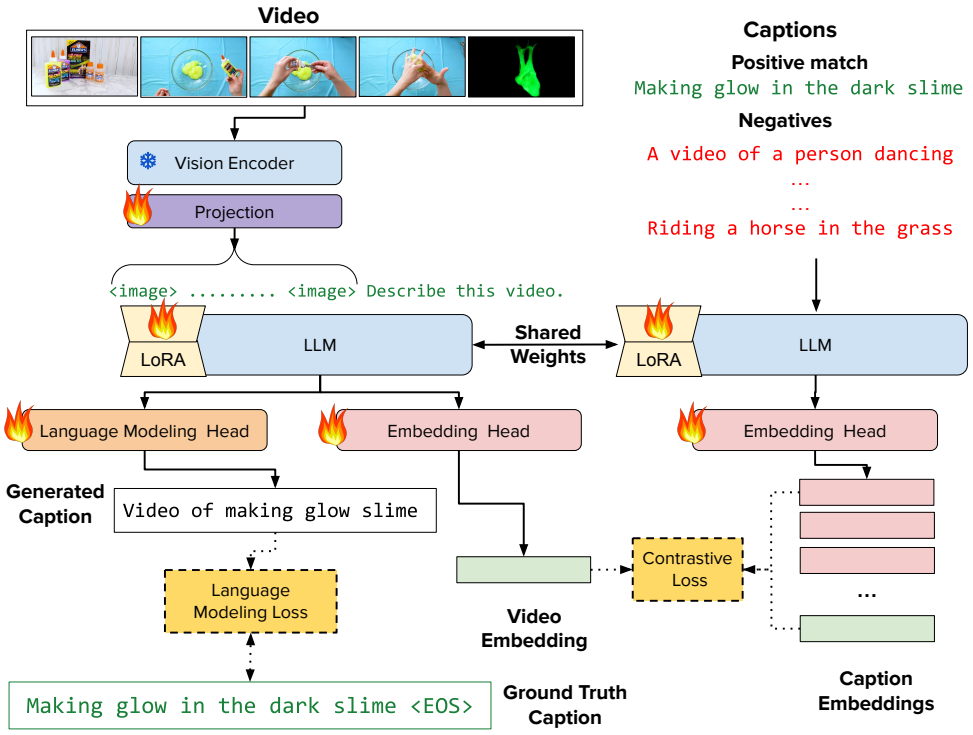


Figure 9: **Stage 1 Training**. Joint Generative and Contrastive Pre-Training. This stage requires 2 forward passes of the model per step. In the first step, the videos along with prompt are passed to the model and the video caption and video embedding are generated simultaneously. In the second forward pass, only the captions are passed to generate caption embeddings. The ground-truth caption for the video serves as the positive match, whereas the captions for the other videos in the batch are negatives for the contrastive loss.

of cartoon birds flying is expanded by ViLL-E into a vivid portrayal of a large flock creating intricate patterns over a mountainous landscape. These comparisons underscore ViLL-E's capacity to provide more nuanced and context-rich scene descriptions.

O Visualization of Video Embeddings

Text and Video embeddings for video from a held out val set for ShutterStock are visualized after being reduced to two dimensions with Barnes-Hut t-SNE (perplexity = 30, init = PCA), producing two coordinates for every video–caption pair while approximately maintaining local neighbourhood structure. To aid qualitative inspection, we cluster the caption embeddings with K-means (k = 20). The dominant, non-stop-word within each cluster's captions is used as an interpretable label (e.g., "beach", "office", "smiling"). The visualization is rendered as a 4 by 5 grid: each panel isolates one cluster, plotting video embeddings as filled circles and caption embeddings as crosses. Grey lines connect every video point to its paired caption point, revealing intra-cluster fine-grained alignment patterns. The resulting figure offers an intuitive map for understanding the retrieval performance of our model across various domains; a high number of intersecting lines indicates poor performance in that cluster and vice versa. Our model exhibits strong performance on videos with concepts like "couple", "flying", "summer" etc.

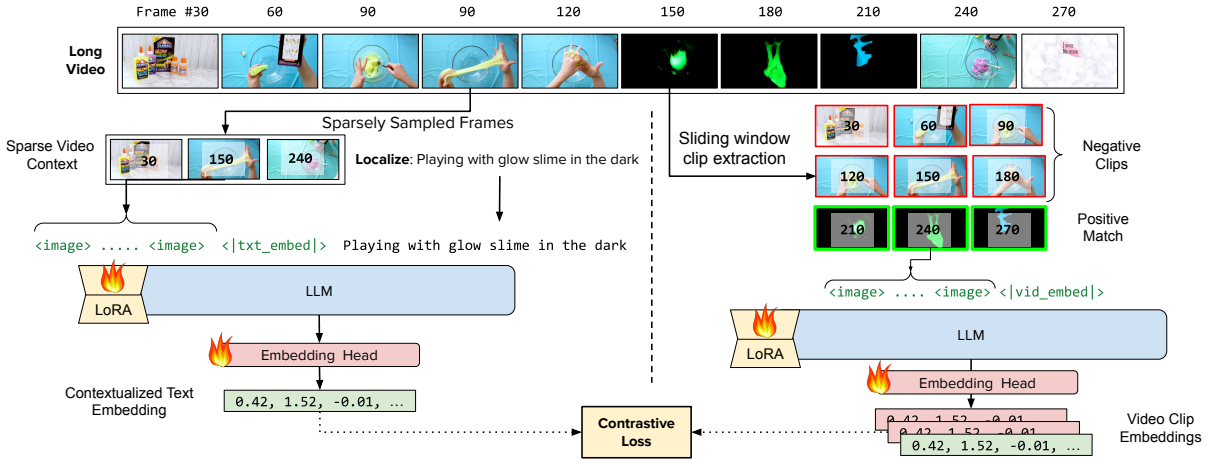
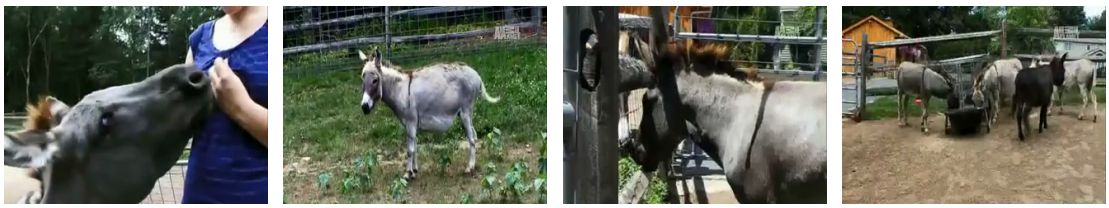


Figure 10: Stage 3: **Temporal Localization Task Training**. Contextualized Text Embeddings (**Left Half**): Text embedding are generated based on text input such as "Playing with glow slime in the dark." A selection of sparse frames from the long video is sampled and added to the prompt to provide context. Sliding Window Video Clip Embeddings (**Right Half**): The long video is divided into clips via a sliding window mechanism, which extracts several short clips. Video embeddings for the clips are used to create the positive and hard negative matches for the text, based on the annotations from the dataset. Training is done using the contrastive loss.

Ground-Truth: a rocket is launching into a blue sky, smoke is emerging from the base of the rocket

ViLLaGE (ours): A rocket launches into the blue sky, leaving a vapor trail over a **forested area**



Ground-Truth: miniature donkeys walking around and making noises

ViLLaGE (ours): A video showcasing donkeys interacting, **being fed**, and displaying social behavior on a farm.

Ground-Truth: cartoon birds are flying

ViLLaGE (ours): A huge flock of birds flies together, **creating intricate patterns** in the sky over a **mountainous** landscape.

Figure 11: Qualitative Video Captioning results on MSR-VTT

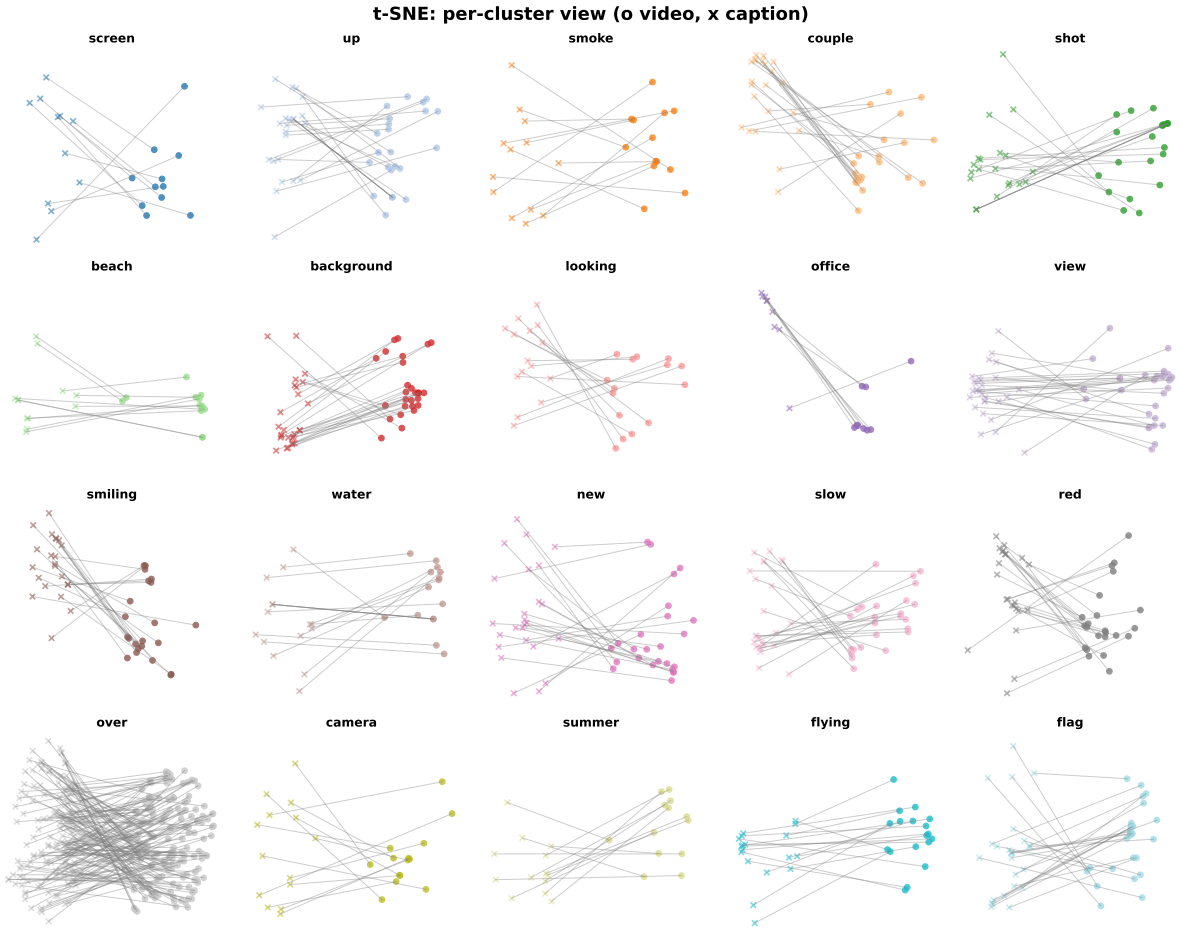


Figure 12: Illustrating our model's retrieval embeddings and their alignment in various sub-domains of videos