VidVec: Unlocking Video MLLM Embeddings for Video-Text Retrieval

Issar Tzachor **Dvir Samuel** **Rami Ben-Ari**

OriginAI

Abstract

Recent studies have adapted generative Multimodal Large Language Models (MLLMs) into embedding extractors for text–vision tasks, typically by fine-tuning them to produce universal representations. However, their performance on video–text retrieval often remains below that of dedicated Video Foundation Models. In this paper, we revisit the video–text retrieval capabilities of MLLMs. We first analyze the zero-shot retrieval capabilities of pretrained MLLMs and show that they already encode substantial retrieval-relevant information: combining intermediate-layer embeddings with a calibrated MLLM head yields strong performance without any training. To further exploit these capabilities, we introduce a lightweight text-based optimization strategy that requires no paired multimodal data. Our method uses dense video captions as rich textual surrogates for video semantics and short summaries as compact, query-like views, showing that a carefully designed text-alignment objective can improve multimodal alignment through the shared MLLM backbone. We backup these findings through in-depth analysis. Remarkably, our lightweight text-only optimization outperforms its counterpart trained with paired video–text supervision, which is substantially more computationally expensive. Furthermore, our method outperforms existing approaches, often by a substantial margin, and achieves state-of-the-art performance across common video-text retrieval benchmarks. More broadly, our findings highlight the untapped retrieval capabilities of pretrained MLLMs and establish text-only optimization as a more efficient and effective substitute for large-scale visual supervision. Project page

1 Introduction

Multimodal Large Language Models (MLLMs) have recently emerged as a dominant paradigm in vision–language understanding, demonstrating strong performance across tasks such as captioning [25], visual question answering [2], and even visual mathematical reasoning [11]. A common design represents visual input as sequences of visual tokens that are fed into a *generative* Large Language Model (LLM) which, after joint fine-tuning, enables open-world reasoning and instruction following over multimodal content. Extending this paradigm to videos has led to rapid progress in video-based MLLMs [70, 14, 67, 49, 65]. Compared to images, videos introduce richer content and temporal dynamics, substantially increasing the challenge of representation learning, reasoning, and retrieval.

As MLLMs become increasingly widespread, multimodal representation learning has attracted growing research interest. Recent large Video Foundation Models (VFMs) have pushed zero-shot and transfer performance by scaling video-text pretraining. InternVideo2 [58] has pretrained on $\sim$100M video-text pairs to reach superior performance, while VideoPrism [71] contrastively trained with $\sim$600M pairs. Recent studies suggest that simply scaling video–text pretraining does not uniformly improve performance [19, 51, 53], motivating alternative approaches that focus on representation quality and task-related alignment [22, 52], rather than relying solely on ever-larger corpora.

Recently, an emerging line of work has investigated whether MLLMs can serve as representation learners across vision–text modalities, reporting impressive results on image retrieval [69, 50, 29, 38, 23, 60, 15]. Yet, video is often not the primary focus, and reported video-retrieval performance often remains below that of dedicated Video Foundation Models (VFMs) [58, 37, 73, 38, 33].

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

In this work, we revisit pretrained MLLMs as embedding extractors *specifically* for video–text retrieval. Although video MLLMs are trained primarily for generative video-language understanding, we show that their hidden representations already contain substantial retrieval-relevant structure. As illustrated in Fig. 1, we first expose this structure in a purely zero-shot setting by extracting video and text embeddings from an intermediate MLLM layer. This yields strong initial retrieval performance without any training (Fig. 1a). We then reuse the MLLM generative head as a calibrated pairwise relevance scorer: given the top retrieved candidates, the model is prompted to answer whether a video matches a sentence, and the likelihood of an affirmative answer is used for reranking (Fig. 1b). This simple two-stage pipeline already outperforms several trained MLLM embedders on standard video–text retrieval benchmarks, suggesting that retrieval-relevant video–text alignment is already encoded in pretrained video MLLMs.

To further unlock these encoded capabilities, we introduce a lightweight *text-only* in-context optimization scheme that refines the embedding space without visual supervision. A few prior MLLM embedders incorporated generic text supervision such as NLI [6] to induce retrieval-friendly representations [26, 38]. We instead propose an in-context text-only optimization paradigm, where the supervision is constructed from textual surrogates of the target multimodal context. For video–text retrieval, we instantiate this paradigm with a video-grounded text-to-text proxy that aligns dense video captions with short summaries, tailoring the optimization signal to the target retrieval setting. Dense video captions serve as rich textual surrogates for video semantics, while short summaries act as compact, query-like views of the same content. Optimizing the MLLM to align these two textual views shapes the shared embedding space to better support video–text retrieval, yielding effective video–text representations at inference time. Notably, using only $\sim$60K text-only pairs, our method not only surpasses direct video–text fine-tuning on the underlying videos but also outperforms MLLM embedders and VFMs trained on orders of magnitude more video–text data.

The key contributions of this work are:

1. We introduce an effective zero-shot scheme that turns off-the-shelf Video MLLM into a competitive video-text retriever.
2. We present an efficient in-context optimization strategy that maps dense video captions to short summaries, enabling task-related embedding learning *without* visual supervision.
3. We show that our text-only optimization is not only substantially more efficient than paired video–text supervision, but also yields stronger retrieval performance.
4. We provide an empirical analysis of our in-context text-only optimization, showing that it induces a reorganization of the embedding space that improves video–text alignment.
5. Without any fine-tuning beyond text, our method achieves state-of-the-art performance on common video retrieval benchmarks.

2 Related Work

Vision–language embeddings for retrieval. Early vision–language models (VLMs) such as CLIP [45], ALIGN [24], BLIP [34], and SigLIP [66] established a powerful dual-encoder paradigm for cross-modal retrieval by aligning image and text representations with contrastive objectives. This recipe has also been extended to video embedding and retrieval [39, 40, 62]. However, dual encoders are less natural for interleaved multimodal inputs, instruction-conditioned retrieval, and video semantics that depend on temporal dynamics, motivating approaches based on instruction-following generative backbones and deeper multimodal fusion.

Video–text retrieval and video representation learning. Video retrieval has traditionally been addressed by video–language models that explicitly encode spatiotemporal structure and are trained on large-scale video–text corpora. Recent Video Foundation Models (VFMs) substantially scale pretraining data and objectives often using huge datasets for training, for the aligned embedding task. For example, InternVideo [57] reports pretraining on roughly 12M video clips spanning multiple domains, HowTo100M trains on nearly 136M short clips [43]. InternVideo2 [58] further scales the data regime to 50M video–text pairs and another 50M multimodal (video–audio–speech–text) pairs. Other methods, *e.g.* , UMT [37], LanguageBind [73] and VideoPrism [71] present state-of-the-art results while building on hundreds of millions of paired samples (video-text), often in addition to millions of image-text pairs.

These approaches typically rely on video-specialized encoders, explicit temporal modeling, and large-scale video–text supervision. In contrast, our work studies *multimodal LLMs as embedders* for video understanding, demonstrated through video retrieval, focusing on (i) whether pretrained video MLLMs already encode retrieval-relevant video–text structure, and (ii) how far video retrieval can be improved using only *text-only* supervision.

Multimodal LLMs as embedders and token-based embedding learning. UniIR [59] introduced instruction-guided multimodal retrieval with image-text paired post-training while MagicLens [68] explored self-supervised instruction-conditioned *image* retrieval using web-mined image pairs and synthesized instructions. Both methods are based on VLM backbones *e.g.* BLIP [34] or CLIP.

In parallel, VLM2Vec [27] proposed an instruction-guided contrastive framework that converts MLLMs into embedding models by LoRA finetuning on large scale MMEB dataset [27]. VLM2Vec-V2 [42] extends this direction by explicitly incorporating videos and visual documents into training.

A growing line of work repurposes multimodal LLMs (MLLMs/LMMs) as embedding models by extracting fixed-dimensional hidden-state representations with carefully designed prompts. E5-V [26] showed that prompt-based "semantic compression" (e.g., asking the model to produce a one-word summary) can create aligned image-text (multimodal) embeddings, even without fine-tuning; critically, E5-V further demonstrated that *single-modality training only on text pairs* from Natural Language Inference (NLI) dataset [6] can generate reasonably aligned multimodal embeddings with low training cost, avoiding the need for image–text training data.

LamRA [38] follows this paradigm, building on large multimodal models and showed significantly improved results with additional image-text paired training. Technically, it uses an "Explicit One-word Limitation" prompt of the form "Summarize ... in one word: `<emb>`" and takes the last hidden state immediately preceding the `<emb>` token as the embedding. It further combines language-only pretraining with multimodal instruction tuning, and uses a separately trained reranking module.

CARE [64] pretrains an MLLM embedder by generating detailed video captions, followed by training on general text pairs. In contrast, UNITE [29] trains on modality-specific pairs (text–text, text–image, and text–video), suggesting that generation-based pretraining does not always improve performance.

Prior MLLM embedders use prompt-based token readout and text-only supervision from generic textual pairs (NLI), but largely overlook the strong zero-shot retrieval signal already present in pretrained MLLMs. In contrast, we first expose this capability through intermediate-layer embeddings, and then introduce text-only *in-context* supervision as a task-grounded alternative, where training pairs are derived from textual views of the target context itself. We instantiate this paradigm for video–text retrieval by aligning dense captions, which proxy video content, with summaries, which provide compact query-like views. This video-specific text-only signal reduces the query–video semantic mismatch, offering an efficient alternative to large-scale video–text contrastive training.

Beyond token prompting and readout design, recent work has improved MLLM-based embeddings through stronger training recipes and richer interaction mechanisms. B3 [50] studies efficient contrastive training with improved batch and negative construction, while UniME-V2 [23] uses an "MLLM-as-a-judge" to assign soft matching scores for hard negative mining. UME-R1 [32] explores reasoning-driven *generative* embeddings with SFT and RL. Qwen3-VL-Embedding [35] combines multimodal embedding learning with a dedicated reranker, Think-Then-Embed [15] conditions embeddings on generated reasoning traces and MetaEmbed [60] introduces compact multi-vector embeddings with flexible late interaction. Our work is complementary: we focus on video–text retrieval and introduce a lightweight dense-caption–to–summary objective that improves video-aligned embeddings without visual supervision.

3 Method

We introduce VidVec, which adapts pretrained generative MLLMs to video–text retrieval while avoiding visual supervision. As illustrated in Fig. 1, VidVec consists of three components. First, in the zero-shot embedding stage (Fig. 1a), we extract video and text representations from an intermediate layer of an off-the-shelf video MLLM. This exposes retrieval-relevant structure already present in the pretrained model, and provides an initial ranking without any training.

Second, in zero-shot reranking (Fig. 1b), we reuse the MLLM language-model head as a calibrated relevance scorer: for each top-ranked candidate pair, the model is prompted to answer whether the video matches the sentence, and the likelihood of an affirmative answer is used for reranking.

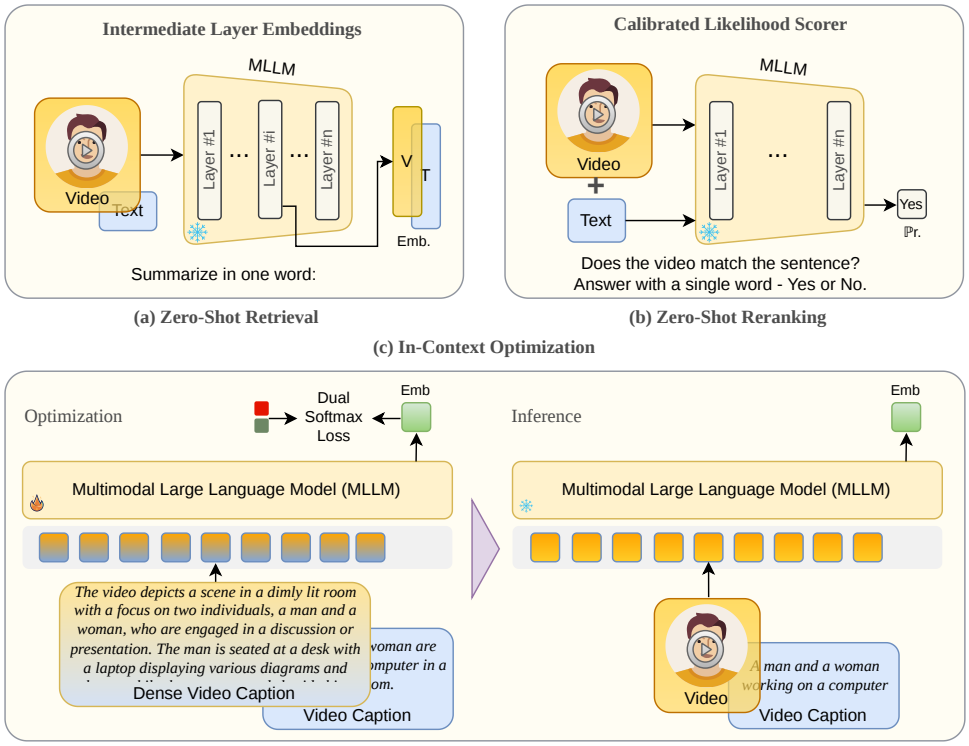


Figure 1: An overview of VidVec. (a) **Zero-shot retrieval:** extract video and text embeddings from an intermediate MLLM layer for initial ranking. (b) **Zero-shot reranking:** leverage the calibrated MLLM head for pairwise scoring to rerank top-K candidates. (c) **In-context optimization:** *lightweight* model alignment using only ∼60K text-only pairs for embedding extraction via a text-to-text mapping from dense video captions to short summaries, designed to mirror the video-text inference setup - dense caption serves as a surrogate for the full video content (*yellow*) and a short summary acts as a compact anchor closer to retrieval queries (*blue*).

Finally, building on this zero-shot capability, we introduce a lightweight text-only *in-context* optimization objective (Fig. 1c). Instead of relying on paired video–text supervision, we use two textual views from the same video context: a dense caption that describes the video in detail, and a short summary that captures its salient semantics in a query-like form. We then align matched dense-caption–summary pairs against mismatched pairs, using the dense caption as a textual proxy for the video and the summary as a proxy for the retrieval query. This in-context text-only optimization improves video–text alignment without direct visual supervision.

We next formalize the retrieval setup, describe the zero-shot embedding and reranking components, and then motivate our *in-context* optimization approach.

We follow the standard video–text retrieval setting, using a generative Multimodal Large Language Model (MLLM). Given a query q (either text or video) and a candidate pool $\Omega = c_1, \ldots, c_N$ of size N, where c_i belongs to the other modality of q, we encode the query and each candidate with an embedding extractor $f_\theta(\cdot) \in \mathbb{R}^d$, where θ denotes the parameters of the MLLM. We score each candidate by cosine similarity and rank Ω accordingly. For second stage re-ranking, we retain the top-K candidates, denoted by $\mathcal{R}_K$, and apply a reranking function over this reduced set to obtain the final ranked list, $\mathcal{R}_{\text{final}} = \Phi_{\text{rerank}}(q, \mathcal{R}_K)$, where $\mathcal{R}_{\text{final}}$ is the final ordered set of candidates produced by the reranking stage.

3.1 Embedding Extraction

Multimodal large language models (MLLMs) typically comprise a vision encoder, a projector that maps visual features into the language-model input space, and a language model. After projection, visual tokens and text tokens are processed by the shared multimodal backbone [70, 3, 54, 67]

To extract embeddings from MLLMs, we follow prior work [26, 38] and adopt an *Explicit One-word Limitation* (EOL) prompting scheme, in which the prompt ends with the instruction to respond ‘‘`in one word`’’ followed by a dedicated `<emb>` token. We take as the embedding the hidden state immediately preceding `<emb>` and refer to it as `<emb-1>`.

3.2 MLLM Head as a Calibrated Likelihood Scorer

For the second stage of re-ranking, we reuse the language model head as a calibrated likelihood scorer. Given the reduced candidate set $\mathcal{R}_K$, each query–candidate pair (q, c_i) is evaluated independently by prompting the model with a binary relevance question. Concretely, we extend the EOL prompting scheme with the instruction ‘‘`Respond in a single word - Yes or No.`’’. We then define the re-ranking score for each candidate c_i as the likelihood of the affirmative tokens `Yes`/`yes`, $S_{\text{rank}}(q, c_i) = P_\theta(\texttt{Yes} \mid q, c_i)$, where S_{rank} is the relevance similarity score used for final ranking. This procedure requires K forward passes and induces the re-ranked list, where candidates in $\mathcal{R}_K$ are ordered by decreasing S_{rank}. Fig. 1b describes this setting.

3.3 In-context Text-only Optimization

In this section, we motivate our text-only in-context optimization as a proxy objective for text–video retrieval while empirical support is provided later in Sec. 6. Fig. 1c illustrates the setup.

Let us consider a triplet $\{v, t_d, t_s\}$ denoting a video sequence, its corresponding dense caption and short summary respectively. Our ultimate retrieval objective is to align a textual query t_q with its corresponding video embedding, i.e.,

$$\min_\theta D\big(f_\theta(t_q), f_\theta(v)\big),$$

where $D(\cdot, \cdot)$ is a distance measure in the embedding space. Since our optimization is text-only, we instead employ the following surrogate objective, aligning dense-captions with short-summaries:

$$\min_\theta D\big(f_\theta(t_s), f_\theta(t_d)\big).$$

Our motivation is based on two observations. First, in our zero-shot analysis (see Sec. 6 and Appendix Sec. I), embeddings of dense captions are better aligned with the paired video embeddings than embeddings of short summaries. We therefore treat the dense caption t_d as a stronger textual proxy for the video's semantic content in the pretrained embedding space. Second, a short summary compactly captures the video's salient semantics. We therefore treat t_s as analogous to a query t_q.

Since the pretrained backbone already exhibits strong zero-shot video–text retrieval capabilities, as demonstrated in Sec. 4.2, the objective should not be interpreted as learning a literal mapping $t_s \longleftrightarrow t_d$. Rather, it serves as a proxy for exposing and refining a shared latent representation that is already encoded in the model.

Our text-only optimization is designed to improve video-text alignment by making the textual representations induced by the same video semantics more consistent. The effect on video representations is indirect but well motivated. Let z_v denote the underlying semantic content of video v. We treat its dense caption t_d and short summary t_s as textual realizations of this content: t_d preserves richer visual and temporal detail, while t_s provides a compact abstraction. Thus, both correspond to the same latent video semantics z_v. We optimize θ such that, for matched pairs (t_s^i, t_d^i), the similarity is maximized relative to mismatched pairs (t_s^i, t_d^j) with $i \neq j$. This encourages invariance to caption length, phrasing, and descriptive granularity, while preserving the shared semantics induced by the same video semantics.

Crucially, both text and video are encoded by the same shared backbone f_θ, while the vision encoder and its projector are frozen. Thus, updating θ to better align the two textual views associated with the same latent video semantics jointly reshapes the induced representation of the corresponding video.

We adopt Dual-Softmax Loss [13] as the objective for our in-context optimization. This loss is commonly used in video–text contrastive training [4, 39, 57]. Given a minibatch of paired dense captions and short summaries, we compute the cross-view similarity matrix between $f_\theta(t_s^i)$ and $f_\theta(t_d^j)$. We then apply softmax normalization along both axes of this matrix, yielding conditional probability distributions for dense-caption-to-summary and summary-to-dense-caption retrieval. These two distributions are multiplicatively combined to emphasize pairs with high mutual matching score under both retrieval directions, and the objective is applied symmetrically.

Overall, this yields a lightweight adaptation scheme that operates entirely in text space using in-context captions derived from the same video. Compared with full video–text fine-tuning, it is far more efficient, avoids the need for paired visual supervision, and provides an effective way to improve modality alignment without directly retraining on video inputs.

4 Evaluation

In this section, we compare our approach against several state-of-the-art MLLM embedders and VFMs. For MLLMs, we evaluate seven methods, including UNITE [29], MMRet-v1.5 (BGE-VL) [72], VLM2Vec-V2 [42], and the recent UniME-V2 [23]. For VFMs, we compare to seven methods, including InternVideo2-6B [58], VideoPrism-g [71], and PE-Core-G [5].

We conduct an evaluation with four common video-text retrieval benchmarks - MSR-VTT [63], MSVD [10], VATEX [56], and DiDeMo [1]. Additional information on the benchmarks is provided in the Appendix. Following prior *zero-shot* protocol [57, 58, 71], we ensure models are not exposed to benchmark training sets and compare only MLLM embedders with confirmed non-exposure [7, 35].

VLM2Vec-V2 [42] introduced the MMEB-V2 benchmark, which spans multiple video tasks. Here we focus on video–text retrieval and use the standard datasets and splits adopted by recent VFMs [71, 58, 5] to enable direct comparison with current state-of-the-art methods. We still compare against VLM2Vec-V2, but report results on the established retrieval splits. While MMEB-V2 includes some existing retrieval datasets, it does not evaluate the V2T setting and modifies several test splits, making direct comparisons to current state-of-the-art methods difficult. For completeness, in the Appendix we compare against the MMEB-V2 reported results on the overlapping video-retrieval datasets, where VidVec achieves superior performance.

We present three variants of our method: (1) a zero-shot two-stage baseline (VidVec-ZS), which involves *no training*, (2) a one-stage in-context optimization approach (VidVec-O), and (3) a two-stage (VidVec) with optional reranking.

4.1 Implementation Details

For our in-context optimization, we leverage the publicly available VideoUFO dataset [55], which contains over 1.09M video clips, each paired with both brief and dense video captions. From this corpus, we sample 60K data entries and fine-tune the model to map dense descriptions to short summaries. We fine-tune a lightweight LoRA [41], and use a batch size of 288 pairs distributed across 4 B200 GPUs, which completes in under 30 minutes. For reranking, we set $K = 5$ - processing each query–candidate pair incurs a latency of 121 ms on a single B200 GPU. We adopt VideoLLaMA3 [67] as our backbone for its superior zero-shot performance (see Appendix). For the zero-shot variant VidVec-ZS, we use embeddings extracted from layer 24, whereas for the optimized variant VidVec-O we use the final-layer embeddings. Videos are sampled at 2 FPS (matching the Qwen2-VL default) and capped at 180 frames. Following prior VFM work, we also apply dual-softmax at inference to boost performance [13, 57, 58, 5]. To ensure a fair comparison, we re-run competing MLLM embedders, where applicable, using the same FPS and a dual-softmax calibrated scoring. This often improves performance relative to the numbers reported in the original papers. For VFMs, we report the published results. Additional implementation details are provided in the Appendix.

4.2 Zero-shot Performance

In Tab. 1, we evaluate our zero-shot two-stage method, against state-of-the-art MLLM embedders *trained* to produce vision–language embeddings. We report results on text-to-video (T2V) retrieval for MSR-VTT, VATEX and DiDeMo. While prior methods rely on contrastive training over vision–text pairs, VidVec-ZS leverages an off-the-shelf MLLM by extracting embeddings from a fixed intermediate layer, followed by a re-ranking stage. Despite requiring no training, VidVec-ZS surpasses prior methods across all three benchmarks, achieving Recall@1 gains of $+2.3\%$, $+6.2\%$, and $+6.8\%$ on MSR-VTT, VATEX, and DiDeMo, respectively. Notably, using intermediate-layer embeddings alone (denoted as VidVec-ZS-Emb), already yields competitive performance and outperforms multiple prior methods. This indicates that off-the-shelf generative MLLMs already encode well-aligned video–text representations internally, a property demonstrated here and not previously established in prior work [26, 38, 23].

Table 1: Text-to-Video (T2V) Retrieval - our two-stage VidVec-ZS (*zero-shot*) vs. state-of-the-art MLLM embedders (7B size).

Model	MSR-VTT	VATEX	DiDeMo
LamRA	48.9	61.4	46.3
VLM2Vec	41.6	53.6	37.7
MMRet-v1.5	45.2	57.7	45.1
B3	46.7	57.3	38.7
VLM2Vec-V2	41.0	52.9	41.7
UNITE	46.5	45.7	43.5
UniME-V2	40.6	52.0	36.5
VidVec-ZS-Emb	42.1	54.7	38.8
VidVec-ZS	**51.2**	**67.6**	**53.1**

Table 2: Text-to-Video (T2V) retrieval - VidVec-O (*Optimized Embedding*) vs. state-of-the-art MLLM embedders MLLM embedders (7B size) on four benchmarks. VidVec-O uses only 60K *text-only* in-context pairs, whereas prior methods are trained on $\times$10-$\times$100 larger scale of *vision-text* data.

	MSR-VTT			MSVD			VATEX			DiDeMo		
Model	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
LamRA	48.9	71.6	79.5	55.7	81.7	88.0	61.4	90.6	95.2	46.3	71.8	80.5
VLM2Vec	41.6	64.5	73.8	52.1	78.4	85.8	53.6	84.4	91.0	37.7	60.9	69.9
MMRet-v1.5	45.2	67.5	75.9	49.5	76.6	83.5	57.7	87.5	93.5	45.1	70.7	78.8
B3	46.7	67.6	75.5	53.8	80.0	86.7	57.3	86.5	92.3	38.7	62.6	70.7
VLM2Vec-V2	41.0	63.6	71.9	48.2	75.4	83.1	52.9	83.2	90.4	41.7	66.5	75.1
UNITE	46.5	69.4	78.0	50.4	78.2	86.4	45.7	74.0	81.0	43.5	69.6	77.3
UniME-V2	40.6	62.5	70.7	52.1	77.5	84.4	52.0	80.8	86.9	36.5	59.2	66.7
VidVec-O	**52.5**	**76.3**	**83.8**	**60.8**	**84.9**	**90.1**	**68.2**	**93.6**	**96.8**	**53.7**	**79.4**	**85.0**

Table 3: Video-to-Text (V2T) retrieval - VidVec-O (**O**ptimized Embedding) vs. state-of-the-art MLLM embedders (7B size) on four benchmarks.

	MSR-VTT			MSVD			VATEX			DiDeMo		
Model	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
LamRA	50.9	72.8	80.0	81.2	92.5	95.5	82.5	95.4	97.7	47.1	72.8	81.5
VLM2Vec	44.9	68.6	76.1	81.3	93.1	95.7	80.2	95.5	98.7	41.3	65.4	72.5
MMRet-v1.5	47.9	70.9	77.7	77.2	90.9	93.4	81.1	95.7	97.7	46.1	73.1	80.7
B3	48.7	70.5	77.9	83.1	95.2	97.2	83.1	96.4	98.5	43.3	65.3	72.5
VLM2Vec-V2	42.9	64.8	72.2	82.1	95.1	97.5	79.9	95.9	98.2	44.2	68.0	76.1
UNITE	45.2	70.3	79.3	76.1	91.3	94.6	78.9	95.3	98.1	40.3	68.7	78.1
UniME-V2	46.3	65.7	72.4	81.8	93.4	95.5	81.5	96.1	98.4	38.7	61.0	69.0
VidVec-O	**54.9**	**77.5**	**84.1**	**85.7**	**95.5**	**97.8**	**89.6**	**98.5**	**99.3**	**56.5**	**79.7**	**86.0**

4.3 Comparison with State-of-the-Art

In this section, we evaluate our *lightweight* in-context optimization approach for extracting output embeddings. We assess the resulting embeddings on four standard video–text retrieval datasets, comparing with state-of-the-art MLLM embedders and VFMs under the same evaluation protocol.

We compare the embeddings extracted by our optimized model, VidVec-O, against recent MLLM embedders. Importantly, VidVec-O uses no reranking. In Tab. 2 and Tab. 3, we report Text-to-Video (T2V) and Video-to-Text (V2T) results respectively. Across both retrieval directions, VidVec-O consistently surpasses existing MLLM embedders, often by a substantial margin. The results highlight the effectiveness of our approach, yielding notable Recall@1 improvements, including V2T +6.5% on VATEX and +9.4% on DiDeMo. This demonstrates that our in-context optimization can effectively elicit the model's internal knowledge using relatively small amounts of textual data, yielding better-aligned video-text embeddings than prior methods trained on millions of vision–text pairs.

While text-based optimization leads to performance gains, the calibrated head reranker in VidVec-ZS provides even larger improvements. Moreover, applying this reranker to only a handful of VidVec-O top candidates further improves text-to-video retrieval, achieving state-of-the-art performance.

Tab. 4 reports results obtained by adding a second-stage reranking on top of VidVec-O. We compare Recall@1 performance against state-of-the-art Video Foundation Models (VFMs).

Regarding training data, Perception Encoder (PE-Core) [5] and VideoPrism [71] used approximately 5.4B image-text (and 22M video–text pairs) and 1B image–text and 600M video–text pairs, including 36.1M manually labeled samples for training. InternVideo2-6B [58], which previously achieved state-of-the-art performance on most benchmarks, is trained on about 300M image–text pairs and 100M video–text pairs, with additional gains achieved from a second-stage re-ranking. VidVec delivers state-of-the-art performance on the majority of evaluated benchmarks in both T2V and V2T retrieval. +1.2% improvement on MSR-VTT V2T, and +1.2% on MSVD T2V. On VATEX, VidVec underperforms by -1.5% on T2V but improves by +4.2% on V2T, and on DiDeMo, while underperforming by -0.6% on V2T, it improves by +3.9% on T2V.

Although VidVec uses a video-pretrained MLLM backbone, it is not explicitly trained for modality alignment or embedding extraction. This highlights the effectiveness of our lightweight in-context optimization and shows that strong video–text representations can be efficiently unlocked from a powerful MLLM. We further demonstrate generalizability across MLLM backbones in Appendix B.

Table 4: Video-Text retrieval performance (Recall@1): VidVec vs. state-of-the-art Video Foundation Models (VFMs). Dashed entries indicate not reported results in the corresponding paper.

Method	V-T Pairs	MSR-VTT		MSVD		VATEX		DiDeMo	
		T2V	V2T	T2V	V2T	T2V	V2T	T2V	V2T
Clip4Clip[39]	n/a	32.0	-	38.5	-	-	-	-	-
ViCLIP[62]	n/a	42.4	41.3	49.1	75.1	-	-	18.4	27.9
InternVideo-L[57]	12M	40.7	39.6	43.4	67.6	49.5	69.5	31.5	33.5
UMT-L[37]	10M	42.6	38.6	49.9	75.4	-	-	48.6	49.9
LanguageBind[73]	10M	44.8	40.9	53.9	72.0	-	-	39.9	-
VideoPrism-g[71]	600M	52.7	51.7	-	-	62.5	77.1	-	-
PE-Core-G[5]	22M	51.2	49.9	59.7	85.4	-	-	-	-
InternVideo2-6B[58]	100M	55.9	53.7	59.3	83.1	**71.5**	85.3	57.9	**57.1**
VidVec	Text-Only	**56.4**	**54.9**	**61.6**	**85.7**	70.5	**89.6**	**62.3**	56.5

5 Direct Video–Text Supervision and Ablations

In Tab. 5, we compare our text-only optimization with direct video–text fine-tuning using the corresponding VideoUFO clips under the same training configuration. Our method substantially outperforms video–text training under matched compute and, remarkably, achieves superior performance even when the latter uses all training examples at approximately 20× the training time. This suggests that textual supervision is not only more efficient but can also be more effective than direct visual supervision. Further details and discussion are provided in Appendix C.

Table 5: Text-only optimization vs. direct video–text fine-tuning.

Supervision	Training Regime	# Train Examples	Training Time	MSR-VTT			DiDeMo		
				R@1	R@5	R@10	R@1	R@5	R@10
Video–text	Compute-matched	∼3.5K	$<$ 0.5 h	43.3	66.6	75.5	35.8	60.9	71.8
Video–text	Full data	∼60K	∼10 h	51.8	73.6	81.9	51.4	74.8	81.4
Text only	ours	∼60K	$<$ 0.5 h	**52.5**	**76.3**	**83.8**	**53.7**	**79.4**	**85.0**

Additional ablation studies on alternative text-to-text mapping strategies, optimization objectives, prompt sensitivity, and zero-shot intermediate-layer performance are provided in Appendix E.

6 In-context Optimization Analysis

In this section, we examine our in-context optimization approach beyond retrieval performance.

In-Context Textual Proxy. To examine the relationship between video embeddings, dense captions, and short summaries, we compute their Centered Kernel Alignment (CKA) [30] values before and after our [dense caption]-[short summary] optimization. CKA measures the similarity between two representation spaces by comparing their centered similarity/Gram matrices. It is useful for assessing whether two embedding sets preserve comparable structure, such as similar pairwise similarity patterns, being robust to orthogonal transformations and isotropic scaling, or feature dimension. Fig. 2 shows alignment patterns. In the zero-shot setting (blue), dense-caption–video alignment is the strongest among all pairwise alignments, suggesting that dense captions can serve as a proxy for the visual video representation in the MLLM embedding space. After optimization (yellow), all CKA values increase substantially. Despite the objective being defined only over [dense-caption]–[short-summary] pairs, we also obtain a significant increase in [short-summary]–[video] alignment which

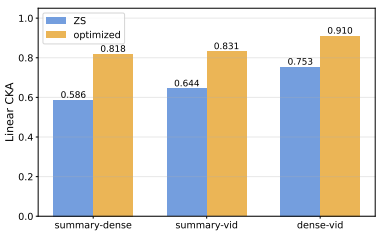


Figure 2: CKA between video embeddings, dense-caption embeddings, and short-summary embeddings before and after our text-only dense-caption–summary optimization. Optimization improves text-text and text-visual alignment.

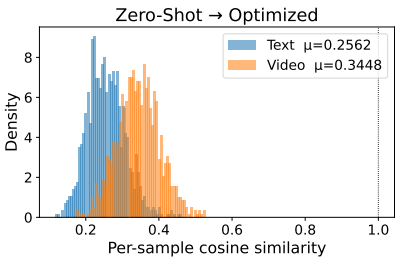


Figure 3: Similarity between each sample and its corresponding pre-optimization representation. The low similarities indicate feature drift in both modalities.

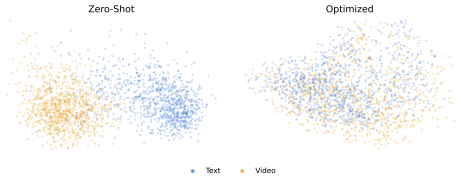


Figure 4: Joint PCA visualization of text and video features before and after ourin-contexttext-only optimization.

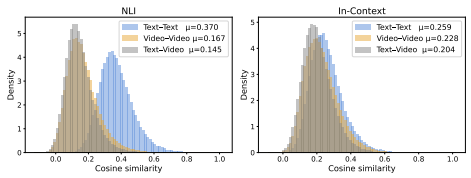


Figure 5: Pairwise cosine similarity distributions for NLI- and in-context-optimized models, where in-context optimization yields more balanced intra- and cross-modal similarities.

helps explain the improved retrieval performance, as short summaries better resemble textual queries. Moreover, although the optimization targets alignment in the text domain, the alignment between dense captions and videos is increased. Together, these results suggest that our optimization might not merely reshape the textual space, but also modifies an underlying latent video semantic space.

In-context optimization changes both text and video representation spaces. To further analyze post-optimization change in the representation spaces, we measure each sample's similarity to its pre-optimization representation and report the results in Fig. 3. The low similarity values indicate substantial post-optimization shifts in both modalities, suggesting that the optimization reshapes the shared multimodal geometry rather than merely improving text-pair alignment.

Optimization promotes a more isotropic representation space for both modalities. Anisotropy is a well-documented property of neural embedding spaces, where representations concentrate in a narrow cone or along a few dominant directions, causing unrelated samples to exhibit high cosine similarity [17, 9, 44]. This issue also appears in modern vision–language models, whose shared embedding spaces can show modality-specific concentration, modality gaps, and limited uniformity [36, 18, 16]. Following [17], we quantify anisotropy ($\mathcal{A} \in [0, 1]$) by the average cosine similarity between randomly sampled features (lower indicates more isotropic space). In the zero-shot regime, both modalities exhibit strong anisotropy, with $\mathcal{A} = 0.84$ for text and 0.86 for video. After optimization, these values decrease to 0.26 and 0.23, respectively, indicating a substantially less collapsed representation space. A similar trend is observed for the inter-modal video-text space, where anisotropy drops from $\mathcal{A} = 0.78$ before optimization to $\mathcal{A} = 0.21$ afterward. These reductions suggest that optimization improves the cross-modal shared space structure.

PCA visualization suggests a smaller modality gap. As a qualitative complement, we visualize text and video embeddings in a joint PCA subspace. In the zero-shot setting, the two modalities appear partially separated, suggesting a visible modality gap in the leading principal directions. After optimization the text and video embeddings appear more intermixed in a shared subspace. Thus, Fig. 4 qualitatively suggests that in-context optimization reduces the visible separation between modalities in the joint embedding space.

In-context optimization is stronger than NLI for video–text alignment. Prior MLLM embedders use text-only NLI supervision [26, 38], which provides a generic language-alignment signal rather than a video-grounded one. Fig. 5 compares NLI with our in-context optimization on 500 random samples. NLI yields high Text–Text similarity, but much lower Video–Video and Text–Video similarities, indicating a text-dominant space. In contrast, our in-context objective, based on dense captions and short summaries of the same video, produces more balanced similarities across modalities, suggesting stronger video–text alignment and explaining its gains over NLI-based LamRA in Tab. 2,3.

Overall, our analysis indicates that the zero-shot model already contains latent cross-modal structure, but this structure is partially obscured by anisotropy and modality gap. The proposed proxy objective reduces this mismatch without requiring direct video–text supervision.

7 Summary

In this work, we study off-the-shelf MLLMs for the task of video–text embedding and retrieval. We start by revealing a previously unrecognized retrieval capability (aligned embedding extraction) already embedded in video MLLMs without further training. To further improve multimodal alignment, we introduce an efficient in-context optimization scheme that relies solely on text-to-text mapping, requiring no visual supervision. When tested on video retrieval, the resulting embeddings significantly outperform recent trained MLLM embedders, and our complete approach achieves state-of-the-art results across multiple benchmarks. We further demonstrate the efficiency and retrieval advantages of text-only optimization over its paired video–text-trained counterpart, and provide an in-depth analysis of its impact on both textual and visual domains, providing explainability for this effect. More broadly, this points to pretrained MLLMs as not only generative models, but also data-efficient embedders whose retrieval capabilities can be exposed through carefully designed text-only supervision.

References

[1] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pages 5803–5812, 2017. 6, 15

[2] 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. 1

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 4

[4] 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. 5, 15

[5] Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, et al. Perception encoder: The best visual embeddings are not at the output of the network. *arXiv preprint arXiv:2504.13181*, 2025. 6, 7, 8

[6] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://aclanthology.org/D15-1075. 2, 3

[7] ByteDance. Seed1.6-embedding. URL https://seed1-6-embedding.github.io/. 6

[8] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015. 15, 17

[9] Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. Isotropy in the contextual embedding space: Clusters and manifolds. In *International conference on learning representations*, 2021. 9

[10] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 190–200, 2011. 6, 15

[11] Felix Chen, Hangjie Yuan, Yunqiu Xu, Tao Feng, Jun Cen, Pengwei Liu, Zeying Huang, and Yi Yang. Mathflow: Enhancing the perceptual flow of mllms for visual mathematical problems. *arXiv preprint arXiv:2503.16549*, 2025. 1

[12] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10638–10647, 2020. 15

[13] Xing Cheng, Hezheng Lin, Xiangyu Wu, Fan Yang, and Dong Shen. Improving video-text retrieval by multi-stream corpus alignment and dual softmax loss. *arXiv preprint arXiv:2109.04290*, 2021. 5, 6

[14] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024. URL https://arxiv.org/abs/2406.07476. 1

[15] Xuanming Cui, Jianpeng Cheng, Hong-You Chen, Satya Narayan Shukla, Abhijeet Awasthi, Xichen Pan, Chaitanya Ahuja, Shlok Kumar Mishra, Taipeng Tian, Qi Guo, Ser-Nam Lim, Aashu Singh, and Xiangjun Fan. Think then embed: Generative context improves multimodal embedding. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=AKXXMK5YTI. 1, 3

[16] Sedigheh Eslami and Gerard de Melo. Mitigate the gap: Improving cross-modal alignment in clip. In *The Thirteenth International Conference on Learning Representations*, 2025. 9

[17] Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. 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 55–65, 2019. 9

[18] Abrar Fahim, Alex Murphy, and Alona Fyshe. It's not a modality gap: Characterizing and addressing the contrastive gap. *arXiv preprint arXiv:2405.18570*, 2024. 9

[19] Christoph Feichtenhofer, Yanghao Li, Kaiming He, et al. Masked autoencoders as spatiotemporal learners. *Advances in neural information processing systems*, 35:35946–35958, 2022. 1

[20] 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. In *CVPR*, 2025. 19

[21] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal Transformer for Video Retrieval. In *European Conference on Computer Vision (ECCV)*, 2020. 15

[22] Michael Green, Matan Levy, Issar Tzachor, Dvir Samuel, Nir Darshan, and Rami Ben-Ari. Find your needle: Small object image retrieval via multi-object attention optimization. *Advances in Neural Information Processing Systems*, 38:121621–121652, 2026. 1

[23] Tiancheng Gu, Kaicheng Yang, Kaichen Zhang, Xiang An, Ziyong Feng, Yueyi Zhang, Weidong Cai, Jiankang Deng, and Lidong Bing. UniME-V2: MLLM-as-a-judge for universal multimodal embedding learning. *AAAI*, 2026. 1, 3, 6

[24] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021. 2

[25] Heng Jia, Yunqiu Xu, Linchao Zhu, Guang Chen, Yufei Wang, and Yi Yang. Mos2: Mixture of scale and shift experts for text-only video captioning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 8498–8507, 2024. 1

[26] Ting Jiang, Minghui Song, Zihan Zhang, Haizhen Huang, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and Fuzhen Zhuang. E5-v: Universal embeddings with multimodal large language models. *arXiv preprint arXiv:2407.12580*, 2024. 2, 3, 4, 6, 9

[27] Ziyan Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. VLM2Vec: Training vision-language models for massive multimodal embedding tasks. *ICLR*, 2025. 3

[28] Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. 18

[29] Fanheng Kong, Jingyuan Zhang, Yahui Liu, Hongzhi Zhang, Shi Feng, Xiaocui Yang, Daling Wang, Yu Tian, Fuzheng Zhang, Guorui Zhou, et al. Modality curation: Building universal embeddings for advanced multimodal information retrieval. *arXiv preprint arXiv:2505.19650*, 2025. 1, 3, 6

[30] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMlR, 2019. 8

[31] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *International Conference on Computer Vision (ICCV)*, 2017. 15

[32] Zhibin Lan, Liqiang Niu, Fandong Meng, Jie Zhou, and Jinsong Su. Ume-r1: Exploring reasoning-driven generative multimodal embeddings. *arXiv preprint arXiv:2511.00405*, 2025. 3

[33] Zhibin Lan, Liqiang Niu, Fandong Meng, Jie Zhou, and Jinsong Su. Llave: Large language and vision embedding models with hardness-weighted contrastive learning. *arXiv preprint arXiv:2503.04812*, 2025. 1

[34] 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. 2, 3

[35] Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv*, 2026. 3, 6, 21

[36] Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. *Advances in Neural Information Processing Systems*, 35:17612–17625, 2022. 9

[37] Ye Liu, Siyuan Li, Yang Wu, Chang-Wen Chen, Ying Shan, and Xiaohu Qie. Umt: Unified multi-modal transformers for joint video moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3042–3051, 2022. 1, 2, 8

[38] Yikun Liu, Yajie Zhang, Jiayin Cai, Xiaolong Jiang, Yao Hu, Jiangchao Yao, Yanfeng Wang, and Weidi Xie. LAMRA: Large multimodal model as your advanced retrieval assistant. In *CVPR*, pages 4015–4025, 2025. 1, 2, 3, 4, 6, 9

[39] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval. *arXiv preprint arXiv:2104.08860*, 2021. 2, 5, 8, 15, 17

[40] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *Proceedings of the 30th ACM international conference on multimedia*, pages 638–647, 2022. 2

[41] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. Peft: State-of-the-art parameter-efficient fine-tuning methods. 2022. 6

[42] Rui Meng, Ziyan Jiang, Ye Liu, Mingyi Su, Xinyi Yang, Yuepeng Fu, Can Qin, Zeyuan Chen, Ran Xu, Caiming Xiong, et al. VLM2Vec-V2: Advancing multimodal embedding for videos, images, and visual documents. *arXiv preprint arXiv:2507.04590*, 2025. 3, 6, 17

[43] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2630–2640, 2019. 2

[44] Jiaqi Mu, Suma Bhat, and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. *arXiv preprint arXiv:1702.01417*, 2017. 9

[45] 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. 2

[46] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pages 606–610. IEEE, 2007. 20

[47] Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. *ICML*, 2025. 19

[48] Tyler Skow, Alexander Martin, Benjamin Van Durme, Rama Chellappa, and Reno Kriz. Rankvideo: Reasoning reranking for text-to-video retrieval. *arXiv preprint arXiv:2602.02444*, 2026. 21

[49] Changli Tang, Yixuan Li, Yudong Yang, Jimin Zhuang, Guangzhi Sun, Wei Li, Zejun Ma, and Chao Zhang. video-salmonn 2: Captioning-enhanced audio-visual large language models. *arXiv preprint arXiv:2506.15220*, 2025. 1

[50] Raghuveer Thirukovalluru, Rui Meng, Ye Liu, Mingyi Su, Ping Nie, Semih Yavuz, Yingbo Zhou, Wenhu Chen, Bhuwan Dhingra, et al. Breaking the batch barrier (b3) of contrastive learning via smart batch mining. *arXiv preprint arXiv:2505.11293*, 2025. 1, 3

[51] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022. 1

[52] Issar Tzachor, Boaz Lerner, Matan Levy, Michael Green, Tal Berkovitz Shalev, Gavriel Habib, Dvir Samuel, Noam Zailer, Or Shimshi, Nir Darshan, et al. Effovpr: Effective foundation model utilization for visual place recognition. In *International Conference on Learning Representations*, volume 2025, pages 42817–42839, 2025. 1, 19

[53] Arnas Uselis, Andrea Dittadi, and Seong Joon Oh. Does data scaling lead to visual compositional generalization? *ICML*, 2025. 1

[54] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 4

[55] Wenhao Wang and Yi Yang. VideoUFO: A million-scale user-focused dataset for text-to-video generation. *arXiv preprint arXiv:2503.01739*, 2025. 6

[56] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. 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, 2019. 6, 15

[57] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022. 2, 5, 6, 8, 17

[58] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. Internvideo2: Scaling foundation models for multimodal video understanding. In *European Conference on Computer Vision*, pages 396–416. Springer, 2024. 1, 2, 6, 7, 8, 17

[59] Cong Wei, Yang Chen, Haonan Chen, Hexiang Hu, Ge Zhang, Jie Fu, Alan Ritter, and Wenhu Chen. Uniir: Training and benchmarking universal multimodal information retrievers. In *European Conference on Computer Vision*, pages 387–404. Springer, 2024. 3

[60] Zilin Xiao, Qi Ma, Mengting Gu, Chun cheng Jason Chen, Xintao Chen, Vicente Ordonez, and Vijai Mohan. Metaembed: Scaling multimodal retrieval at test-time with flexible late interaction. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=yKDqg9HwZX. 1, 3

[61] Tianwei Xiong, Yuqing Wang, Daquan Zhou, Zhijie Lin, Jiashi Feng, and Xihui Liu. Lvd-2m: A long-take video dataset with temporally dense captions. *Advances in Neural Information Processing Systems*, 37:16623–16644, 2024. 18

[62] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *EMNLP*, 2021. 2, 8

[63] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 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, 2016. 6, 15

[64] Yifan Xu, Xinhao Li, Yichun Yang, Desen Meng, Rui Huang, and Limin Wang. Carebench: A fine-grained benchmark for video captioning and retrieval. *arXiv preprint arXiv:2501.00513*, 2024. 3

[65] Biao Yang, Bin Wen, Boyang Ding, Changyi Liu, Chenglong Chu, Chengru Song, Chongling Rao, Chuan Yi, Da Li, Dunju Zang, et al. Kwai keye-vl 1.5 technical report. *arXiv preprint arXiv:2509.01563*, 2025. 1

[66] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023. 2

[67] Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025. 1, 4, 6

[68] Kai Zhang, Yi Luan, Hexiang Hu, Kenton Lee, Siyuan Qiao, Wenhu Chen, Yu Su, and Ming-Wei Chang. Magiclens: Self-supervised image retrieval with open-ended instructions. *arXiv preprint arXiv:2403.19651*, 2024. 3

[69] Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. GME: improving universal multimodal retrieval by multimodal llms. *CVPR*, 2025. 1

[70] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024. 1, 4

[71] Long Zhao, Nitesh B Gundavarapu, Liangzhe Yuan, Hao Zhou, Shen Yan, Jennifer J Sun, Luke Friedman, Rui Qian, Tobias Weyand, Yue Zhao, et al. Videoprism: A foundational visual encoder for video understanding. *ICML*, 2024. 1, 2, 6, 7, 8

[72] Junjie Zhou et al. Megapairs: Massive data synthesis for universal multimodal retrieval, 2024. URL https://arxiv.org/abs/2412.14475. 6

[73] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, HongFa Wang, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, et al. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. *arXiv preprint arXiv:2310.01852*, 2023. 1, 2, 8

Appendix

This Appendix includes details on the following topics:

A. Benchmark Datasets
B. Generalization Across Backbones
C. Controlled Video–Text Supervision
D. Evaluations on more datasets
E. Ablation Study
F. Zero-Shot Study
G. Reranking
H. More implementation details
I. Further In-Context Optimization Analysis
J. Limitations
K. Societal Impact
L. Licenses

A Benchmark Datasets

In this section we provide detailed information on the evaluated benchmarks - MSR-VTT, MSVD, VATEX, DiDeMo and ActivityNet. A summary of the datasets is provided in Tab. 6.

MSR-VTT [63] contains 10,000 videos (10–32s each) with 200,000 associated captions. We follow the standard 1k-A test split [4, 39], which consists of 1,000 video–text pairs.

MSVD [10] contains 1,970 videos (1–62s). The train/validation/test splits include 1,200/100/670 videos, respectively. Each video has approximately 40 English captions. In the V2T setting, we treat any caption for a given video as a positive.

VATEX [56] contains 25,991 training videos, 3,000 validation videos, and 6,000 test videos, with 10 paired captions per video in both English and Chinese. Following [12], we evaluate on 1,500 videos from the validation set using English captions only.

DiDeMo [1] contains over 10,000 videos (25-30s), annotated with natural language descriptions of temporally localized events. Following [21, 39], the descriptions associated with each video are concatenated into a single paragraph, and video–paragraph retrieval is evaluated on the standard test split, comprising 1,004 video–text pairs.

ActivityNet [8, 31] contains 20,000 videos. Following [21, 39], we concatenate all descriptions for each video into a single paragraph and evaluate video–paragraph retrieval on the 'val1' split.

Table 6: Benchmark datasets statistics, with standard test subsets.

Dataset	#Videos	#Captions
MSRVTT	1,000	1,000
MSVD	670	27,763
VATEX	1,500	15,000
DiDeMo	1,004	1,004
ActivityNet	4,917	4,917

B Generalization Across Backbones

To evaluate the generalizability of our proposed text-only optimization, we extend our experiments beyond the default VideoLLaMA3 backbone to two additional video MLLMs: Qwen3-VL and Keye-VL-1.5. For each backbone, we apply the same text-only in-context optimization procedure and evaluate the resulting representations under identical settings on four video-text retrieval benchmarks: MSR-VTT, MSVD, VATEX, and DiDeMo. Tables 7 and 8 report text-to-video (T2V) and video-to-text (V2T) retrieval performance, respectively.

The results demonstrate that our text-only optimization can be applied to multiple video MLLM backbones, yielding strong retrieval performance across all four benchmarks and both retrieval direc-

Table 7: Text-to-video retrieval performance across different video MLLM backbones. All models are evaluated after applying our text-only optimization procedure. Results are reported as Recall@K.

	MSR-VTT			MSVD			VATEX			DiDeMo		
Backbone	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Qwen3-VL	50.1	73.2	82.6	52.6	79.2	86.3	60.4	89.6	94.8	46.9	72.0	79.1
Keye-VL-1.5	51.7	75.7	**84.4**	59.8	84.3	89.8	68.0	91.3	95.6	51.2	77.5	83.2
VideoLLaMA3	**52.5**	**76.3**	83.8	**60.8**	**84.9**	**90.1**	**68.2**	**93.6**	**96.8**	**53.7**	**79.4**	**85.0**

Table 8: Video-to-text retrieval performance across different video MLLM backbones. All models are evaluated after applying our text-only optimization procedure. Results are reported as Recall@K.

	MSR-VTT			MSVD			VATEX			DiDeMo		
Backbone	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Qwen3-VL	51.5	75.8	83.0	77.5	89.6	93.1	84.3	96.8	98.3	48.0	71.4	79.1
Keye-VL-1.5	**54.9**	77.2	**85.2**	84.9	**95.5**	**98.1**	88.9	97.1	98.9	51.9	75.6	81.8
VideoLLaMA3	**54.9**	**77.5**	84.1	**85.7**	**95.5**	97.8	**89.6**	**98.5**	**99.3**	**56.5**	**79.7**	**86.0**

tions. In particular, Keye-VL-1.5 achieves performance comparable to VideoLLaMA3, occasionally surpassing it at R@10, while Qwen3-VL also exhibits competitive performance despite generally underperforming the other two backbones. These findings support the applicability of our approach beyond the default backbone, while indicating that retrieval performance remains dependent on the underlying model.

C Controlled Video–Text Supervision

In Sec. 5 of the main paper, we directly compare our text-only optimization with conventional video–text fine-tuning using a controlled baseline trained on the same VideoUFO examples. Specifically, we retrieve the corresponding video clips for the $\sim$60K examples used in text-only optimization and replace the dense-caption–summary supervision with video–brief-caption pairs.

We keep the VideoLLaMA3 backbone, LoRA configuration, optimization objective, data split, and training recipe fixed. Since processing raw video is substantially more expensive than processing text, we evaluate the video–text baseline under two settings: (i) a compute-matched setting, in which the number of training examples is reduced to approximately match the training compute of our text-only method, and (ii) a full-data setting, in which the video–text baseline is trained on all $\sim$60K examples.

As shown in Tab. 5, text-only optimization is substantially more compute-efficient than direct video–text training. Under a matched training budget, the text-only approach can use substantially more training examples from distinct videos and consequently achieves considerably stronger retrieval performance. For example, on DiDeMo, R@1 increases from 35.8 to 53.7.

Interestingly, the advantage is not explained by compute efficiency alone. Even when the video–text baseline is trained on all approximately 60K examples, requiring roughly $20\times$ more training time, it remains below the text-only model across all reported metrics. One possible explanation is that dense captions provide a compact and semantically explicit representation of the video content relevant to text alignment, whereas raw videos contain substantial temporal redundancy and require processing many visual tokens. Consequently, learning the same semantic alignment from video may be both computationally more expensive and more difficult to optimize.

These results demonstrate that, in the controlled VideoLLaMA3 setting, dense-caption–summary supervision is both more computationally efficient and more effective than direct video–text fine-tuning. However, this finding does not necessarily generalize to other architectures, objectives, or training regimes.

D More Evaluations

In this section, we report additional results for VidVec.

D.1 MMEB-v2 Comparison

Recently, [42] suggested a new benchmark, MMEB-v2, for video tasks, which includes few text-to-video retrieval datasets. In Tab. 9 we compare our method with existing MLLM embedders, using the benchmark's publicly available results. We report results on MSR-VTT and DiDeMo, whose text and video splits follow the standard evaluation and were not modified in MMEB-v2. VidVec raw scores outperform other methods also here. However, note that the LamRA paper reports substantially higher results (including for VLM2Vec), suggesting that the MMEB-v2 evaluation protocol or implementation may be suboptimal.

Table 9: Comparison to MMEB-v2 results on Text-to-Video retrieval (Recall@K).

	MSR-VTT			DiDeMo		
Model	R@1	R@5	R@10	R@1	R@5	R@10
LamRA	25.0	47.4	57.6	22.8	44.7	56.9
VLM2Vec	34.5	58.5	67.9	29.3	53.5	62.2
VLM2Vec-v2	28.3	50.4	59.4	30.4	53.9	63.1
UniME-v2	27.6	44.9	52.9	31.5	53.0	61.9
VidVec-O	**48.9**	**71.4**	**81.1**	**43.0**	**69.0**	**76.3**

D.2 ActivityNet Evaluation

We conduct an additional evaluation on ActivityNet [8]. Tab. 10 summarizes the results on this dataset. Since VideoLLaMA3 was exposed to ActivityNet data during its generative training, we exclude these results from the main paper; nevertheless, we report them here for completeness and to support future studies. Notably, VidVec outperforms even methods that were directly finetuned on ActivityNet's training set (+5.1% gain relative to a fine-tuned InternVideo2-6B)

Table 10: ActivityNet retrieval (R@K): Text-to-Video (left) and Video-to-Text (right). *Finetuned on ActivityNet:* CLIP4Clip (ft), ViCLIP (ft), UMT-L (ft), InternVideo2-6B (ft).

	Text-to-Video (T2V)			Video-to-Text (V2T)		
Model	R@1	R@5	R@10	R@1	R@5	R@10
Multimodal LLM Embedders						
LamRA	58.5	85.4	91.6	58.7	84.8	92.5
VLM2Vec	32.8	52.9	61.3	40.4	59.2	66.2
MMRet-v1.5	50.6	76.6	85.8	52.7	77.8	86.8
B3	33.8	51.9	59.6	39.2	57.4	65.3
VLM2Vec-V2	35.5	54.4	61.4	38.8	58.0	65.6
UNITE	25.8	42.9	51.6	31.4	47.6	55.0
UniME-V2	31.4	50.9	59.9	38.5	57.2	64.5
Video Foundation Models						
ViCLIP	15.1	-	-	24.0	-	-
InternVideo-L	30.7	-	-	31.4	-	-
UMT-L	42.8	69.6	79.8	40.7	67.6	78.6
LanguageBind	41.0	68.4	80.0	-	-	-
VideoPrism-g	52.7	79.4	-	50.3	77.1	-
PE-Core-G	54.7	-	-	51.2	-	-
InternVideo2-6B	63.2	85.6	92.5	56.5	82.8	90.3
Fine-tuned on ActivityNet						
CLIP4Clip (ft)	40.3	–	–	41.6	–	–
ViCLIP (ft)	49.8	–	–	48.1	–	–
UMT-L (ft)	66.8	–	–	64.4	–	–
InternVideo2-6B (ft)	74.1	–	–	69.7	–	–
VidVec	**78.4**	**89.7**	**95.2**	**71.9**	**90.8**	**95.8**

In this context, we also revisit Qwen3-VL-Embedding, for which non-exposure to benchmark training sets is not explicitly stated. Its results are comparable to the *fine-tuned* regime, where models are trained on the corresponding benchmark splits [39, 57, 58]. Specifically, on MSR-VTT, VATEX, and ActivityNet, Qwen3-VL-Embedding achieves T2V R@1 scores of 61.3, 71.6, and 75.3, respectively, compared with 62.8, 75.5, and 74.1 for fine-tuned InternVideo2-6B.

E Ablation Study [1]

E.1 Impact of In-Context Optimization

We evaluate the impact of different textual optimization approaches on Recall@1 performance on MSR-VTT. To distinguish the benefits of video-related textual supervision from those of our proposed in-context formulation, we compare against optimization using brief video description pairs (*Video Captions*), in addition to the NLI-based textual data used in prior work. An illustration of NLI and *Video Captions* is provided in Fig. 6, while our in-context approach is described in Fig. 1.

For *Video Captions* optimization, we use the same data split as our in-context optimization. We prompt an LLM (Gemma3 [28]) to generate a revised short caption from the dense caption, avoiding the original short caption. This approach improves over NLI-based optimization, demonstrating the benefit of video-related textual supervision, while our in-context approach performs higher (46.7% vs. 47.7% R@1).

To further assess the impact of caption source, we repeat the optimization using dense–brief caption pairs from the YouTube-clips subset of LVD-2M [61], while keeping the model and training protocol unchanged. This yields comparable, albeit slightly lower, performance (46.2% vs. 46.7% R@1), suggesting that the benefits generalize across caption datasets, although the choice of caption source can influence retrieval performance.

Table 11: Effect of different textual pairs on MSR-VTT T2V Recall@1.

Context	R@1
Zero-shot	14.3
NLI (LamRA)	45.0
Video Captions	46.7
LVD-2M (YT)	47.2
In-Context	47.7

Table 12: Effect of optimization Loss on MSR-VTT T2V Recall@1.

Optimization Strategy	R@1
Zero-shot	14.3
InfoNCE	47.7
DSL	48.9

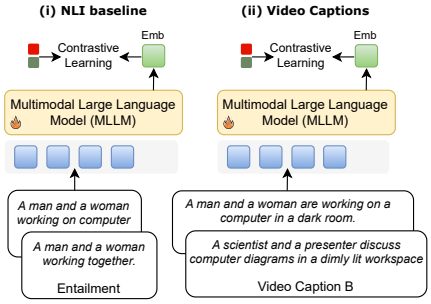


Figure 6: Illustration of the different textual data variants used in our ablation study.

E.2 Optimization Loss

In Table 12 we further report the impact of Dual Softmax Loss (DSL) (only in train) on the results.

E.3 Prompt Selection

We ablate several prompts for embedding extraction, all following the EOL prompting scheme, and find that task-aware prompting plays an important role in off-the-shelf MLLM video processing. In particular, adding the prefix prompt "`Recover the main subject or subjects, appearance and setting, and main activity in the video`" leads to a noticeable performance improvement. We therefore adopt this prefix for the main zero-shot video–text retrieval experiments. After optimization, however, the ablated prompts yield results comparable to the standard prompt used in prior work, "`Summarize above sentence/video in one word: `". To ensure a clean comparison, we use this standard prompt for both the optimized setting and the remaining zero-shot experiments.

[1] The ablation experiments through the appendix were conducted without dual-softmax to isolate the effect of the examined component.

F Zero-Shot Study

We study how zero-shot retrieval performance varies across the internal layers of off-the-shelf MLLMs. Specifically, we ablate the choice of layer used for embedding extraction across multiple Qwen-VL generations and selected video MLLMs of comparable scale, chosen based on their performance on the Video-MME short-clip benchmark [20]. We evaluate Qwen2-VL, Qwen2.5-VL-7B, Qwen3-VL-8B, VideoLLaMA3-7B, and Keye-VL-8B. For each backbone, representations are extracted from the selected layer using the embedding extraction procedure described in Sec. 3.1, and performance is measured on MSR-VTT text-to-video retrieval using Recall@1.

As shown in Fig. 7, layer selection has a substantial impact on zero-shot retrieval performance [47, 52]. Early layers contain little retrieval-relevant signal, whereas several mid- to late-stage layers achieve markedly stronger performance, even without explicit alignment or retrieval-specific fine-tuning. Although deeper layers generally improve retrieval accuracy, the final layer is not optimal in the zero-shot setting. Across the evaluated backbones, VideoLLaMA3-7B achieves the strongest overall performance, with layer 24 yielding the best Recall@1. We therefore use VideoLLaMA3-7B as the default backbone, including for our lightweight in-context optimization, and use its layer-24 representations in the main zero-shot experiments.

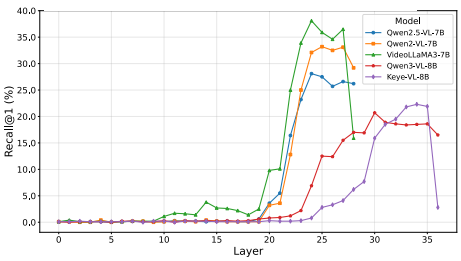


Figure 7: Layer-wise zero-shot embedding performance across different MLLMs on MSR-VTT text-to-video Recall@1. While deeper layers generally yield stronger retrieval performance, the optimal results are not achieved at the final layer. VideoLLaMA3-7B attains the best overall performance.

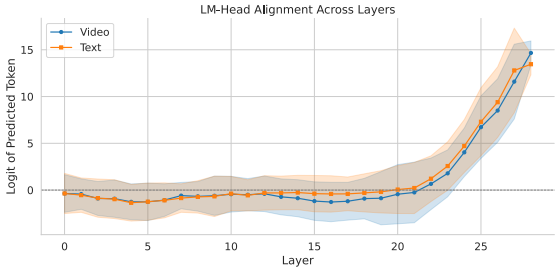


Figure 8: Layer-wise probing with the LM head on VideoLLaMA3 in the zero-shot MSR-VTT setting. Higher layers are increasingly shaped by the LM head's next-token prediction objective.

To better understand why retrieval performance peaks at intermediate layers and degrades near the output layers, we conduct two complementary analyses. First, in Fig. 8 we perform layer-wise probing using the model's next-token prediction layer, i.e., the LM head. This analysis shows that the final layers become increasingly coupled to the autoregressive generation objective. Second, in Fig. 9 we visualize representations across layers using t-SNE, which provides qualitative evidence of a depth-wise representational transition: early layers exhibit a pronounced modality gap, late-intermediate layers show stronger cross-modal semantic alignment. This pattern suggests that intermediate layers are most suitable for zero-shot retrieval because they encode shared video–text semantics while remaining less specialized to next-token prediction.

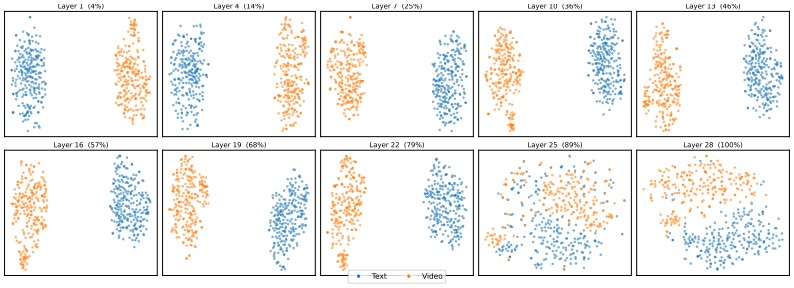


Figure 9: Layer-wise t-SNE visualizations. In the early to intermediate layers, the embeddings of the two modalities occupy relatively distinct regions, indicating the presence of a modality gap. As the layers progress, however, the model eventually maps both modalities into a more closely aligned semantic space.

Interestingly, this behavior changes after optimization. In the optimized setting, the final-layer output becomes competitive with, and in practice preferable to, intermediate-layer embeddings: using the best-performing zero-shot intermediate layer after optimization yields only a negligible Recall@1 gain of less than one point. This indicates that optimization reduces the need for careful intermediate-layer selection, whereas such selection is crucial in the purely zero-shot setting.

F.1 Effective-Rank-Based Layer Selection

To provide a principled criterion for selecting a layer in a new backbone, we investigate the effective rank of the embedding Gram matrix (eRank) [46]. For each layer, we construct the Gram matrix of its embeddings, normalize its eigenvalues to sum to one, and compute eRank as the exponential of the entropy of the resulting eigenvalue distribution. We compute this score independently for the video and text modalities, average the two scores, and select the layer with the highest average eRank.

We evaluate this criterion on the MSR-VTT validation set across the five zero-shot MLLMs considered in this work. For four of the five backbones, eRank selects exactly the same layer as the retrieval-based sweep. Across all five models, the average performance gap relative to the oracle layer selected by the retrieval sweep is only 0.2 R@1 points, suggesting that eRank provides an effective retrieval-independent criterion for layer selection.

F.2 Computational Cost of Layer Selection

The layer-wise sweep is performed only once for each backbone, prior to training. After selecting the optimal layer, inference proceeds normally and incurs no additional overhead. The selection procedure is computationally inexpensive. We extract the hidden states of all intermediate layers in a single forward pass, rather than running the backbone separately for each candidate layer. The subsequent computation of the layer-selection metric introduces negligible additional cost.

We additionally find that the layer selected using the MSR-VTT validation set, containing 497 text-video samples, generalizes well across all evaluated datasets, avoiding the need to repeat the sweep for each dataset.

For our selected backbone, the complete layer-selection procedure takes approximately 1 minute and 45 seconds on a single NVIDIA B200 GPU with a batch size of 1, including model loading and setup. The encoding-time breakdown is reported in Tab. 13.

The additional computational cost of eRank is negligible. It reuses the hidden states extracted during the same forward pass used for layer evaluation and requires only one eigendecomposition per layer and modality. In our implementation, computing the eRank scores across all candidate layers takes less than one second.

Table 13: Encoding runtime for layer selection on the MSR-VTT validation set using the selected backbone. Hidden states from all intermediate layers are extracted in a single forward pass.

Phase	Duration	Time per Sample
Video encoding	1 min 6 sec	$\sim$0.134 sec
Text encoding	10 sec	$\sim$0.021 sec

G Reranking

G.1 Effect of Reranking Depth

We further analyze the effect of the reranking depth K on both retrieval performance and inference cost. Tab. 14 reports the effect of varying K on MSR-VTT text-to-video retrieval.

Reranking consistently improves R@1 over the first-stage retrieval baseline. The best R@1 is obtained with $K = 5$, while increasing the reranking depth leads to a small reduction in R@1. In contrast, larger values of K can improve higher-rank recall, with $K = 10$ yielding the best R@5 and $K = 20$ the best R@10. These results indicate that shallow reranking captures most of the benefit, while larger reranking depths introduce additional inference cost with limited gains in performance.

Table 14: Effect of reranking depth K on MSR-VTT text-to-video retrieval.

Setting	R@1	R@5	R@10
First stage	52.5	76.3	83.8
$K = 5$	56.4	76.3	83.8
$K = 10$	56.2	77.4	83.8
$K = 20$	55.3	76.9	85.0

The reranker requires 121 ms per query-candidate pair on a single NVIDIA B200 GPU. Since candidates are evaluated independently, sequential reranking therefore requires approximately 0.61 s, 1.21 s, and 2.42 s per query for $K = 5$, $K = 10$, and $K = 20$, respectively.

G.2 Comparison with Reranking Methods

We compare our calibrated-head reranker with two recent reranking approaches, RankVideo [48] and Qwen3-VL-Reranker [35], while keeping the first-stage retriever and the top-5 candidate set fixed. Results on MSR-VTT text-to-video retrieval are reported in Tab. 15.

Table 15: Comparison of reranking methods on MSR-VTT text-to-video retrieval. All methods rerank the same top-5 candidates produced by the first-stage retriever.

Method	R@1
First stage	52.5
RankVideo	51.7
Qwen3-VL-Reranker	57.1
Ours	56.4

RankVideo does not improve over the first-stage retriever in this setting, whereas both Qwen3-VL-Reranker and our calibrated-head reranker yield substantial gains. Qwen3-VL-Reranker achieves the highest R@1, outperforming our method by 0.7 points, suggesting that more sophisticated reranking approaches may offer additional improvements. Nevertheless, our lightweight approach remains competitive while requiring neither a dedicated reranking architecture nor reranker-specific training.

H More Implementation Details

In Tab. 16 we report full model names of MLLM embedders used in our evaluation.

Table 16: MLLM embedder baselines used in our evaluation.

Method	Model Name
LamRA	LamRA-Ret
VLM2Vec	VLM2Vec-Qwen2VL-7B
MMRet-v1.5	BGE-VL-v1.5-mmeb
B3	B3_Qwen2_7B
VLM2Vec-V2	VLM2Vec-V2.0
UNITE	Unite-Base-Qwen2-VL-7B
UniME-V2	UniME-V2-Qwen2VL-7B

I Further In-Context Optimization Analysis

To further understand our in-context optimization scheme, beyond the CKA results reported in the main paper, we examine the relationship between video embeddings, dense captions, and short summaries through pairwise similarity distributions. Overall, these distributions are consistent with the CKA findings. As shown in Fig. 10, in the zero-shot setting (blue), the higher video–dense-caption similarity suggests that dense captions can serve as a proxy for the visual video representation in the MLLM embedding space.

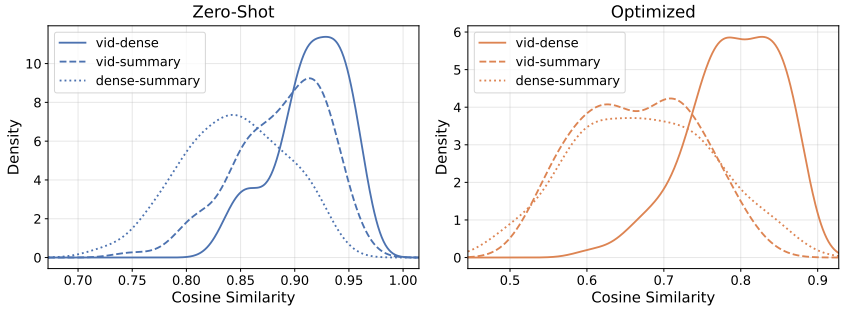


Figure 10: Pairwise similarity distributions among video embeddings, dense captions, and short summaries in the zero-shot setting and after in-context optimization. The high video–dense-caption similarity in the zero-shot setting indicates that dense captions can serve as a proxy for the visual video representation, while the post-optimization distribution shifts suggest improved alignment in a shared video–text semantic space. Results are computed on 100 triplets from the VideoUFO dataset.

After optimization (Fig. 10, yellow), the video–summary similarity distribution closely follows the optimized dense-caption–summary distribution, suggesting that the learned text-only alignment

induces a compatible structure in the video–text embedding space. Moreover, despite being optimized only toward short summaries, dense captions remain highly aligned with the corresponding video embeddings. Since video embeddings are never used in the objective, this indicates that the text-only optimization does not merely reshape the textual space, but also reinforces a shared latent semantic structure already present in the MLLM.

The overall post-optimization shift toward lower cosine similarities reflects an increased isotropy in the embedding space, as discussed in the main paper.

In-Context Optimization. We use LoRA by PEFT on the LLM backbone, optimizing its output token at the `<emb-1>` position. LoRA rank is 64, and alpha is 128. We run our single epoch optimization using deepspeed zero3, with 72 pairs per B200 GPU X 4 resulting in 288 in a batch.

Evaluation. Recent state-of-the-art VFMs (e.g., InternVideo2 and PE-Core) use dual-softmax score calibration, which accounts for the query distribution at inference time. To ensure a fair comparison, we apply the same protocol to our method and to all MLLM embedders. Specifically, we tune one fixed temperature per retrieval direction of T2V and V2T on the MSR-VTT validation set, scale the similarity matrix, and apply dual-softmax by taking the softmax over both columns and rows.

Analysis. The analysis experiments were done on MSR-VTT dataset, except when explicitly stated differently. For zero-shot setting, we used standard EOL prompt.

J Limitations

J.1 Textual Supervision

Although our text-only optimization slightly outperforms its video–text-trained counterpart in overall retrieval accuracy (see Sec. C), we further investigate whether native visual supervision provides advantages for specific visual semantics.

Retrieval-level analysis. On MSR-VTT, the text-only and video-supervised models achieve 54 and 59 exclusive top-1 retrieval successes, respectively. Among queries where either model fails at rank 1, the ground-truth video still appears within ranks 2–5 in 83–89% of cases, suggesting that most errors reflect relatively small ranking differences. We further manually examine the 113 exclusive successes for action verbs, spatial relations, and temporal expressions. These categories are not underrepresented among the text-only model's successes, with spatial and temporal cues appearing more frequently in this group. Thus, we find no aggregate evidence that text-only supervision is selectively weaker on visually demanding queries.

Similarity-level analysis. To examine finer-grained representational differences, we conduct a controlled probe using 20 held-out VideoUFO videos with detailed captions containing visual attributes, spatial relations, and temporal dynamics. For each caption, we construct three manually verified semantic edits and measure their effect on video–text cosine similarity. Altering fine-grained visual attributes produces comparable similarity reductions for the text-only and video-supervised models (0.07 vs. 0.08). In contrast, modifying event order yields a smaller reduction for the text-only model (0.02 vs. 0.04), suggesting a modest advantage of native video supervision in capturing temporal ordering. Spatial-relation edits have limited effects on both models.

Overall, these findings suggest that text-only supervision effectively preserves much of the visual information encoded in the pretrained MLLM, while native video supervision may better capture certain temporal distinctions. Such differences can remain obscured by standard retrieval metrics, as captions often contain multiple redundant identifying cues.

J.2 Reranking

Our full model includes an inference-time reranking stage, bares additional computational cost through additional forward passes over the top-K candidates. This may limit applicability for large K values. Furthermore, our reranking relies on a simple pairwise scoring using the MLLM head, and exploring more advanced reranking strategies remains future work.

K Societal Impact

In our method, biases may be inherited both from the pretrained MLLM backbone and from the text-only alignment data. Since our in-context optimization relies on dense video captions and short summaries without visual supervision, hallucinations, omissions, or stereotypes in the captions may propagate into the embedding space and affect retrieval fairness.

L Licenses

We leverage existing resources, namely the VideoLLaMA3 model and the VideoUFO dataset. VideoLLaMA3 is released under the Apache 2.0 license, while VideoUFO is released under the CC BY 4.0 license. We use these resources in accordance with their respective license terms.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: it reflects

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: in the appendix

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: not applicable.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: implementation details are provided.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: backbone model and data are already public, with sufficient instructions to infer and train. we will publicly release our trained model upon acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: it is detailed in the Evaluation section and the provided implementaion details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We follow the common video retrieval benchmarking, with no error bars.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: we provided training time and resources in the implementation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: we conform with neurips code of ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: see societal impact section.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: similar models are already freely available. although we advance the state of the art, we are not aware of a special high risk of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: in the appendix

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: no new assets

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: does not involve crowdsourcing

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: not applicable

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: core method dev did not involve LLM.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.