LV-MAE: Learning Long Video Representations through Masked-Embedding Autoencoders

Ilan Naiman Emanuel Ben-Baruch Oron Anschel Alon Shoshan
Igor Kviatkovsky Manoj Aggarwal Gérard Medioni

Amazon

{naimanil, emanbb, oronans, alonshos, kviat, manojagg, medioni}@amazon.com

Abstract

*In this work, we introduce long-video masked-embedding autoencoders (LV-MAE), a self-supervised learning framework for long video representation. Our approach treats short- and long-span dependencies as two separate tasks. Such decoupling allows for a more intuitive video processing where short-span spatiotemporal primitives are first encoded and are then used to capture long-range dependencies across consecutive video segments. To achieve this, we leverage advanced off-the-shelf multimodal encoders to extract representations from short segments within the long video, followed by pre-training a masked-embedding autoencoder capturing high-level interactions across segments. LV-MAE is highly efficient to train and enables the processing of much longer videos by alleviating the constraint on the number of input frames. Furthermore, unlike existing methods that typically pre-train on short-video datasets, our approach offers self-supervised pre-training using long video samples (*e.g.*, 20+ minutes video clips) at scale. Using LV-MAE representations, we achieve state-of-the-art results on three long-video benchmarks – LVU, COIN, and Breakfast – employing only a simple classification head for either attentive or linear probing. Finally, to assess LV-MAE pre-training and visualize its reconstruction quality, we leverage the video-language aligned space of short video representations to monitor LV-MAE through video-text retrieval.*

1. Introduction

In recent years, substantial progress has been achieved in the development of models for short-video representation learning. Numerous [5, 28, 30, 32, 44, 47, 49, 51, 55] approaches have been introduced, demonstrating state-of-the-art performance across various tasks such as video classification, action recognition, text-video retrieval, video captioning, and video question answering. While models targeting short video clips (5–15 seconds long) are effective at capturing isolated moments, atomic actions, or specific events, they usually struggle to capture long-range temporal dependencies essential for understanding complex, extended narratives. For example, grasping the full emotional journey of characters in a drama or following the progression of a complex plot in a movie often requires viewing the entire film, not just a brief scene.

Meanwhile, developing models that can effectively handle long video content (*e.g.*, 20 minutes long) remains a significant challenge. First, most current methods process spatio-temporal tokens at the frame level, restricting the input video length they can handle. Second, scaling these methods to accommodate longer sequences becomes computationally prohibitive, as training on extended videos demands significant GPU memory, along with increased training cost and time, limiting accessibility for many researchers and organizations. Third, annotating long videos requires substantial effort, and defining unambiguous annotation guidelines is challenging, often leading to inconsistencies.

Recently, approaches for handling long-video content have been proposed based on state-space models (SSMs) to address some of the above challenges. ViS4mer [21] utilizes a transformer encoder to extract patch-level features from each individual frame, followed by an efficient aggregation of these representations across all frames using the S4 decoder [14]. VideoMamba [27], a fully SSM-based video model, directly processes spatio-temporal patches from video frames using multiple bidirectional Mamba blocks [56]. These architectures successfully bypass the quadratic complexity of the transformer's attention mechanism and are trained on sequences of up to 64 frames [21, 27]. Yet, their computational demands rise significantly when applied to videos of extended duration.

In this paper, we tackle the challenges of learning representations for long video content by proposing a self-supervised approach to pre-train on videos ranging from minutes to hours. Our method is not constrained by the

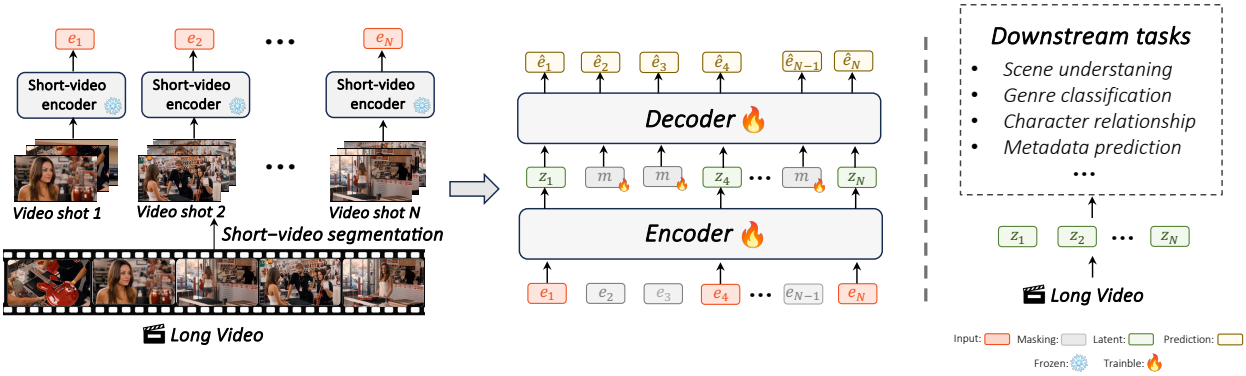

Figure 1. **Overview of the LV-MAE method.** LV-MAE first utilizes short-video representations extracted by advanced multimodal off-the-shelf encoders (*e.g.*, LanguageBind [55], InternVideo2 [44]) to capture low-level knowledge of atomic actions and localized events. Next, we pre-train a masked embedding autoencoder to learn long-range dependencies across video segments in a self-supervised manner. After pre-training, the LV-MAE encoder is used to extract high-level representations for long-video downstream tasks.

number of frames and is highly efficient in terms of training cost. Our key idea is to decouple the extraction of low-level representations, which can be effectively achieved using high-performing off-the-shelf short-video models, from the task of modeling long-range dependencies across short video segments.

Specifically, we propose Long-Video MAE (LV-MAE), a transformer-based architecture that uses representations extracted from a sequence of short video segments by off-the-shelf models to learn high-level, long-video representations through self-supervised training. Our approach begins by segmenting a long video into short clips (*e.g.*, five seconds long) and utilizing a multimodal model (*e.g.*, LanguageBind [55], IntenVideo2 [44]), pre-trained for video-text alignment, to extract low-level embeddings for each segment. We then train a masked-embedding autoencoder that operates on these low-level embedding tokens to learn a high-level representation of the entire video. In particular, we adopt the asymmetric encoder-decoder design from [16] and demonstrate that reconstructing masked embeddings effectively captures general high-level semantic knowledge across extended video sequences. Furthermore, we employ a padding strategy and leverage masked self-attention to accommodate videos of arbitrary length. An overview of the LV-MAE method is provided in Fig. 1.

To highlight the difference between our approach and existing methods, consider the tokens required to process a two-minute long video: while frame-level approaches like ViS4mer [21] and VideoMamba [27] require 11,760 tokens (60 frames $\times$ 14 $\times$ 14 patches for a standard 16 $\times$ 16 image grid), our approach processes only 24 tokens in total, with one token per five-second segment. This significantly reduces the computational burden on the self-attention layer, which has quadratic complexity. Furthermore, our method unlocks the capacity to handle a significantly larger number of frames, potentially reaching thousands. Additionally, we bypass the challenge posed by the lack of annotated long-video datasets, as existing datasets contain only short samples of a few minutes.

A key contribution of our work is enabling pre-training on substantially longer video samples through self-supervised learning. We pre-trained LV-MAE on a large and diverse dataset comprising long-length movies and TV series. Our LV-MAE approach is highly efficient to train, requiring only 2.5 days on a single NVIDIA A10 GPU and 20 hours on 8 NVIDIA A10 GPU.

LV-MAE effectively learns long-range dependencies that generalize well to various downstream tasks. By leveraging the representations learned through LV-MAE, we achieve state-of-the-art performance with simple attentive probing [53] or even linear probing across three long-video benchmarks: LVU [46], COIN [40], and Breakfast [23]. Additionally, we provide interpretable visualizations that monitor the quality of the pre-trained encoder by utilizing the aligned text-video space of the short-video encoder. Specifically, for each reconstructed masked token, we perform retrieval against a large set of captions, resulting in high-quality matches demonstrating the ability to reconstruct the semantic meanings of the masked embeddings.

Our contributions are summarized as follows: (1) We introduce LV-MAE, a method for learning long-range dependencies across short-video segments by pre-training masked-embedding autoencoders on very long video samples (*e.g.*, 20+ minutes). Our method is highly efficient to train and enables the processing of much longer videos by alleviating the constraint on the number of frames. To the best of our knowledge, this is the first approach to apply masked autoencoders on sequences of embedding vectors instead of image patches. (2) Leveraging LV-MAE representations, we achieve state-of-the-art results with mini-

mal fine-tuning, using techniques such as attentive or linear probing, on three long-video downstream tasks: LVU [46], COIN [40], and Breakfast [23]. (3) By exploiting the video-language shared space of the short-video encoder, we introduce an interpretable technique to visualize how semantic knowledge is reconstructed within the LV-MAE encoder.

2. Related Work

Self-supervised learning (SSL). Self-supervised learning (SSL) has brought significant advancements in representation learning by leveraging large-scale unlabeled data [4, 6, 13, 34, 35]. In natural language processing (NLP), models like BERT [8] utilize masked token training strategies to learn contextual embeddings. Inspired by these successes, SSL methods have been adapted to computer vision tasks. In the vision domain, masked image modeling approaches like BEiT [3] and MAE [16] have demonstrated remarkable performance in learning visual representations by reconstructing masked image patches. MAE introduces an asymmetric encoder-decoder architecture that reconstructs masked image patches, leading to efficient training and strong downstream performance. In this work, we adapt the MAE asymmetric encoder-decoder architecture and apply it to embeddings instead of images. Similar to how BERT captures semantic context in language by modeling token dependencies, we aim to learn the semantic context between embedding vectors for long-video understanding tasks.

Masked autoencoders for video understanding. VideoMAE [41] and VideoMAE V2 [42] extend masked autoencoders to video data by reconstructing masked video patches, enabling efficient self-supervised learning. However, they focus on short video clips up to 32 frames, and scaling to longer videos is computationally challenging due to the quadratic complexity of attention mechanisms with respect to sequence length. A decoupling strategy is explored by CoSeg [43], yet it also operates at the *frame* level and therefore, it inherits the same scaling bottleneck. In contrast to these methods, our work addresses long-video understanding by operating on sequences of embeddings and aims to learn long-range dependencies without being constrained by the number of frames. Our novelty lies in the granularity of the low-level representations, leveraging clip-level features to enable efficient training, avoiding costly frame processing.

Long-video understanding methods. Understanding long videos, such as movies or instructional content, is challenging due to the computational cost of processing lengthy sequences and the need to model long-range temporal dependencies. State-space models (SSMs) have been proposed to address some of these challenges. Vis4mer [21] combines a transformer encoder for extracting spatial features with a structured state-space sequence model (S4) [14] decoder to efficiently aggregate representations across frames. VideoMamba [27] is a fully SSM-based video model that processes spatiotemporal patches directly using bidirectional Mamba blocks [56]. While SSMs reduce computational complexity compared to standard transformers, operating directly at the frame level does not fully alleviate computational demands, limiting processing to up to 64 frames [21, 27]. Our work focuses on enhancing long-video representations through a novel approach, specifically designed for long-form content. We evaluate our LV-MAE method on long-form video classification and regression tasks, demonstrating its effectiveness in handling extended temporal sequences. A detailed discussion of long-video language understanding methods that leverage Large Language Models (LLMs) is provided in App. 12, as these approaches, while related, address different aspects of the long-video understanding challenge.

3. Method

In this section, we introduce the LV-MAE framework for learning long-video representations. Our approach operates hierarchically: first, we extract short-video representations using off-the-shelf models such as LanguageBind [55] and InternVideo2 [44]. Next, we capture long-range dependencies across video segments through self-supervised learning. Specifically, we propose a masked-embedding autoencoder that operates on these short-video embeddings, with an architecture inspired by the asymmetric encoder-decoder design proposed in MAE [16]. In the following subsections, we delineate each component of the framework.

3.1. Short-video Representation

Given a full video V, we first segment it into a set of N consecutive short videos (*e.g.*, five seconds long), represented as $\mathcal{V} = \{\mathbf{v}_i\}_{i=1}^{N}$. For each segment, we extract T frames, denoted by $\mathbf{v}_i \in \mathbb{R}^{T\times C\times H\times W}$, where C, H, and W are the number of channels, height, and width of the frames, respectively. Each video segment $\mathbf{v}_i$ is then processed by a multimodal model, pre-trained for short video-text alignment, such as LanguageBind [55] or InternVideo2 [44]. It produces a set of short-video embeddings $\mathcal{E} = \{\mathbf{e}_i\}_{i=1}^{N}$, where $\mathbf{e}_i \in \mathbb{R}^d$ and d is the embedding dimension.

3.2. Masked-embedding Autoencoder

MAE [16] and its video-based variants [41, 42] operate directly on video frames, masking a subset of image patches and reconstructing them at the decoder output. In contrast, our approach processes embedding vectors that represent sequential short video segments.

Formally, given a set of short-video embeddings $\mathcal{E}$, we mask a subset $\mathcal{M} \subset \mathcal{E}$, while only the remaining tokens

$\mathcal{E} \setminus \mathcal{M}$ are passed as input to the encoder,

$$\mathcal{Z} = \text{Encoder}(\mathcal{E} \setminus \mathcal{M}), \tag{1}$$

where $\mathcal{Z}$ is the set of encoded visible embeddings. We then provide the decoder with both $\mathcal{Z}$ and $|\mathcal{M}|$ mask tokens, where each mask token is represented by a shared, learnable vector, denoted as $\mathbf{m}$. Additionally, we incorporate positional encoding at the encoder and decoder to indicate each token's location within the long video sequence.

The decoder then outputs a set of reconstructed embeddings $\hat{\mathcal{E}} = \{\hat{\mathbf{e}}_i\}_{i=1}^{N}$, *i.e.*,

$$\hat{\mathcal{E}} = \text{Decoder}\left(\mathcal{Z}, \mathbf{m}, \mathcal{P}\right), \tag{2}$$

where $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^{N}$ is the set of positional embeddings added to both visible and mask tokens corresponding to their original locations.

Loss function. We use mean squared error (MSE) loss during training, specifically computing the MSE between the original masked embeddings $\mathbf{e}_i \in \mathcal{M}$ and their corresponding reconstructed embeddings $\hat{\mathbf{e}}_i$ as

$$\mathcal{L} = \frac{1}{|\mathcal{M}|} \sum_{\mathbf{e}_i \in \mathcal{M}} \|\mathbf{e}_i - \hat{\mathbf{e}}_i\|_2^2 . \tag{3}$$

Although simple, we find that the MSE loss is effective for pre-training long videos represented as sequences of embedding vectors.

Masking. As previously mentioned, in LV-MAE, we mask a portion (*e.g.*, 50%) of the sequential embedding tokens by removing a subset $\mathcal{M}$ from the input set of tokens $\mathcal{E}$. In this work, we explore two masking strategies: *random* masking and *semantic* masking. In random masking, the subset $\mathcal{M}$ is simply generated by randomly sampling $|\mathcal{M}|$ embeddings and excluding them from $\mathcal{E}$.

Recently, several works have proposed alternative masking strategies based on masking salient regions within the spatio-temporal tokens [10, 17, 26]. Inspired by these efforts, we propose a *semantic* masking strategy that focuses on masking distinct or salient elements within the embedding sequence. Specifically, we employ a simple approach that leverages the semantic information within short-video representations. First, we compute the cosine similarity between each embedding $\mathbf{e}_i$ and its preceding embedding $\mathbf{e}_{i-1}$. Then, we select the embeddings with the lowest cosine similarity values and mask them. This encourages the model to learn more complex dependencies between video segments, enhancing its capacity for in-depth long-video understanding. We demonstrate that in some cases, the semantic strategy yields better results, particularly when employing linear probing for classification.

Handling varying video lengths To handle videos of varying lengths and to maintain computational efficiency, we adopt techniques from BERT [8]. Each sequence of short-video segments is limited to a maximum of 256 tokens, with shorter sequences padded using a special `[PAD]` token. During the self-attention process, we employ an attention mask to prevent these padding tokens from influencing model training.

3.3. Training and Data

Diverse data sources. Our model is pre-trained on diverse datasets across multiple video domains, enhancing its robustness and adaptability to various tasks. Specifically, we pre-train on over 1,000 long-length movies and TV series, encompassing a wide range of genres, styles, and lengths to ensure comprehensive content diversity. In addition, we incorporate three publicly available datasets. *FineVideo* [12] includes a vast collection of diverse videos with more than 40,000 videos covering a wide spectrum of categories and domains. With thousands of hours of content, FineVideo represents a rich resource for capturing varied human activities, scenes, and interactions. Additionally, we leverage the train set of *MovieClips* [46] that contains approximately 7,000 video clips, typically ranging from one to three minutes, curated from various movies. It serves as a diverse collection of cinematic content. Finally, *ActivityNet* [9], a dataset that focuses on complex human activities that are relevant to daily life, offering a broad range of action categories and scenarios. Our diverse video dataset supports training across varied domains, enhancing generalization. Since our framework is self-supervised, adding more data is easy and requires no manual labels.

Training strategy. To train our model, we employ a dynamic sampling strategy to handle videos of varying lengths. For each video, we randomly sample a duration in seconds, and this ensures that each batch contains videos of different lengths. It allows us to train on entire videos or portions of them, providing our model with exposure to both short and long video segments, which is particularly useful for handling complex temporal dependencies in long-form video understanding tasks. In practice, for simplicity, we set each short video segment to a length of five seconds. In App. 9 we explore the impact of segment length.

Training efficiency. Our pre-training method is highly efficient compared to existing approaches. This efficiency is demonstrated by the reduced number of tokens required during training. Specifically, since we leverage short-video representations extracted by off-the-shelf models, each video segment is represented by a single token. Thus, with each segment lasting five seconds, processing a two-minute video with LV-MAE requires only 24 tokens, compared to the 11,760 tokens required by other frame-level methods (60 frames $\times$ 14 $\times$ 14 patches for a standard

16 × 16 image grid) [21, 27, 31, 41, 42]. As a result, we can pre-train LV-MAE in only 20 hours on 8 × A10 GPUs while achieving state-of-the-art results on downstream tasks. Our optimized framework makes pre-training on long videos computationally feasible without compromising performance. Additional implementation details are provided in App. 7.

3.4. Interpretable Predictions

One potential limitation in training masked-embedding autoencoders, where we process sequences of embeddings rather than spatio-temporal image patches at the frame level, is that it can be challenging to directly interpret the quality of token reconstruction produced by the decoder (*e.g.* as done in [16]). To address this, we leverage the aligned subspace of language-video embeddings from the pre-trained multimodal model to interpret predictions and assess the model's ability to reconstruct embeddings with accurate semantic meaning. Specifically, we propose using retrieval for each reconstructed masked token against a large set of captions to visualize and evaluate the matches.

Constructing caption database. We begin by annotating a large set of short video segments from the LVU dataset, utilizing Claude-3.5 Sonnet [1] to automatically generate textual descriptions for each segment. Specifically, we uniformly sample 20 frames from each segment and prompt the LLM to provide a concise description that captures the key visual or semantic information in the segment. Formally, for each segment v_i, we obtain an annotation a_i. Next, we use LanguageBind's language encoder to extract textual embedding for each segment's description, $\mathbf{e}_i^\ell$.

Masked-embeddings retrieval. Given a reconstructed embeddings output by the decoder $\hat{\mathbf{e}}_j$, corresponding to the j-th video segment, we retrieve the top-matching captions with the highest response by computing the cosine similarity between $\hat{\mathbf{e}}_j$ and each caption's textual embedding, $\mathbf{e}_i^\ell$. This way, these top-matched retrieved captions can be inspected to ensure that the semantic meaning of the reconstructed masked embeddings is well captured by the decoder. Visualization examples are provided in section 4.4.

4. Experiments

We present results obtained by our method and compare them to the latest state-of-the-art (SOTA) long video understanding models such as VideoBERT [39], ViS4mer [21], Turbo [15] and VideoMamba [27]. In particular, we show that LV-MAE outperforms previous SOTA methods on downstream tasks from diverse domains such as movies, activity prediction, and procedural tasks. Additionally, we perform thorough ablation studies to assess the effectiveness of various design choices in our approach. We provide implementation details in App. 7.

4.1. Benchmarks

Long-Video Understanding (LVU). A challenging suite of tasks aimed at testing models' capabilities in understanding extended movie clips [46]. The benchmark consists of nine diverse tasks divided into three main categories: content understanding, metadata prediction, and user engagement prediction. The *content understanding* tasks involve predicting attributes like "relationship", "speaking style", and "scene/place". *Metadata prediction* tasks focus on identifying movie-related attributes such as "director", "genre", "writer", and "release year". Lastly, the *user engagement* tasks measure how well the model can predict YouTube-based metrics such as the "like ratio" and "popularity". The LVU benchmark is built from the MovieClip dataset that is comprising approximately 30,000 videos sourced from around 3,000 movies. Each video spans one to three minutes, reflecting real-world movie scenes that require both short-term and long-term temporal reasoning.

Breakfast. Video classification of cooking activities. The dataset [23] consists of 1,712 videos, covering 10 complex cooking activities and totaling 77 hours of footage.

COmprehensive INstruction video analysis (COIN). Video classification of procedural tasks depicted in videos. The COIN dataset [40] includes 11,827 videos, representing 180 distinct procedural tasks, with an average video length of 2.36 minutes.

4.2. Main Results

We evaluate our method on the aforementioned long-video understanding benchmarks, using the top-1 accuracy metric for classification tasks and mean-squared error (MSE) for regression tasks. Initially, we pre-trained our model on a large, diverse video dataset (see Sec. 3.3 for details). To obtain short-video representations, we experimented with either LanguageBind [55] or InternVideo2 [44], keeping their weights frozen throughout pre-training. After pre-training, we discard the decoder and use the frozen encoder to extract the long-video representation $\mathcal{Z}$. We apply attentive probing (AP) [24, 53] for classification tasks by tuning a single transformer encoder layer with a linear projection head (details are provided in the appendix), while for regression tasks in LVU, we use a regression model on the global average pooling of the latent representation.

We present our results in Table 1 and Table 2. As observed, LV-MAE with LanguageBind embeddings significantly outperforms existing approaches. Specifically, on the LVU benchmark, our model achieves superior performance in seven out of nine tasks across all categories, with an average accuracy improvement of 5.3% on the classification tasks. Notably, LV-MAE achieves improvements exceeding 10% on certain tasks, underscoring the effectiveness of our SSL framework. On the COIN dataset, LV-MAE outperforms all existing methods using both InternVideo2

Table 1. **LVU benchmark results.** We compare the Top-1 accuracy results obtained by LV-MAE and other methods. "FB" (Frozen Backbone) indicates that only a small subset of parameters is fine-tuned, in contrast to other methods that tune the entire model. "Dir." and "Rel." refer to "Director" and "Relationship," respectively. LanguageBind/InternVideo2-Baseline refers to using the raw embeddings.

Method	FB	Metadata ↑				Content ↑			Avg.	User ↓	
		Dir.	Genre	Writer	Year	Scene	Speak	Rel.		Likes	Views
VideoBERT [39]	✗	47.30	51.90	38.50	36.10	54.90	37.90	52.80	45.6	0.32	4.46
Object Transformer [46]	✗	51.20	54.60	34.50	39.10	56.90	39.40	53.10	46.9	0.23	3.55
LST [21]	✗	56.07	52.70	42.26	39.16	62.79	37.31	52.38	48.9	0.31	3.83
Performer [21]	✗	58.87	49.45	48.21	41.25	60.46	38.80	50.00	49.6	0.31	3.93
Orthoformer [21]	✗	55.14	55.79	47.02	43.35	66.27	39.30	50.00	50.9	0.29	3.86
ViS4mer [21]	✗	62.61	54.71	48.80	44.75	67.44	40.79	57.14	53.7	0.26	3.63
VideoMamba [27]	✗	67.29	65.24	52.98	48.23	70.37	40.43	**62.50**	58.1	0.26	2.90
LanguageBind-Transformer	✗	24.30	57.88	16.07	18.44	35.80	32.45	51.22	33.7	0.56	3.04
LanguageBind-Mamba	✗	61.68	70.38	51.78	56.74	74.04	38.83	43.90	56.8	0.30	2.98
InternVideo2-Baseline [44]	✓	45.79	54.79	5.36	8.51	34.57	30.32	51.22	32.9	0.24	3.22
LanguageBind-Baseline [55]	✓	64.48	61.47	5.95	29.08	30.86	31.91	48.78	38.9	**0.21**	2.90
LV-MAE$_{\text{InternVideo2}}$(Ours)	✓	62.62	68.15	57.14	**58.15**	**77.78**	39.89	53.66	59.6	0.23	2.68
LV-MAE$_{\text{LanguageBind}}$(Ours)	✓	**77.57**	**71.57**	**64.28**	**58.15**	72.84	**40.95**	58.53	**63.4**	0.23	**2.52**

Table 2. **Breakfast and COIN benchmark results**. We compare the Top-1 accuracy results obtained by LV-MAE and other methods. "FB" (Frozen Backbone) indicates that only a small subset of parameters is fine-tuned, in contrast to other methods that tune the entire model. Baseline refers to using the raw embeddings.

Method	FB	Breakfast Top-1	COIN Top-1
Timeception [19]	✗	71.3	-
VideoGraph [20]	✗	69.5	-
GHRM [54]	✗	75.5	-
Distant Supervision [31]	✗	89.9	90.0
ViS4mer [21]	✗	88.2	88.4
Turbo [15]	✗	91.3	87.5
VideoMamba [27]	✗	**96.9**	90.4
InternVideo2-Baseline [44]	✓	83.94	88.41
LanguageBind-Baseline [55]	✓	73.52	91.13
LV-MAE$_{\text{InternVideo2}}$ (Ours)	✓	93.24	**92.72**
LV-MAE$_{\text{LanguageBind}}$ (Ours)	✓	91.55	92.42

and LanguageBind embeddings. Finally, on the Breakfast dataset, LV-MAE achieves the second-best performance.

We note that previous works fine-tune all model weights for each benchmark, whereas in our approach, only a small number of parameters are trained through attentive probing. Nevertheless, LV-MAE achieves superior performance on most tasks. Furthermore, previous works typically pre-train their models on short-video datasets, such as Kinetics [22] (e.g., ViS4mer [21] and VideoMamba [27]). Our approach enables label-free pre-training on long videos by leveraging short-video segments, capturing long-range dependencies and robust temporal patterns.

To further validate the effectiveness of the representations learned by our masked-embedding autoencoder, we compare LV-MAE to a baseline method that applies attentive probing directly to the raw embeddings of either LanguageBind or InternVideo2 (referred to as LanguageBind-Baseline and InternVideo2-Baseline in Table 1 and Table 2). The results show that while raw embeddings yield reasonable results for certain tasks, our masked-embedding autoencoder consistently produces superior performance, highlighting its effectiveness in capturing high-level context from long videos beyond the raw embedding level. Notably, on some tasks, using raw InternVideo2 or LanguageBind embeddings leads to poor accuracy (e.g., for the "Writer," "Year," and "Scene" tasks). However, pre-training with these embeddings following the LV-MAE scheme significantly improves accuracy. For instance, on the "Writer" task, LV-MAE with LanguageBind improves accuracy from 5.95% to 64.28%. Additionally, we conducted end-to-end training experiments using both transformer and Mamba models on LanguageBind embeddings. Results on the LVU benchmark are presented in Table 1. These experiments demonstrate that our self-supervised MAE approach outperforms direct end-to-end training from sequences of short video-clip embeddings.

4.3. Main Properties

We evaluate the influence of various components in our method. Specifically, we investigate the use of linear probing (LP) versus attentive probing (AP) as classification heads, examine the effectiveness of our proposed semantic masking strategy compared to random masking, test different masking ratios, and compare results obtained with various model sizes (encoder depth).

Attentive probing vs. linear probing. In Table 3, we compare LP and AP using either InternVideo2 or LanguageBind embeddings on the LVU benchmark, Breakfast, and COIN datasets. For the LVU benchmark, we report the average score for classification tasks, with full results available

Table 3. **Masking strategy for linear and attentive probing**. Top-1 accuracy comparison for different masking strategies using InternVideo2 or LanguageBind evaluated on LVU (Avg.), Breakfast (BF), and COIN. Default settings are marked in gray.

Masking	Embedding	LVU	BF	COIN
Linear Probing				
Semantic	LanguageBind	**59.02**	81.97	91.54
Random	LanguageBind	58.32	81.69	91.46
Semantic	InternVideo2	52.60	**85.92**	**91.63**
Random	InternVideo2	53.63	85.63	91.29
Attentive Probing				
Semantic	LanguageBind	**63.41**	90.70	91.8
Random	LanguageBind	62.54	91.55	92.42
Semantic	InternVideo2	59.63	**93.24**	92.72
Random	InternVideo2	58.43	92.68	**92.88**

in App. 8. Our findings show that attentive probing consistently outperforms linear probing. This is likely because AP's additional attention mechanism enables it to capture task-specific context more effectively, whereas LP averages information across the sequence, potentially oversmoothing important details. For instance, on the Breakfast dataset, attentive probing proves especially useful in distinguishing between cooking activities that share similar steps but differ in subtle nuances. When using linear probing, averaging the latent representations may smooth out important distinctions between actions, such as cracking eggs directly into a pan versus stirring them first before cooking. These minor variations in the long video are critical in tasks like identifying scrambled versus fried eggs, where actions overlap but diverge in essential details. Attentive probing preserves these fine-grained nuances across the sequence, making it more effective for such distinctions.

Masking strategy. We compare our proposed semantic masking strategy against the random masking strategy as employed by the standrad MAE model. As shown in Table 3, semantic masking consistently performs slightly better than random masking, suggesting that selectively masking more semantically relevant embeddings enhances model performance. We observe that both masking strategies are simple yet effective, achieving state-of-the-art results.

Masking ratio. Additionally, we explore various masking ratios and find that moderate masking ratios yield the best results, with 40% for random masking and 50% for semantic masking producing the highest performance (Figure 2). Finally, we assess the effect of model size on performance.

Model size. Table 4 reports accuracy results for different numbers of transformer encoder blocks. As shown, increasing the number of encoder blocks improves accuracy in nearly all tested cases, indicating that deeper encoders lead to stronger performance.

4.4. Interpretable Predictions Experiment

The ability to understand model predictions is essential to facilitate human understanding of what the model learns.

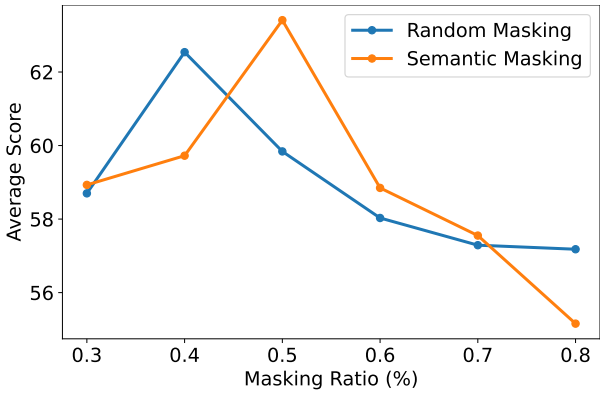

Figure 2. **Masking ratio**. The y-axes are LVU average accuracy scores. A moderate masking ratio of $40-50\%$ works well for attentive probing.

Table 4. **Encoder depth**. A deep encoder can improve accuracy. Here, we use attentive probing and Top 1 accuracy. We evaluated on LVU (Avg.), Breakfast (BF), and COIN datasets. Default settings are marked in gray.

Depth	LVU	BF	COIN
16	59.73	87.04	92.47
24	60.04	89.58	**93.26**
32	**63.41**	**90.70**	91.8

In Sec. 3.4 we propose an interpretability technique for our framework. In this section, we show its effectiveness and verify the reconstructions produced by the decoder. In practice, we used the MovieClips dataset to extract captions. For each short-video segment, v_i, in the dataset, we generate concise annotations, a_i, using the Claude Sonnet 3.5 LLM [1]. Next, we sample a long video from the dataset and extract its corresponding embeddings, $\mathcal{E}$, masking 40% of them. Our model is fed with the visible embeddings and tasked with reconstructing all embeddings, resulting in the predicted set, $\hat{\mathcal{E}}$. Each annotation is transformed into a language embedding using LanguageBind, which has the same embedding space as $\mathcal{E}$. Using the cosine similarity, we perform retrieval by selecting the most similar textual embeddings. We conduct this retrieval for both the original embeddings, $\mathcal{E}$, and the predicted embeddings, $\hat{\mathcal{E}}$. We provide a scheme of the process in Figure 4. The original model retrieval accuracy (R@5) is 75.34%.

For concise visualization of the results, in Fig. 3, we subsample short video segments and present the interpreted model's predictions. We observe that in most cases, our model predicts embeddings that are semantically close to the original ones. This means that the predicted embeddings align with either the same or semantically similar enough textual embedding. As expected, we observe that the retrieved textual embeddings tend to be more abstract or high-level, rather than precise descriptions of the scene. For instance, in one example, the ground truth annotation is "Blonde woman sips drink by the sea", while the retrieved

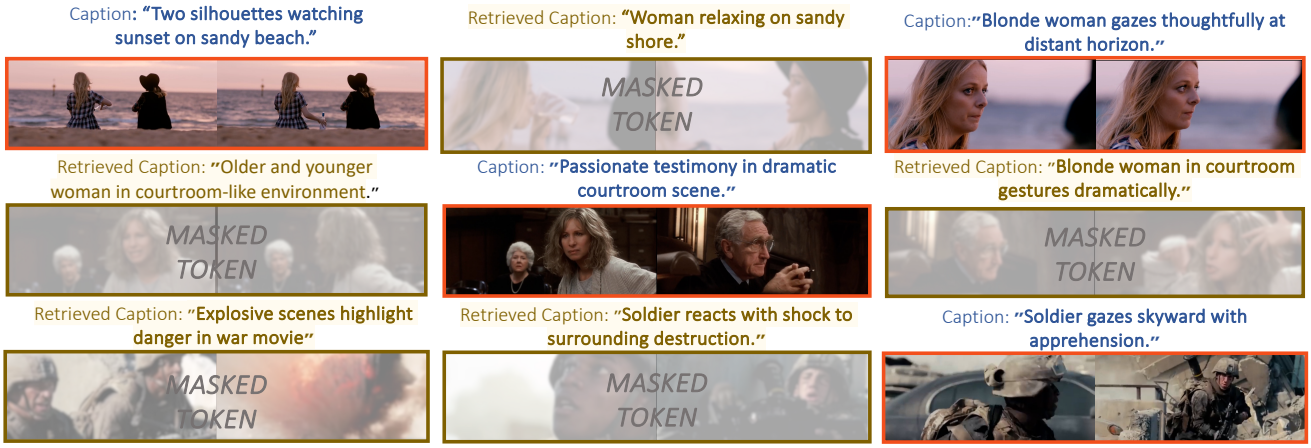

Figure 3. **Interpretable predictions – examples**: Each row visualizes three consecutive five-second segments. Above each segment, we show the original caption for the visible tokens and the retrieved caption for the reconstructed masked tokens. As shown, the model successfully reconstructs the semantic meaning of the masked embeddings, offering insight into the model's effectiveness and capabilities.

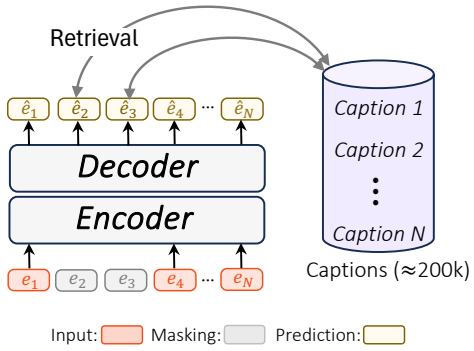

Figure 4. **Interpretable predictions – scheme**: For each reconstructed masked embedding, we perform retrieval against a large set of captions collected from the MovieClip dataset. The top matches can then be used to assess the model's quality.

annotation according to the prediction is "Woman relaxing on sandy shore." This behavior is akin to what has been observed with MAE models in the image domain [16], where predicted image patches are often blurry and lack fine details. Similarly, our model's predictions can be seen as abstract representations, capturing the essence of a scene but missing some specific details.

5. Discussion and Limitations

Our method achieves state-of-the-art results across diverse tasks and benchmarks, yet there are several limitations to consider. First, our approach relies on leveraging low-level representations from pre-trained, off-the-shelf models. Since these representations are frozen during our training, our model's performance is inherently constrained by the quality of these pre-trained embeddings. If the pre-trained model fails to effectively capture the content of short video clips, this deficiency may propagate into our long-range understanding framework, affecting overall representation quality. However, by choosing high-performing video-text alignment models such as LanguageBind, we mitigate this limitation and ensure strong baseline representations.

Second, unlike methods operating on image patches, where masked token predictions can be easily visualized, our approach operates on embeddings, which are inherently more abstract and less interpretable. This makes it challenging to directly assess the model's performance on masked token predictions. To address this, we propose an interpretability strategy that leverages captions or generates them when unavailable, providing a proxy for visualizing and understanding the model's predictions. This approach aids in interpreting predictions, though it remains an approximation rather than a direct visualization of learned embeddings.

6. Conclusions

While new video understanding methods are emerging rapidly, most are confined to short clips and operate at the frame level, limiting their scalability to longer content. In this work, we introduce a novel self-supervised representation learning model for long-video understanding. Our method builds on the well-established masked autoencoder design, utilizes high-quality embeddings from pre-trained short-video models, and efficiently handles videos ranging from a few seconds to potentially several hours. This allows us to capture richer temporal dynamics and semantic structures over extended durations. Empirical results show that our model consistently outperforms or matches state-of-the-art approaches on long-video tasks. Looking ahead, our framework offers strong potential for broader applications such as generation, retrieval, and comprehensive long-form analysis.

References

[1] Anthropic. Introducing claude 3.5 sonnet. *Anthropic News*, 2024. 5, 7

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021. 3

[3] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEit: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022. 3

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 3

[5] Sihan Chen, Handong Li, Qunbo Wang, Zijia Zhao, Ming-Ting Sun, Xinxin Zhu, and J. Liu. Vast: A vision-audio-subtitle-text omni-modality foundation model and dataset. *ArXiv*, abs/2305.18500, 2023. 1

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 1597–1607. PMLR, 2020. 3

[7] Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024. 3

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. 3, 4, 1

[9] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia 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. 4

[10] David Fan, Jue Wang, Shuai Liao, Yi Zhu, Vimal Bhat, Hector J. Santos-Villalobos, M. V. Rohith, and Xinyu Li. Motion-guided masking for spatiotemporal representation learning. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5596–5606, 2023. 4

[11] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6824–6835, 2021. 3

[12] Miquel Farré, Andi Marafioti, Lewis Tunstall, Leandro Von Werra, and Thomas Wolf. Finevideo. https://huggingface.co/datasets/HuggingFaceFV/finevideo, 2024. 4

[13] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Steffen Cavalier, Sylvain Gelly, Remi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020. 3

[14] Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. 1, 3

[15] Tengda Han, Weidi Xie, and Andrew Zisserman. Turbo training with token dropout. In *British Machine Vision Conference*, 2022. 5, 6

[16] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 2, 3, 5, 8, 1

[17] Bingkun Huang, Zhiyu Zhao, Guozhen Zhang, Y. Qiao, and Limin Wang. Mgmae: Motion guided masking for video masked autoencoding. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13447–13458, 2023. 4

[18] De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin, Pavlo Molchanov, Zhiding Yu, and Jan Kautz. Lita: Language instructed temporal-localization assistant. In *European Conference on Computer Vision*, pages 202–218. Springer, 2024. 3

[19] Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Timeception for complex action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 254–263, 2019. 6

[20] Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Videograph: Recognizing minutes-long human activities in videos. *arXiv preprint arXiv:1905.05143*, 2019. 6

[21] Md Mohaiminul Islam and Gedas Bertasius. Long movie clip classification with state-space video models. In *European Conference on Computer Vision*, pages 87–104. Springer, 2022. 1, 2, 3, 5, 6

[22] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Apostol Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *ArXiv*, abs/1705.06950, 2017. 6, 3

[23] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 780–787, 2014. 2, 3, 5

[24] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pages 3744–3753. PMLR, 2019. 5, 2

[25] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7331–7341, 2021. 3

[26] Gang Li, Heliang Zheng, Daqing Liu, Bing Su, and Changwen Zheng. Semmae: Semantic-guided masking for learning masked autoencoders. *ArXiv*, abs/2206.10207, 2022. 4

[27] Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. Videomamba: State space model for efficient video understanding, 2024. 1, 2, 3, 5, 6

[28] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, 2023. 1

[29] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pages 323–340. Springer, 2024. 3

[30] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *ArXiv*, abs/2311.10122, 2023. 1

[31] Xudong Lin, Fabio Petroni, Gedas Bertasius, Marcus Rohrbach, Shih-Fu Chang, and Lorenzo Torresani. Learning to recognize procedural activities with distant supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13853–13863, 2022. 5, 6

[32] 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. *Neurocomputing*, 508:293–304, 2021. 1

[33] 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. 3

[34] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. Available at: https://arxiv.org/abs/1301.3781. 3

[35] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *OpenAI preprint*, 2018. 3

[36] 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. 3

[37] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14313–14323, 2024. 3

[38] Yan Shu, Peitian Zhang, Zheng Liu, Minghao Qin, Junjie Zhou, Tiejun Huang, and Bo Zhao. Video-xl: Extra-long vision language model for hour-scale video understanding. *arXiv preprint arXiv:2409.14485*, 2024. 3

[39] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7464–7473, 2019. 5, 6

[40] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. COIN: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1207–1216, 2019. 2, 3, 5

[41] 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. 3, 5

[42] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14549–14560, 2023. 3, 5

[43] Xiao Wang, Jingen Liu, Tao Mei, and Jiebo Luo. Coseg: Cognitively inspired unsupervised generic event segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):12507–12517, 2023. 3

[44] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, Yansong Shi, Tianxiang Jiang, Songze Li, Hongjie Zhang, Yifei Huang, Yu Qiao, Yali Wang, and Limin Wang. Internvideo2: Scaling video foundation models for multimodal video understanding. *ArXiv*, abs/2403.15377, 2024. 1, 2, 3, 5, 6

[45] Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. Longvlm: Efficient long video understanding via large language models. In *European Conference on Computer Vision*, pages 453–470. Springer, 2024. 3

[46] Chao-Yuan Wu and Philipp Krahenbuhl. Towards long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1884–1894, 2021. 2, 3, 4, 5, 6

[47] Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. Pllava : Parameter-free llava extension from images to videos for video dense captioning. *ArXiv*, abs/2404.16994, 2024. 1

[48] Mingze Xu, Mingfei Gao, Zhe Gan, Hong-You Chen, Zhengfeng Lai, Haiming Gang, Kai Kang, and Afshin Dehghan. Slowfast-llava: A strong training-free baseline for video large language models. *arXiv preprint arXiv:2407.15841*, 2024. 3

[49] Hongwei Xue, Yuchong Sun, Bei Liu, Jianlong Fu, Rui Song, Houqiang Li, and Jiebo Luo. Clip-vip: Adapting pre-trained image-text model to video-language alignment. In *International Conference on Learning Representations*, 2022. 1

[50] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *Proceedings of*

the IEEE/CVF international conference on computer vision, pages 1686–1697, 2021. 3

[51] Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10714–10726, 2023. 1

[52] Xubing Ye, Yukang Gan, Xiaoke Huang, Yixiao Ge, Ying Shan, and Yansong Tang. Voco-llama: Towards vision compression with large language models. *arXiv preprint arXiv:2406.12275*, 2024. 3

[53] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*, 2022. 2, 5

[54] Jiaming Zhou, Kun-Yu Lin, Haoxin Li, and Wei-Shi Zheng. Graph-based high-order relation modeling for long-term action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8984–8993, 2021. 6

[55] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, WANG HongFa, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, Cai Wan Zhang, Zhifeng Li, Wei Liu, and Li Yuan. LanguageBind: Extending video-language pretraining to n-modality by language-based semantic alignment. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 2, 3, 5, 6

[56] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *ArXiv*, abs/2401.09417, 2024. 1, 3

LV-MAE: Learning Long Video Representations through Masked-Embedding Autoencoders

Supplementary Material

7. Implementation Details

Short-video segmentation. To optimize training time, we pre-process all data once prior to pre-training. Specifically, for each dataset, the process outlined in Sec. 3.1 is applied only once. This ensures efficient handling of video inputs during the training phase.

Architectural design. We adopt an asymmetric encoder-decoder architecture inspired by [16]. The encoder processes only the visible, unmasked tokens from the video input. Positional embeddings are added to each token to encode temporal relationships. The embedded tokens are then passed through a Transformer encoder with K layers, where each layer includes a multi-head self-attention mechanism, a multi-layer perceptron (MLP), and LayerNorm. Unlike ViT architectures designed for fixed-size inputs, such as images, our encoder accommodates varying input lengths, enabling it to handle videos of arbitrary durations. Mask tokens are not used during this stage, ensuring computational efficiency by focusing exclusively on visible tokens.

After the encoder processes the visible tokens, the decoder reconstructs the full sequence by introducing shared, learned mask tokens to replace the missing inputs. These mask tokens are inserted at their respective positions in the sequence. To ensure that the mask tokens carry information about their temporal location, positional embeddings are applied to all tokens in the decoder input, including the mask tokens. Without these positional embeddings, the mask tokens would have no information about their location in the video. The decoder processes the full sequence through a series of Transformer layers. Following [16], it is designed to be shallower than the encoder to maintain computational efficiency while providing accurate reconstructions.

To manage videos of varying lengths, we adapt sequence-processing techniques from BERT [8]. Each sequence of short-video embeddings is capped at 256 tokens, with shorter sequences padded using a special `[PAD]` token. During the self-attention process, an attention mask prevents these padding tokens from influencing training. This design ensures that the model maintains computational efficiency while supporting inputs of arbitrary length. Practically, our framework allows for increasing the maximum token limit, enabling the processing of even longer videos. However, since the benchmark datasets used in this study do not exceed 20 minutes, we leave such extensions for future exploration.

Table 5. **Pre-training setting.**

Config	Value
optimizer	AdamW
base learning rate	1.5e-4
weight decay	0.05
optimizer momentum	β_1, β_2=0.9, 0.95
batch size	16
learning rate schedule	cosine decay
warmup epochs	40
epochs	150
number of tokens	256
short video length	5 sec

Table 6. **Linear probing setting.**

Config	Value
optimizer	Adam
base learning rate	1e-4
optimizer momentum	β_1, β_2=0.9, 0.999
batch size	16
epochs	30

Table 7. **Attentive probing setting.**

Config	Value
optimizer	Adam
learning rate	1e-3
learning rate schedule	ExponentialLR, γ=0.9
optimizer momentum	β_1, β_2=0.9, 0.999
batch size	16
epochs	20

Finally, during pre-training, an auxiliary `[CLS]` token is appended to the input sequence in the encoder. This token serves as a representation of the entire video sequence and is specifically used for downstream tasks, such as classification. In attentive probing fine-tuning, this `[CLS]` token is adapted to generate task-specific predictions.

Pre-training hyperparameters. Our pre-training hyperparameters are detailed in Table 5. We closely follow the hyperparameter settings from [16] with a few adjustments. Specifically, we use a smaller batch size and reduce the number of epochs during pre-training. We set the limit for the number of tokens to 256. Finally, we set the short video segment length to five seconds. The choice of a five-second segment length balances two considerations: it is

Table 8. **Additional LVU benchmark results.** In the main paper, we provided the average Top-1 accuracy results obtained by LV-MAE with linear probing (LP) and attentive probing (AP) with random masking. Here, we provide extended results per task. Masking indicates the masking technique that was applied. "Dir." and "Rel." refer to "Director" and "Relationship," respectively.

Method	Masking	Metadata ↑				Content ↑			Avg.
		Dir.	Genre	Writer	Year	Scene	Speak	Rel.	
LV-MAE$_{\text{InternVideo2}}$(LP)	Semantic	55.14	58.56	49.40	43.26	72.84	37.77	51.22	52.60
LV-MAE$_{\text{InternVideo2}}$(LP)	Random	57.01	57.36	52.38	49.65	70.37	39.89	48.78	53.63
LV-MAE$_{\text{InternVideo2}}$(AP)	Random	54.21	66.95	55.36	56.03	80.25	42.55	53.66	58.43
LV-MAE$_{\text{LanguageBind}}$(LP)	Semantic	71.03	67.47	54.76	53.90	67.90	42.02	56.09	59.02
LV-MAE$_{\text{LanguageBind}}$(LP)	Random	71.96	67.12	55.36	53.19	71.60	37.76	51.22	58.32
LV-MAE$_{\text{LanguageBind}}$(AP)	Random	78.50	69.17	60.12	61.70	72.84	39.36	56.09	62.54

short enough to capture low-level spatiotemporal patterns effectively using an off-the-shelf frozen model and sufficiently long to reduce the number of tokens required for longer videos. In future work, we plan to analyze the impact of segment lengths (e.g., 10–15 seconds) on performance, as longer segments could offer greater efficiency for processing extended videos. However, this may risk reduced performance from the frozen model due to challenges in extracting representations from longer, more complex segments.

8. Training on Downstream Tasks

We experiment with two approaches to train our frozen model to solve downstream tasks utilizing its latent representations.

Linear probing. For linear probing, we append a simple linear layer to the encoder. This layer operates on the global average pooling of the latent representations $\mathcal{Z}$. The linear layer is optimized with cross-entropy loss. We report the optimizer and other hyperparameters we use in Table 6.

Attentive probing. To implement attentive probing [24, 53], we add a lightweight transformer encoder layer to the pre-trained model. This additional block consists of a multi-head self-attention mechanism, an MLP, and LayerNorm. The layer is fine-tuned exclusively on task-specific datasets, focusing on adapting the `[CLS]` token to generate final predictions through a linear classifier optimized with cross-entropy loss. This approach minimizes computational overhead while effectively tailoring the model for specific tasks. We report the optimizer and other hyperparameters we use in Table 7.

9. Extended LVU ablation

In Table 8, we provide full results of the LVU benchmark for the reported average classification score from Table 3.

Impact of Segment Length. In the main paper, we report results obtained by partitioning each long video into *five* second clips. To assess the sensitivity of our method to this design choice, we also trained models on longer fixed-length clips (10 seconds and 15 seconds) as well as on variable-length, shot-based segments. Table 9 summarizes the average performance across all LVU downstream tasks. The *five* second configuration remains the most effective, achieving an average score of 63.40 and consistently outperforming the longer and shot-based alternatives. Finally, we examined the use of *overlapping five* second clips; this introduces redundant context and reduces the average score by 4.5, from 63.40 to 58.90.

Table 9. **Impact of Segment Length.** The average Top-1 accuracy results obtained by LV-MAE on the LVU benchmark using different segment lengths.

5 seconds	10 seconds	15 seconds	shots
63.40	61.43	61.05	62.17

Architecture and Clip-Length Ablations. Our performance gains stem from both the two-stage architecture and long-video training capability. While prior works are limited to ∼60 frames, our method can process much longer sequences. To highlight the importance of this capability, we cap the training clips at *five minutes*, which lowers the average LVU score to 55.58 %. Comprehensive ablations, removing the MAE stage or replacing it with plain Transformer or Mamba backbones, are reported in Table 10. The results demonstrate that *both* the MAE formulation and exposure to extended temporal context are critical for achieving state-of-the-art performance.

Impact of Input Frame Count. To quantify the role of temporal coverage, we trained models with varying numbers of input frames; the resulting trend is depicted in Fig. 5.

Table 10. Results on the LVU benchmark using different architectures and clip-length ablations.

Method	FB	Metadata ↑				Content ↑			Avg.
		Dir.	Genre	Writer	Year	Scene	Speak	Rel.	
ViS4mer	✗	62.61	54.71	48.80	44.75	67.44	40.79	57.14	53.7
VideoMamba	✗	67.29	65.24	52.98	48.23	70.37	40.43	**62.50**	58.1
LanguageBind-Transformer	✗	24.30	57.88	16.07	18.44	35.80	32.45	51.22	33.7
LanguageBind-Mamba	✗	61.68	70.38	51.78	56.74	**74.04**	38.83	43.90	56.8
LV-MAE(frame-MAE feature extractor)	✓	64.49	62.50	49.4	48.23	67.90	36.70	51.22	54.3
LV-MAE$_{\text{LanguageBind}}$(Ours)	✓	**77.57**	**71.57**	**64.28**	**58.15**	72.84	**40.95**	58.53	**63.4**

Performance rises monotonically as the frame count increases, confirming that a broader temporal window enables the model to capture motion cues more effectively. We extract 5-second clip embeddings with clip count varying by video length. All methods use identical input resolution, but ours processes significantly more frames than prior work (∼60 frames limit) efficiently, which is a key contribution enabling better temporal understanding.

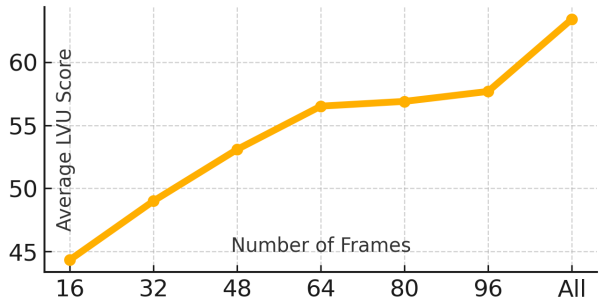

Figure 5. Average performance of LVU benchmark rises monotonically as the frame count increases.

10. Short-Video Performance

Our approach preserves performance on short-video understanding tasks. Specifically, we preserve the accuracy of LanguageBind on Kinetics-400 (600) [22]. We achieved 78.2% on K400 and 79.1% on K600 compared to 77.6% and 79.5% in LanguageBind.

11. Additional Interpretable Predictions Results

In Fig. 6, we attach additional examples of the interpretable predictions experiment from Sec. 4.4.

12. Additional Related Work

Short-video understanding methods. Numerous models have been proposed for short-video understanding, achieving remarkable performance on tasks such as action recognition [2], video classification [11], text-video retrieval [33], video captioning [50], and video question answering [25]. Multimodal models like CLIP [36], InternVideo [44], and LanguageBind [55] have demonstrated strong capabilities in aligning video and language representations. In this work, we leverage the output embedding of these models to learn long-video representations.

Long-video language understanding methods. Several recent works have advanced the field of video understanding using large language models. Video-XL [38] and LongVLM [45] both tackle the challenge of processing long-form videos, with Video-XL focusing on hour-scale video understanding and LongVLM proposing efficient mechanisms for extended video content. LongVILA [7] further contributes to this direction by scaling visual language models for long video comprehension. In the domain of efficient video processing, VoCo-LLaMA [52] explores video compression using large language models, while LLaMA-VID [29] proposes a compact two-token representation for video content. SlowFast-LLaVA [48] provides a training-free baseline approach for video large language models. Addressing temporal aspects, TimeChat [37] develops time-sensitive capabilities for video understanding, while LITA [18] focuses on precise temporal localization within videos. In contrast to these works that focus on video-language integration, we explore long-video masked-embedding autoencoders in long-form video classification tasks and aim to find effective representation learning methods specifically designed for long-form videos.

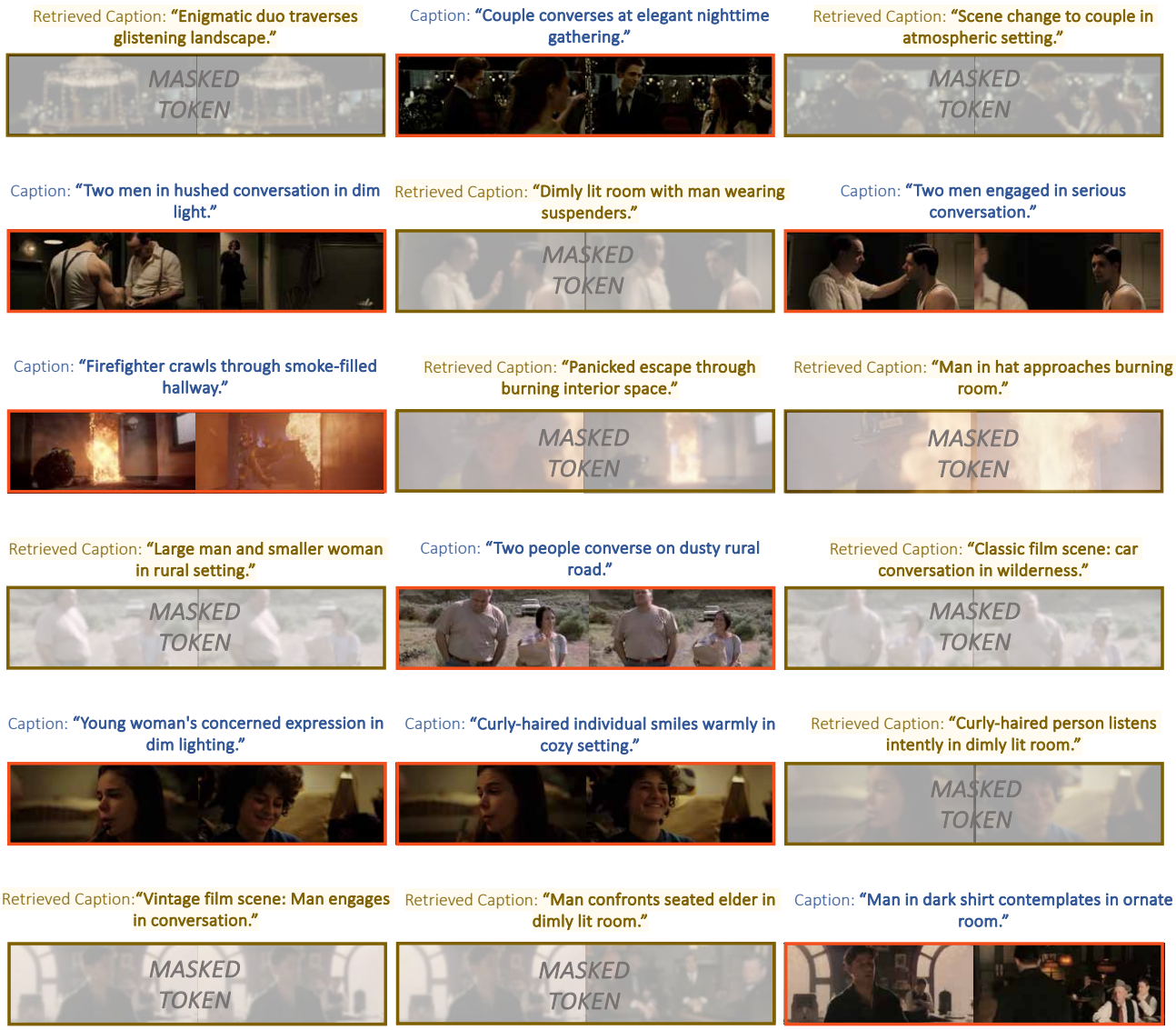

Figure 6. **Interpretable predictions – additional examples**: Each row visualizes three consecutive five-second segments. Above each segment, we show the original caption for the visible tokens and the retrieved caption for the reconstructed masked tokens. As shown, the model successfully reconstructs the semantic meaning of the masked embeddings, offering insight into the model's effectiveness and capabilities.