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	<u>40.79</u>	57.14	53.7
VideoMamba	✗	<u>67.29</u>	65.24	<u>52.98</u>	48.23	70.37	40.43	**62.50**	<u>58.1</u>
LanguageBind-Transformer	✗	24.30	57.88	16.07	18.44	35.80	32.45	51.22	33.7
LanguageBind-Mamba	✗	61.68	<u>70.38</u>	51.78	<u>56.74</u>	**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**	<u>72.84</u>	**40.95**	<u>58.53</u>	**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.

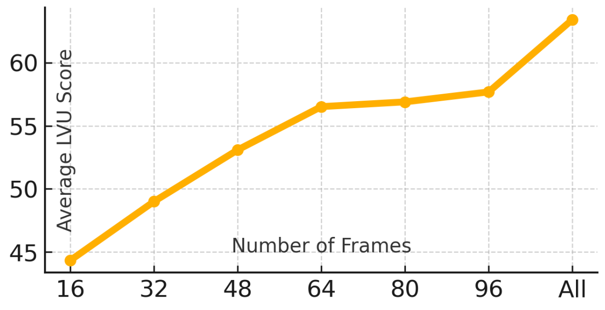

Figure 5. Average performance of LVU benchmark rises monotonically as the frame count increases.

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.

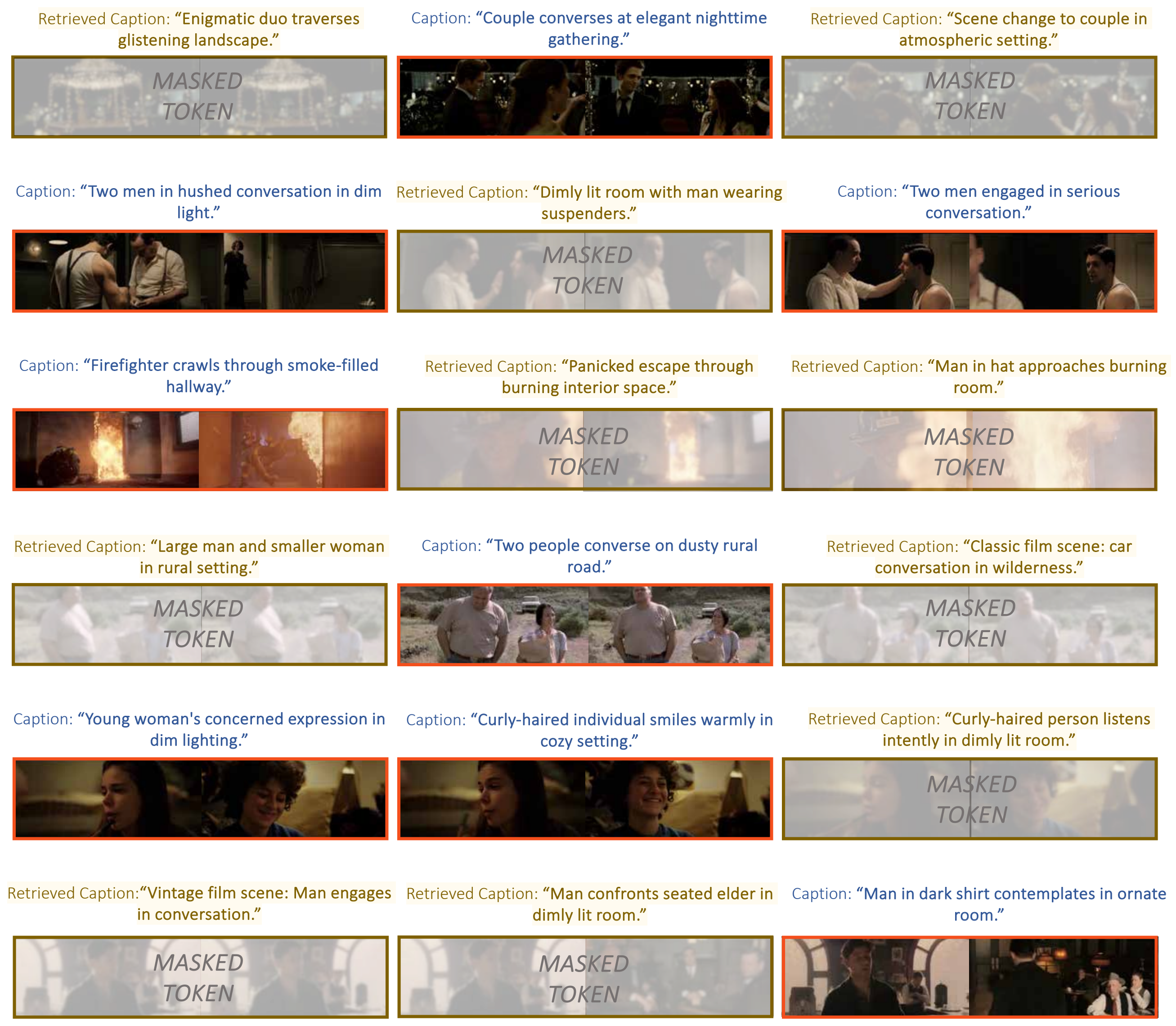
Retrieved Caption: "Enigmatic duo traverses glistening landscape."
Caption: "Couple converses at elegant nighttime gathering."
Retrieved Caption: "Scene change to couple in atmospheric setting."
MASKED TOKEN
Caption: "Two men in hushed conversation in dim light."
Retrieved Caption: "Dimly lit room with man wearing suspenders."
Caption: "Two men engaged in serious conversation."
Caption: "Firefighter crawls through smoke-filled hallway."
Retrieved Caption: "Panicked escape through burning interior space."
Retrieved Caption: "Man in hat approaches burning room."
Retrieved Caption: "Large man and smaller woman in rural setting."
Caption: "Two people converse on dusty rural road."
Retrieved Caption: "Classic film scene: car conversation in wilderness."
Caption: "Young woman's concerned expression in dim lighting."
Caption: "Curly-haired individual smiles warmly in cozy setting."
Retrieved Caption: "Curly-haired person listens intently in dimly lit room."
Retrieved Caption:"Vintage film scene: Man engages in conversation."
Retrieved Caption: "Man confronts seated elder in dimly lit room."
Caption: "Man in dark shirt contemplates in ornate room."

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.