MUSE: Mamba is Efficient Multi-scale Learner for Text-video Retrieval

Haoran Tang[1,2], **Meng Cao**[1], **Jinfa Huang**[1], **Ruyang Liu**[1,2], **Peng Jin**[1,2], **Ge Li**[1*], **Xiaodan Liang**[3]

[1] Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology, Shenzhen Graduate School, Peking University
[2] Peng Cheng Laboratory
[3] Sun Yat-sen University

Abstract

Text-Video Retrieval (TVR) aims to align and associate relevant video content with corresponding natural language queries. Most existing TVR methods are based on large-scale pre-trained vision-language models (e.g., CLIP). However, due to CLIP's inherent *plain* structure, few TVR methods explore the multi-scale representations which offer richer contextual information for a more thorough understanding. To this end, we propose MUSE, a multi-scale mamba with linear computational complexity for efficient cross-resolution modeling. Specifically, the multi-scale representations are generated by applying a feature pyramid on the last single-scale feature map. Then, we employ the Mamba structure as an efficient multi-scale learner to *jointly* learn scale-wise representations. Furthermore, we conduct comprehensive studies to investigate different model structures and designs. Extensive results on three popular benchmarks have validated the superiority of MUSE.

Code — https://github.com/hrtang22/MUSE
Extended version — https://arxiv.org/abs/2408.10575

Introduction

Text-Video Retrieval (TVR) (Gabeur et al. 2020; Gorti et al. 2022; He et al. 2021; Lei et al. 2021; Luo et al. 2022; Ma et al. 2022; Wang et al. 2022) is a fundamental task in multimodal research. Its objective is to locate the most relevant video content within a repository in response to a text query and vice versa.

Based on large-scale image-text pre-trained model CLIP (Radford et al. 2021), most current TVR methods focus on transferring CLIP to the video-text domain. To achieve fine-grained representations, mainstream methods capture cross-modal alignment at different granularities, including video-sentence (Ma et al. 2022), frame-sentence (Gorti et al. 2022; Ma et al. 2022) or even patch-word (Wang et al. 2023) levels.

However, CLIP is inherently a *plain* structure with the identical token length for all the layers. Therefore, these methods ignore the exploration of representations of different scales, which provides more valuable contextual information for comprehensive understanding. For example, in Figure 1(a), the textual query aims to retrieve a video where "people are carrying `torches` and chasing a giant squidward". As shown, the most discriminative object `torch` is not highlighted in the original resolution[1]. The loss of such detailed yet important information leads to incorrect retrieval results. In contrast, we can see that the torch region is correctly highlighted when focusing on higher-resolution representations. Therefore, information hidden in higher resolution should be considered for TVR.

In light of this, two issues naturally arise. 1) how to *generate* multi-scale representations? Since the vanilla CLIP architecture is non-hierarchical, it maintains a single-scale feature map. Following the spirit of "fewer inductive biases" proposed in (Li et al. 2022), we build a feature pyramid from the last single-scale feature map via convolution or pooling operations. Compared to the ConvNet-based methods, *e.g.*, Swin Transformers (Liu et al. 2021), such design does not require the introduction of additional modules and is, therefore, more efficient; 2) how to efficiently model *cross-resolution* correlations? One intuitive idea is to jointly model various resolutions in a holistic manner, *i.e.*, flattening resolution-wise representations and modeling the comprehensive correlations with the widely-used attention mechanism (Vaswani et al. 2017). This strategy inevitably introduces a huge amount of computation, which is quadratically correlated with the length of the sequence (e.g., Transformer requires 36.8GB GPU memory when the input frame is 12, as shown in Figure 1(b)). To this end, we argue that **M**amba is an efficient m**U**lti-**S**cal**E** learner (dubbed as **MUSE**) for text-video retrieval. Specifically, MUSE is proposed with linear computational complexity for efficient cross-resolution modeling. Through extensive experiments, we can conclude that Mamba-like structures are efficient cross-resolution context learners, which leads to superior performance compared to the Transformer-based methods. As shown in Figure 1(b), our MUSE achieves state-of-the-art performance on the MSR-VTT dataset and a relatively small memory footprint and tunable parameters.

Since the community has few empirical experiences modeling multi-scale correlations in a linear complexity, we conduct extensive exploratory studies to find the optimal

*Corresponding Author

[1] We use "scale" for feature-level representation and "resolution" for pixel-level raw image.

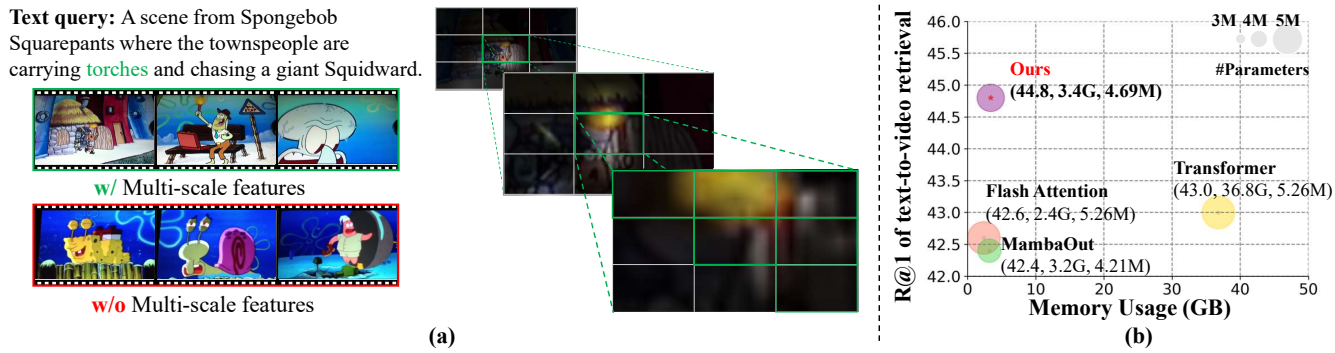

Figure 1: **(a) Illustration of multi-scale features.** Giving the text query, the model without multi-scale features retrieves the relevant but incorrect video because the small but crucial object *"torches"* can not be identified by only using frame-level feature representation (e.g., `[CLS]` tokens). We visualize the token similarity of the word *"torches"* and our extracted multi-scale features by organizing the attention map in a feature pyramid style from resolution low to high. Our model aggregates patches of the object *"torches"* that have a green boundary from multiple granularities to finally build a correlation between word *"torches"* and its visual entity in the video; **(b) Efficiency-performance comparisons.** The horizontal axis reflects memory usage, and the vertical is the R@1 metric of text-to-video retrieval on the MSR-VTT dataset. Marker sizes are proportional to the number of tunable parameters. Memory and parameters are calculated only on video learners without adding the backbone.

training architecture. We explore the following aspects: 1) **Plug-and-play manner**. The proposed MUSE is model-agnostic and can be compatible with existing TVR methods; 2) **Correlation modeling strategies**: We experiments popular architectures including MambaOut (Yu and Wang 2024), FlashAttention (Dao et al. 2022), and Mamba (Zhu et al. 2024); 3) **Scan strategies**: We experiment with existing sequence scan manners to find the optimal design; 4) **Scale combination manners**: Obtaining the representations in various scales, how to combine and arrange these representations is worth exploring. We hope our extensive explorations can shed light on effective and efficient linear attention modeling in multi-scale scenarios.

To conclude, the main contributions of this work are:

- We propose MUSE to explore the multi-scale representations for TVR, which are generated by applying a feature pyramid on the last single-scale feature map.
- We experiment with both Transformer and popular linear-attention architectures for joint resolution modeling and argue that Mamba is an efficient multi-scale learner for TVR.
- Extensive experiments show that our proposed MUSE achieves state-of-the-art performance on MSR-VTT, DiDeMo, and ActivityNet benchmarks.

Related Works

Text-Video Retrieval. TVR (Yu, Kim, and Kim 2018; Gabeur et al. 2020; Lei et al. 2020; Gorti et al. 2022; He et al. 2021; Lei et al. 2021; Luo et al. 2022; Ma et al. 2022; Wang et al. 2022; Cao et al. 2024; Jiang et al. 2022; Jin et al. 2023b, 2022, 2023c; Wu et al. 2024) is a pivotal task for video cross-modal learning, which has widespread application in video understanding (Liu et al. 2024c; Zhang et al. 2021, 2022; Li et al. 2023a; Cao et al. 2022c,a; Li et al. 2023b; Cao et al. 2023, 2022a) and multi-modal interactions (Cao et al. 2021, 2022b; Liu et al. 2023a; Ye et al. 2023; Ji et al. 2023; Luo et al. 2024; Liu et al. 2024b; Yang et al. 2021; Yang, Cao, and Zou 2023; Li et al. 2024a). With the advancement of image-text pretraining, recent works (Luo et al. 2022; Ma et al. 2022; Tu et al. 2022; Gorti et al. 2022; Jiang et al. 2022; Jin et al. 2023b) resort to image-text pretraining model CLIP (Radford et al. 2021) and focus on image-to-video transferring fine-tuning. The primary work CLIP4clip (Luo et al. 2022) investigates three kinds of temporal aggregation manners, which enlightens the follow-up works. To achieve fine-grained representations, X-CLIP (Ma et al. 2022) explores cross-grained contrastive learning, including video-sentence, video-word, frame-word and frame-sentence. Hunyuan_tvr (Jiang et al. 2022) divides the video-language interaction into frame-word, clip-phrase, and video-sentence granularities. DiCoSA (Jin et al. 2023b) further improves the fine-grained alignments by disentangling video into visual concepts. However, these methods neglect the cross-resolution relationships, which offer another perspective over the resolution-wise feature correlations. Our proposed MUSE bridges this gap by presenting scale-aware representations.

Multi-scale Video Modeling. Recent progress in image detection and segmentation finds that simply using the ViT (Dosovitskiy et al. 2020) output feature is insufficient for fine-grained image understanding. Thus, ViTDet (Li et al. 2022) has explored the plain vision transformer architecture to build a feature pyramid from ViT outputs, which achieves progress on dense image prediction. ViT-Adapter (Chen et al. 2022) trains an additional visual adapter to obtain multi-scale representations which considers both tasks prior and the input images. For video understanding, SlowFast (Feichtenhofer et al. 2019) uses the temporal-wise multi-scale branches with both low and high frame rates. MS-TCT

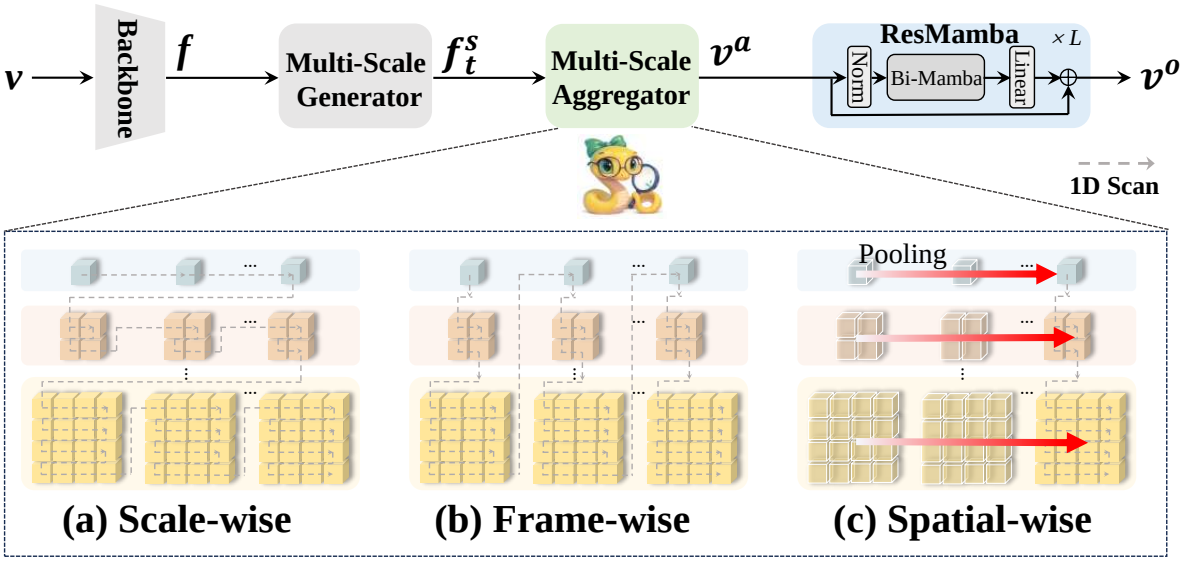

Figure 2: **Illustration of MUSE.** Our proposed method consists of three modules applied after video backbones. The generation module generates multi-scale video features based on single-scale visual output. Then, for the aggregation module, we test three different aggregation manners to aggregate multi-scale features into a 1D sequence. Finally, we design a residual architecture with Mamba to capture crucial video information from different granularities.

(Dai et al. 2022) proposes a temporal scale mixer module to effectively fuses multi-scale features. These methods fails to jointly model cross-scale features in a holistic manner. Therefore, we leverage Mamba (Gu and Dao 2023) of linear complexity as multi-scale video learner and design different manners for feature aggregation.

Mamba for Video. Based on the success of Mamba in language modeling, Vim (Zhu et al. 2024) and VMamba (Liu et al. 2024d) has pioneered Mamba architecture in vision by designing a bidirectional State Space Model and 2D selective scan manners for image recognition. The follow-up methods (Yang et al. 2024; Huang et al. 2024; Pei, Huang, and Xu 2024) explore different architecture design and selective scan manners. Also, some works (Shen et al. 2024; Ma, Li, and Wang 2024; Liu et al. 2024a; Chen et al. 2024b) have extended the success of Mamba to areas such as medical image processing, 3D reconstruction, point cloud understanding, etc. For video understanding, VideoMamba (Li et al. 2024b) first trains a video foundation model with Mamba backbone using the paradigm of unmasked teacher (Li et al. 2023c) and shows its efficiency compared with the transformer counterpart. Video mamba suite (Chen et al. 2024a) explores Mamba's effectiveness in the video downstream tasks by replacing attention blocks with its DBM blocks. In this work, we built a Mamba learner with a simple gated structure based on the Bidirectional Mamba (Zhu et al. 2024). We find that Mamba is an efficient multi-scale video learner that surpasses Transformer and other linear attention methods in text-video retrieval.

Methodology

In this section, we first introduce the architecture of standard text-video retrieval methods and how to extract multi-scale video features. Then, we thoroughly present our proposed method MUSE which consists of three components: Multi-scale generator, Multi-scale aggregator and ResMamba.

Overview

Feature extraction. Given a video $\boldsymbol{v} \in \mathcal{V}$ and the corresponding query $\boldsymbol{t} \in \mathcal{T}$, we utilize CLIP (Radford et al. 2021) with ViT (Dosovitskiy et al. 2020) to extract features. For the textual branch, we select the `[EOT]` token as text representations following (Radford et al. 2021). For the visual branch, former methods (e.g. CLIP4clip (Luo et al. 2022)) regard the frame-wise `[CLS]` token as the video-level representations $\boldsymbol{f}_{cls} \in \mathbb{R}^{T\times C}$. In contrast, to extract fine-grained video information, we utilize all visual tokens as video-level representations $\boldsymbol{f} \in \mathbb{R}^{T\times N\times C}$, where T is the frame number and N denotes the number of visual tokens.

Multi-scale feature generation. As shown in Figure 2, based on visual features $\boldsymbol{f}$, we aim to extract multi-scale video features $\boldsymbol{f}_t^s$, which denotes the feature representations for t-th frame at the scale $\boldsymbol{s}_i$, $s \in [1, S]$. We follow ViTDet (Li et al. 2022) by applying convolution or pooling operations on $\boldsymbol{f}$.

$$\boldsymbol{f}_t^s = \text{Pool}\left(\text{Conv}(\boldsymbol{f})\right). \tag{1}$$

Optimization. We follow (Radford et al. 2021) by using cross-entropy loss for optimization.

$$\mathcal{L} = -\log \frac{\exp(\boldsymbol{v}\cdot\boldsymbol{t}/\tau)}{\sum_{\boldsymbol{t}^-}\exp\left(\boldsymbol{v}\cdot\boldsymbol{t}^-/\tau\right)}, \tag{2}$$

where $\boldsymbol{t}^-$ is the unmatched language query.

Multi-scale Feature Aggregation

To conduct joint cross-resolution feature aggregations, we design three different methods: scale-wise, frame-wise, and spatial-wise.

Scale-wise. We first aggregate the video features from the same scale by temporal order, then rearrange the tokens from the same scale as a 1D sequence as shown in Figure 2(a). Finally, we concatenate tokens following scale orders from low resolution to high resolution. After aggregation, the video feature can be formulated as:

$$\begin{aligned} \boldsymbol{f}^s &= \{\boldsymbol{f}_1^s, \boldsymbol{f}_2^s, \cdots, \boldsymbol{f}_T^s\} \\ \boldsymbol{v}^a &= \{\boldsymbol{f}^s\}_{s=1}^S. \end{aligned} \tag{3}$$

Frame-wise. Different from the scale-wise manner, we first aggregate the video tokens from the same video frame and then rearrange them as a 1D sequence following scale order as shown in Figure 2(b). Finally, the video tokens are concatenated in frame order. In this manner, the aggregated video feature can be formulated as follows:

$$\begin{aligned} \boldsymbol{f}_t &= \{\boldsymbol{f}_t^1, \boldsymbol{f}_t^2, \cdots, \boldsymbol{f}_t^S\} \\ \boldsymbol{v}^a &= \{\boldsymbol{f}_t\}_{t=1}^T. \end{aligned} \tag{4}$$

Spatial-wise. In this manner, the aggregate order is the same as frame-wise for each frame. Differently, we first pool the video tokens by temporal dimension to aggregate temporal information and only keep spatial information. Then, we rearrange the tokens in a frame-wise manner. This manner is spatial-wise as it only focuses on the spatial dimension. Details are shown in Figure 2(c), and the final video feature can be formulated as follows:

$$\begin{aligned} \boldsymbol{f}^s &= \mathrm{meanpool}(\boldsymbol{f}_1^s, \boldsymbol{f}_2^s, \cdots, \boldsymbol{f}_T^s) \\ \boldsymbol{v}^a &= \{\boldsymbol{f}^s\}_{s=1}^S. \end{aligned} \tag{5}$$

In practice, We find that rearranging multi-scale video features as a 1D sequence in a scale-wise manner achieves the best performance as shown in Table 6. In Section 4, we will examine different scan methods and explain why scale-wise is effective. In the following, we will explain how we design the feature aggregation variants.

Mamba As Video Learner

To modify Mamba as an effective multi-scale video learner, we design a residual network following TimeSformer (Bertasius, Wang, and Torresani 2021) that can be noted as:

$$\boldsymbol{v}^o = \mathrm{ResMamba}(\boldsymbol{v}^a). \tag{6}$$

Specifically, with experiments we find that gated residual architecture works best for multi-scale video learning. In practice, we leverage a single *Linear* layer with zero initialization after Mamba block as gated function $\mathcal{G}(.)$ and the learning process can be formulated as:

$$\begin{aligned} \boldsymbol{h}_l &= \mathbf{A}\boldsymbol{h}_{l-1} + \mathbf{B}\boldsymbol{v}_l^a \\ \boldsymbol{y}_l &= \mathbf{C}\boldsymbol{h}_l \\ \boldsymbol{v}_{l+1}^a &= \mathcal{G}(\mathrm{Norm}(\boldsymbol{y}_l)) + \boldsymbol{v}_l^a, \end{aligned} \tag{7}$$

where l denotes the l_{th} layer in L Mamba layers. For Mamba block, $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the evolution parameter, $\mathbf{B} \in \mathbb{R}^{N\times 1}$ and $\mathbf{C} \in \mathbb{R}^{1\times N}$ are the projection parameters.

Experiment

Experimental Settings

Implementation Details. We set the input frame length to 12, 64, 64 and the caption token length to 32, 64, and 64 for MSR-VTT, DiDeMo, and ActivityNet, respectively. For fine-tuning, we keep the training hyperparameters and settings of the base model unchanged and train MUSE with a learning rate of 10 times higher (e.g., 1e-4 for CLIP4clip and 1e-3 for MUSE). The Layer number of ResMamba is set to 4, and the scale selected is $\{1, 3, 7, 14\}$. All experiments were carried out on 8 NVIDIA A100 GPUs.

Evaluation Metrics. For evaluation, we test the performance with standard retrieval metrics following CLIP4clip (Luo et al. 2022), which includes recall at rank K (R@K, higher is better), median rank (MdR, lower is better) and mean rank (MnR, lower is better). R@K defines the recall percentage of samples whose correct answer is found in the top-K retrieved results. We set K to $\{1, 5, 10\}$ following CLIP4clip (Luo et al. 2022). MdR is defined as the median of the ground-truth results rank in the result ranking list, while MnR is defined as the mean rank of all the correct results.

Datasets. To validate the effectiveness of our proposed model(MUSE), we test our model on three benchmarked datasets: **MSR-VTT** (Xu et al. 2016) contains 10,000 YouTube videos, and each video is associated with 20 textual descriptions. We follow the 1k-A split (Yu, Kim, and Kim 2018) where 9,000 videos are used for training and 1,000 videos for testing. **ActivityNet** (Krishna et al. 2017) comprises 20,000 untrimmed videos of complex human activities with an average duration of two minutes. We report results on the "val1" split (including 10,009 training videos and 4,917 testing videos) following (Gabeur et al. 2020). **DiDemo** (Anne Hendricks et al. 2017) consists of 10,464 unedited, personal videos in diverse visual settings annotated with 40,543 text descriptions. We follow the training and evaluation protocol in (Luo et al. 2022).

Performance Comparison

In this section, we validate the effectiveness and generalizability of our proposed method MUSE. Table 1 shows that our method can be applied to modern text-video retrieval models as a plug-and-play module. In Table 2 and Table 3, we compare our performance with state-of-the-art (SOTA) methods, and we find that directly appending MUSE to the former SOTA methods can achieve new SOTA performance on three widely used benchmarks.

Plug-and-play. To validate the generalizability of our proposed method, we test MUSE on four mainstream TVR baselines, including CLIP4clip (Luo et al. 2022), EMCL-Net (Jin et al. 2022), STAN (Liu et al. 2023b), and T-MASS (Wang et al. 2024). Specifically, we add MUSE as a video aggregator after video feature extraction of the baseline models, without any special modification except adjusting the learning rate of MUSE. As shown in Table 1, our proposed method outperforms baseline methods by 2.2 (+5.2%), 1.7 (+3.6%), 1.1 (+2.4%), and 0.9 (+1.8%) on MSR-VTT text-to-video retrieval R@1 result. For other

Methods	Text->Video			Video->Text		
	R@1↑	R@5↑	R@10↑	R@1↑	R@5↑	R@10↑
CLIP4Clip† (Luo et al. 2022)	42.6	70.8	79.9	43.9	70.0	81.4
+ MUSE (Ours)	**44.8 (+2.2)**	**71.6 (+0.8)**	**82.1 (+2.2)**	**44.9 (+1.0)**	**70.8 (+0.8)**	**82.2 (+0.8)**
EMCL-Net† (Jin et al. 2022)	47.1	72.7	82.3	44.4	72.6	82.6
+ MUSE (Ours)	**48.8 (+1.7)**	**74.1 (+1.4)**	**83.4 (+1.1)**	**47.4 (+3.0)**	**75.8 (+3.2)**	**82.9 (+0.3)**
STAN† (Liu et al. 2023b)	46.2	72.6	81.1	44.5	71.9	81.7
+ MUSE (Ours)	**47.3 (+1.1)**	**73.1 (+0.5)**	**82.2 (+1.1)**	**45.5 (+1.0)**	**73.1 (+1.4)**	**81.8 (+0.1)**
T-MASS† (Wang et al. 2024)	50.0	75.3	84.2	46.0	77.1	86.2
+ MUSE (Ours)	**50.9 (+0.9)**	**76.7 (+1.5)**	**85.6 (+1.4)**	**49.7 (+3.7)**	**77.8 (+0.7)**	**86.5 (+0.3)**

Table 1: Plug-and-play experiments on MSR-VTT. We compare the text-to-video retrieval results before and after adding our proposed method, **MUSE**, on four baseline models. † denotes our reproduction of the method.

Methods	Text → Video					Video → Text				
	R@1↑	R@5↑	R@10↑	MdR↓	MnR↓	R@1↑	R@5↑	R@10↑	MdR↓	MnR↓
CLIP4Clip(Luo et al. 2022)	44.5	71.4	81.6	2.0	15.3	42.7	70.9	80.6	2.0	11.6
X-Pool(Gorti et al. 2022)	46.9	72.8	82.2	2.0	14.3	44.4	73.3	84.0	2.0	9.0
STAN(Liu et al. 2023b)	46.9	72.8	82.8	2.0	-	-	-	-	-	-
EMCL-Net(Jin et al. 2022)	46.8	73.1	83.1	2.0	-	46.5	73.5	83.5	2.0	-
HBI(Jin et al. 2023a)	48.6	74.6	83.4	2.0	12.0	46.8	74.3	84.3	2.0	8.9
DiffusionRet(Jin et al. 2023c)	49.0	75.2	82.7	2.0	12.1	47.7	73.8	84.5	2.0	8.8
T-MASS(Wang et al. 2024)	50.2	75.3	85.1	1.0	11.9	47.7	**78.0**	86.3	2.0	8.0
MUSE (Ours)	**50.9**	**76.7**	**85.6**	**1.0**	**10.9**	**49.7**	<u>77.8</u>	**86.5**	**2.0**	**7.4**

Table 2: Comparisons with state-of-the-art methods on MSR-VTT dataset. Models are tested with CLIP-ViT-B/32(Radford et al. 2021). The best performance is in **bold** and the second best is <u>underlined</u>.

evaluation metrics, our proposed method also outperforms the baseline method with a considerable improvement (3.0 (+6.8%) of EMCL-Net and 3.7 (+8.0%) of T-MASS on video-to-text R@1 result). It is worth mentioning that, based on the former SOTA method, T-MASS, adding MUSE still makes considerable improvements, especially in video-to-text retrieval. The improvement is because T-MASS mainly designed its model on the text branch and implemented fewer modifications on the video branch where our MUSE fills this blank. From this point of view, the least improvement of STAN among the four selected baselines is because that STAN has designed a new branch beside CLIP visual encoder to extract more complex video embeddings. The above results further confirm the generalizability of MUSE and the potential of MUSE to be an improvement module for any CLIP-based TVR model.

Comparison with state-of-the-arts. The comparisons are shown in Table 2 and Table 3. As mentioned above, our proposed method is a plug-and-play model that can be applied on most ViT-based TVR models. Thus, we directly add our designed module to the former state-of-the-art (SOTA) methods and evaluate it on three popular TVR datasets. For the result of MSR-VTT, in Table 2, combined with T-MASS, our proposed method shows superior performance on almost all the evaluation metrics, achieving the new SOTA among MSR-VTT CLIP-ViT-B/32 based methods. Also we achieves SOTA performance on DiDeMo and Activity-Net by applying MUSE on the baseline models. Specifically, we select T-MASS, DiffusionRet for DiDeMo, ActivityNet considering performance and implementation difficulty.

Ablative Analysis of Correlation Modeling

Why using Mamba? To answer the question of why Mamba is our model selection, we testify the superiority of Mamba in two perspectives: **(1) Memory Usage.** We compare the memory usage between Transformer attention of quadratic complexity and Mamba of Linear complexity by calculating the memory usage as the input frame number grows. As shown in Figure 3, we test the memory usage based on CLIP4clip with batch size 16 and scale {1, 3, 7, 14}. The green line reflects the memory usage of the base model using mean pooling without applying MUSE. The pink line with marker star and blue line with dot reflects the memory growth when leveraging Mamba and Transformer as the video learner after multi-scale feature extraction and aggregation. From the comparison, we can easily identify that Mamba has a remarkably lower computation resource requirement than Transformer attention. For instance, when the input frame number comes to 20, the model's memory with Transformer needs more than 80GB GPU mem-

Methods	DiDeMo R@1↑	R@5↑	R@10↑	MdR↓	MnR↓	ActivityNet R@1↑	R@5↑	R@10↑	MdR↓	MnR↓
CLIP4Clip (Luo et al. 2022)	43.4	70.2	80.6	2.0	17.5	40.5	72.4	-	2.0	7.4
X-CLIP(Ma et al. 2022)	45.2	74.0	-	-	14.6	44.3	74.1	-	-	7.9
HBI(Jin et al. 2023a)	46.9	74.9	82.7	2.0	12.1	42.2	73.0	84.6	2.0	6.6
DiCoSA(Jin et al. 2023b)	45.7	74.6	83.5	2.0	11.7	42.1	73.6	84.6	2.0	6.8
DiffusionRet(Jin et al. 2023c)	46.7	74.7	82.7	2.0	14.3	45.8	75.6	86.3	2.0	6.5
T-MASS(Wang et al. 2024)	50.9	77.2	85.3	1.0	12.1	-	-	-	-	-
MUSE(Ours)	**51.5**	**77.7**	**86.0**	**1.0**	**11.3**	**46.2**	**76.9**	**86.8**	**2.0**	**5.8**

Table 3: Comparisons with state-of-the-art methods on DiDeMo and ActivityNet Datasets.

Module	R@1↑	R@5↑	R@10↑	MnR↓	Memory(GB)↓
Transformer	43.0	71.1	80.0	16.3	36.80
FlashAttention	42.6	69.3	79.7	16.3	**2.38**
MambaOut	42.4	70.2	80.7	**15.4**	3.28
Mamba	**44.8**	**71.6**	**82.1**	15.6	3.40

Table 4: Ablations on model selection. Input frame number is set to 12. The layer number of video learner is set to 4. Memory footprint is evaluated with batch size of 16.

Scan Type	R@1↑	R@5↑	R@10↑	MnR↓
"none"	44.1	71.7	80.5	**14.4**
"v1"	44.0	71.0	80.7	14.9
"v2"	**44.8**	**71.6**	**82.1**	15.6

Table 5: Ablations on scan strategies.

ory on a single GPU, which exceeds the memory-bound of most modern hardware accelerators. In contrast, Mamba only needs 15.37GB memory at input frame 16, which saves **79.7%** (76.34GB) memory resources of the Transformer. Due to the linear complexity of Mamba, the memory growth is more acceptable even when it comes to 64 frames input (47.20GB in memory). **(2) Performance.** To further validate the effectiveness of Mamba architecture, we compare the performance of Mamba with Transformer and other models with linear computational complexity including MambaOut (Yu and Wang 2024) and FlashAttention (Dao et al. 2022). As shown in Table 4, Mamba performs best on MSR-VTT among the four listed models. We replace Mamba block with other counterparts for a fair comparison. Compared with FlashAttention and Transformer, Mamba achieves better performance with almost the same memory usage as FlashAttention. To validate the effectiveness of SSM architecture in Mamba, we compare it with MambaOut which is implemented by only removing the SSM module. Results show that most of the ability to model video sequences comes from the SSM structure of Mamba.

Based on the above data, Mamba performs well in terms of performance and efficiency. Therefore, we leverage Mamba for multi-scale video sequence modeling.

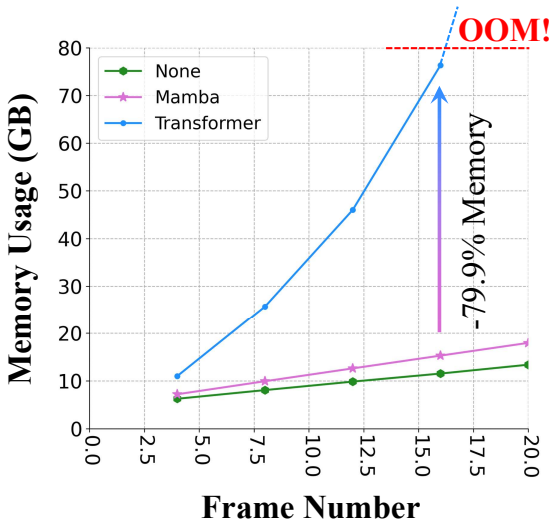

Figure 3: **Comparison of the memory usage among Transformer, Mamba, and Baseline.** The baseline selected is CLIP4clip(Luo et al. 2022) with mean pooling for feature aggregation.

Ablative Analysis of Scan Strategies

Then, we also ablate the effects of scanning manners provided by the Mamba block from Vim (Zhu et al. 2024) in Table 5. For scan type, "none" refers to the one-direction scan of the original Mamba block, while "v1" and "v2" refer to the bidirectional scan. The difference between "v1" and "v2" is that "v1" shares projection weights of forward and backward while "v2" does not. We found that "v2," which denotes the bidirectional scanning type with separate projection weights, performs best in video multi-scale learning. This demonstrates that scanning from low-to-high and high-to-low resolutions is crucial to the overall performance.

Ablative Analysis of Scale Combination

Ablations on aggregation manners. We experiment with three different manners for feature aggregation in Figure 2. In Table 6, we ablate the performance of the three types of aggregation manners. We find that aggregating features scale-wise achieves the best performance for feature aggre-

Agg. Mode	R@1↑	R@5↑	R@10↑	MnR↓
Frame	43.5	70.6	80.0	**15.2**
Spatial	44.4	70.3	80.9	15.4
Scale	**44.8**	**71.6**	**82.1**	15.6

Table 6: Ablations on aggregation manners.

Scale	Memory(GB)↓	R@1↑	R@5↑	R@10↑	MnR↓
{1}	**7.62**	43.6	71.2	81.8	15.2
{1, 3}	7.82	44.0	70.8	81.2	15.8
{1, 3, 7}	8.76	44.3	**71.8**	81.7	15.9
{1, 3, 7, 14}	12.60	**44.8**	71.6	**82.1**	15.6
{1, 3, 7 14, 28}	29.36	42.5	71.4	81.6	**15.1**

Table 7: Ablations on video feature scale selection. {1} denotes the scale of `[CLS]` tokens, and numbers in curly brackets denote scale multiple times larger than `[CLS]` tokens in both width and height.

gation. This demonstrates that scale is essential for modeling video multi-scale features.

Ablations on scale selection. Table 7 compares different video scale selection manners. We find that simply adding our Mamba Learner to the original CLIP4clip at scale {1} still improves the performance of text-to-video retrieval results on MSR-VTT. We believe that this is the natural superiority of Mamba architecture in multi-scale video modeling. From the results, we can tell that with larger scales, our model has a trend of improving its performance except for the scale at 4x the original scale. That is because as the scale goes 4x larger, the newly added tokens get 16x longer, which brings too much redundancy when finding key information. Moreover, longer tokens also bring computation costs, which makes the model inefficient (e.g., GPU memory grows by 16.76GB). Thus, we design our method with a scale selection of {1, 3, 7, 14} considering both efficiency and performance.

Ablations on Layer numbers. To further validate the effectiveness of our method, we conduct experiments on the structure of Mamba Lenarner. Specifically, we ablate the layer number L of the ResMamba block in Table 8. As shown in Table 8, we can tell that with more ResMamba blocks, the performance still gets better, demonstrating our model design's superiority. However, we only use four layers of the ResMamba block as Mamba Learner due to the consideration of computation cost and applicability to the base models.

Visualization Results

In this section, we visualize some text-to-video retrieval samples from the MST-VTT testing split. As shown in Figure 4, videos in green are the correct answers retrieved by adding MUSE on the base model and videos in red are failure results retrieved by the base model without any modification. We choose the base model as CLIP4clip with mean pooling at 12 frame input. We mark the crucial visual hints that distinguish correct answers from incorrect ones with orange boxes. The top left example reflects our model's capability of capturing small visual entity ("a little brush" in frames 1 and 3) which is essential for retrieving correct answers. Likewise, the bottom left shows that our model can notice objects of multiple granularities, e.g.,"A man" in the left part of the picture which can be treated as a large entity. This is because Mamba is an effective model for multi-scale video sequence modeling. The top right example shows that fine-grained features improve the recognition of visual entities ("a cat" rather than "a dog" or "a bird"). In the bottom right, the case shows our model's potential capability of identifying the relationship between visual entities ("into the crowd" is essential to distinguish the two videos). The above examples demonstrate that multi-scale features are critical for correct video retrieval and our proposed method MUSE improves this capability of the base model.

Layers	Memory(GB)↓	R@1↑	R@5↑	R@10↑	MnR↓
L=0	**9.20**	42.6	70.8	79.9	16.1
L=2	10.46	44.0	70.9	80.6	15.0
L=4	12.60	44.8	71.6	**82.1**	15.6
L=8	16.69	45.0	72.1	81.4	**14.6**
L=16	25.05	**45.6**	**72.4**	81.5	14.7

Table 8: Ablations on ResMamba Layer number L.

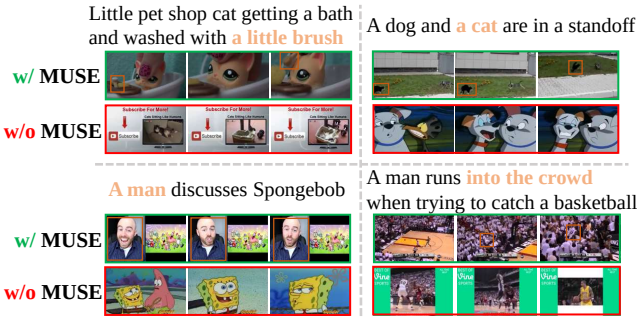

Figure 4: **Visualization of text-video retrieval examples.** We sorted results based on their similarity scores and visualized the rank one result. Green: correct with MUSE; Red: incorrect without MUSE. Crucial visual hints are marked with orange boxes.

Conclusion

This paper presents MUSE acting as an efficient multi-resolution learner for text-video retrieval. Based on the plain structure of the pre-trained CLIP model, we generate multi-scale features by simply applying a feature pyramid on the last layer feature. For the cross-resolution feature integration, we leverage Mamba to achieve effective and efficient context modeling. Extensive experiments illustrate that MUSE achieves state-of-the-art performance and scalable plug-and-play characteristics.

Acknowledgments

This work is supported in part by National Key Research and Development Program of China (2024YFE0203100) and Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology (Grant No. 2024B1212010006).

References

Anne Hendricks, L.; Wang, O.; Shechtman, E.; Sivic, J.; Darrell, T.; and Russell, B. 2017. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, 5803–5812.

Bertasius, G.; Wang, H.; and Torresani, L. 2021. Is space-time attention all you need for video understanding? In *ICML*, volume 2, 4.

Cao, M.; Chen, L.; Shou, M. Z.; Zhang, C.; and Zou, Y. 2021. On pursuit of designing multi-modal transformer for video grounding. *arXiv preprint arXiv:2109.06085*.

Cao, M.; Jiang, J.; Chen, L.; and Zou, Y. 2022a. Correspondence matters for video referring expression comprehension. In *Proceedings of the 30th ACM International Conference on Multimedia*, 4967–4976.

Cao, M.; Tang, H.; Huang, J.; Jin, P.; Zhang, C.; Liu, R.; Chen, L.; Liang, X.; Yuan, L.; and Li, G. 2024. RAP: Efficient Text-Video Retrieval with Sparse-and-Correlated Adapter. *arXiv preprint arXiv:2405.19465*.

Cao, M.; Wei, F.; Xu, C.; Geng, X.; Chen, L.; Zhang, C.; Zou, Y.; Shen, T.; and Jiang, D. 2023. Iterative proposal refinement for weakly-supervised video grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6524–6534.

Cao, M.; Yang, T.; Weng, J.; Zhang, C.; Wang, J.; and Zou, Y. 2022b. Locvtp: Video-text pre-training for temporal localization. In *European Conference on Computer Vision*, 38–56. Springer.

Cao, M.; Zhang, C.; Chen, L.; Shou, M. Z.; and Zou, Y. 2022c. Deep motion prior for weakly-supervised temporal action localization. *IEEE Transactions on Image Processing*, 31: 5203–5213.

Chen, G.; Huang, Y.; Xu, J.; Pei, B.; Chen, Z.; Li, Z.; Wang, J.; Li, K.; Lu, T.; and Wang, L. 2024a. Video mamba suite: State space model as a versatile alternative for video understanding. *arXiv preprint arXiv:2403.09626*.

Chen, K.; Chen, B.; Liu, C.; Li, W.; Zou, Z.; and Shi, Z. 2024b. Rsmamba: Remote sensing image classification with state space model. *IEEE Geoscience and Remote Sensing Letters*.

Chen, Z.; Duan, Y.; Wang, W.; He, J.; Lu, T.; Dai, J.; and Qiao, Y. 2022. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*.

Dai, R.; Das, S.; Kahatapitiya, K.; Ryoo, M. S.; and Brémond, F. 2022. Ms-tct: Multi-scale temporal convtransformer for action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20041–20051.

Dao, T.; Fu, D.; Ermon, S.; Rudra, A.; and Ré, C. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35: 16344–16359.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Feichtenhofer, C.; Fan, H.; Malik, J.; and He, K. 2019. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6202–6211.

Gabeur, V.; Sun, C.; Alahari, K.; and Schmid, C. 2020. Multi-modal transformer for video retrieval. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, 214–229. Springer.

Gorti, S. K.; Vouitsis, N.; Ma, J.; Golestan, K.; Volkovs, M.; Garg, A.; and Yu, G. 2022. X-pool: Cross-modal language-video attention for text-video retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5006–5015.

Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

He, F.; Wang, Q.; Feng, Z.; Jiang, W.; Lü, Y.; Zhu, Y.; and Tan, X. 2021. Improving video retrieval by adaptive margin. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1359–1368.

Huang, T.; Pei, X.; You, S.; Wang, F.; Qian, C.; and Xu, C. 2024. Localmamba: Visual state space model with windowed selective scan. *arXiv preprint arXiv:2403.09338*.

Ji, J.; Cao, M.; Song, T.; Chen, L.; Wang, Y.; and Zou, Y. 2023. Video Referring Expression Comprehension via Transformer with Content-conditioned Query. In *Proceedings of the 1st International Workshop on Deep Multimodal Learning for Information Retrieval*, 39–48.

Jiang, J.; Min, S.; Kong, W.; Wang, H.; Li, Z.; and Liu, W. 2022. Tencent text-video retrieval: Hierarchical cross-modal interactions with multi-level representations. *IEEE Access*.

Jin, P.; Huang, J.; Liu, F.; Wu, X.; Ge, S.; Song, G.; Clifton, D.; and Chen, J. 2022. Expectation-maximization contrastive learning for compact video-and-language representations. *Advances in neural information processing systems*, 35: 30291–30306.

Jin, P.; Huang, J.; Xiong, P.; Tian, S.; Liu, C.; Ji, X.; Yuan, L.; and Chen, J. 2023a. Video-text as game players: Hierarchical banzhaf interaction for cross-modal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2472–2482.

Jin, P.; Li, H.; Cheng, Z.; Huang, J.; Wang, Z.; Yuan, L.; Liu, C.; and Chen, J. 2023b. Text-video retrieval with disentangled conceptualization and set-to-set alignment. *arXiv preprint arXiv:2305.12218*.

Jin, P.; Li, H.; Cheng, Z.; Li, K.; Ji, X.; Liu, C.; Yuan, L.; and Chen, J. 2023c. Diffusionret: Generative text-video retrieval with diffusion model. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2470–2481.

Krishna, R.; Hata, K.; Ren, F.; Fei-Fei, L.; and Carlos Niebles, J. 2017. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, 706–715.

Lei, J.; Li, L.; Zhou, L.; Gan, Z.; Berg, T. L.; Bansal, M.; and Liu, J. 2021. 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*, 7331–7341.

Lei, J.; Yu, L.; Berg, T. L.; and Bansal, M. 2020. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, 447–463. Springer.

Li, H.; Cao, M.; Cheng, X.; Li, Y.; Zhu, Z.; and Zou, Y. 2023a. G2l: Semantically aligned and uniform video grounding via geodesic and game theory. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 12032–12042.

Li, H.; Cao, M.; Cheng, X.; Li, Y.; Zhu, Z.; and Zou, Y. 2024a. Exploiting auxiliary caption for video grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 18508–18516.

Li, H.; Cao, M.; Cheng, X.; Zhu, Z.; Li, Y.; and Zou, Y. 2023b. Generating templated caption for video grounding. *arXiv preprint arXiv*, 2301: 2.

Li, K.; Li, X.; Wang, Y.; He, Y.; Wang, Y.; Wang, L.; and Qiao, Y. 2024b. Videomamba: State space model for efficient video understanding. *arXiv preprint arXiv:2403.06977*.

Li, K.; Wang, Y.; Li, Y.; Wang, Y.; He, Y.; Wang, L.; and Qiao, Y. 2023c. Unmasked teacher: Towards training-efficient video foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 19948–19960.

Li, Y.; Mao, H.; Girshick, R.; and He, K. 2022. Exploring plain vision transformer backbones for object detection. In *European conference on computer vision*, 280–296. Springer.

Liu, J.; Wang, Z.; Ye, Q.; Chong, D.; Zhou, P.; and Hua, Y. 2023a. Qilin-med-vl: Towards chinese large vision-language model for general healthcare. *arXiv preprint arXiv:2310.17956*.

Liu, J.; Yu, R.; Wang, Y.; Zheng, Y.; Deng, T.; Ye, W.; and Wang, H. 2024a. Point mamba: A novel point cloud backbone based on state space model with octree-based ordering strategy. *arXiv preprint arXiv:2403.06467*.

Liu, R.; Huang, J.; Li, G.; Feng, J.; Wu, X.; and Li, T. H. 2023b. Revisiting temporal modeling for clip-based image-to-video knowledge transferring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6555–6564.

Liu, R.; Li, C.; Ge, Y.; Li, T. H.; Shan, Y.; and Li, G. 2024b. BT-Adapter: Video Conversation is Feasible Without Video Instruction Tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13658–13667.

Liu, R.; Li, C.; Tang, H.; Ge, Y.; Shan, Y.; and Li, G. 2024c. ST-LLM: Large Language Models Are Effective Temporal Learners. *arXiv preprint arXiv:2404.00308*.

Liu, Y.; Tian, Y.; Zhao, Y.; Yu, H.; Xie, L.; Wang, Y.; Ye, Q.; and Liu, Y. 2024d. VMamba: Visual State Space Model. arXiv:2401.10166.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Luo, H.; Ji, L.; Zhong, M.; Chen, Y.; Lei, W.; Duan, N.; and Li, T. 2022. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 508: 293–304.

Luo, Y.; Li, H.; Wu, X.; Cao, M.; Huang, X.; Zhu, Z.; Liao, P.; Chen, H.; and Zhang, Y. 2024. Textual Inversion and Self-supervised Refinement for Radiology Report Generation. *arXiv preprint arXiv:2405.20607*.

Ma, J.; Li, F.; and Wang, B. 2024. U-mamba: Enhancing long-range dependency for biomedical image segmentation. *arXiv preprint arXiv:2401.04722*.

Ma, Y.; Xu, G.; Sun, X.; Yan, M.; Zhang, J.; and Ji, R. 2022. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia*, 638–647.

Pei, X.; Huang, T.; and Xu, C. 2024. Efficientvmamba: Atrous selective scan for light weight visual mamba. *arXiv preprint arXiv:2403.09977*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Shen, Q.; Yi, X.; Wu, Z.; Zhou, P.; Zhang, H.; Yan, S.; and Wang, X. 2024. Gamba: Marry gaussian splatting with mamba for single view 3d reconstruction. *arXiv preprint arXiv:2403.18795*.

Tu, Y.; Li, L.; Su, L.; Gao, S.; Yan, C.; Zha, Z.-J.; Yu, Z.; and Huang, Q. 2022. I 2 Transformer: Intra-and inter-relation embedding transformer for TV show captioning. *IEEE Transactions on Image Processing*, 31: 3565–3577.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, J.; Sun, G.; Wang, P.; Liu, D.; Dianat, S.; Rabbani, M.; Rao, R.; and Tao, Z. 2024. Text Is MASS: Modeling as Stochastic Embedding for Text-Video Retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16551–16560.

Wang, Q.; Zhang, Y.; Zheng, Y.; Pan, P.; and Hua, X.-S. 2022. Disentangled representation learning for text-video retrieval. *arXiv preprint arXiv:2203.07111*.

Wang, Z.; Sung, Y.-L.; Cheng, F.; Bertasius, G.; and Bansal, M. 2023. Unified coarse-to-fine alignment for video-text retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2816–2827.

Wu, X.; Li, H.; Luo, Y.; Cheng, X.; Zhuang, X.; Cao, M.; and Fu, K. 2024. Uncertainty-aware sign language video retrieval with probability distribution modeling. In *European Conference on Computer Vision*, 390–408. Springer.

Xu, J.; Mei, T.; Yao, T.; and Rui, Y. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5288–5296.

Yang, B.; Cao, M.; and Zou, Y. 2023. Concept-aware video captioning: Describing videos with effective prior information. *IEEE Transactions on Image Processing*.

Yang, C.; Chen, Z.; Espinosa, M.; Ericsson, L.; Wang, Z.; Liu, J.; and Crowley, E. J. 2024. Plainmamba: Improving non-hierarchical mamba in visual recognition. *arXiv preprint arXiv:2403.17695*.

Yang, D.; Zou, Y.; Zhang, C.; Cao, M.; and Chen, J. 2021. RR-Net: Relation reasoning for end-to-end human-object interaction detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(6): 3853–3865.

Ye, Q.; Liu, J.; Chong, D.; Zhou, P.; Hua, Y.; and Liu, A. 2023. Qilin-med: Multi-stage knowledge injection advanced medical large language model. *arXiv preprint arXiv:2310.09089*.

Yu, W.; and Wang, X. 2024. MambaOut: Do We Really Need Mamba for Vision? *arXiv preprint arXiv:2405.07992*.

Yu, Y.; Kim, J.; and Kim, G. 2018. A joint sequence fusion model for video question answering and retrieval. In *Proceedings of the European conference on computer vision (ECCV)*, 471–487.

Zhang, C.; Cao, M.; Yang, D.; Chen, J.; and Zou, Y. 2021. Cola: Weakly-supervised temporal action localization with snippet contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16010–16019.

Zhang, C.; Yang, T.; Weng, J.; Cao, M.; Wang, J.; and Zou, Y. 2022. Unsupervised pre-training for temporal action localization tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14031–14041.

Zhu, L.; Liao, B.; Zhang, Q.; Wang, X.; Liu, W.; and Wang, X. 2024. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*.

Appendix

In the appendix, we list some of the detailed illustrations of our proposed method following the suggestions of reviewers. These include illustration of multi-scale feature extraction, training time, GPU memory consumption, and illustration of ResMamba architecture.

A. Illustration of multi-scale feature extraction

In our implementation, we utilize several Conv2d and MaxPooling modules to extract multi-scale video features based on the output from CLIP, as discussed in the "Multi-scale Feature Generation" section. For instance, if the CLIP output has dimensions 14x14 and the scale factor is s=3, we first apply a max-pooling operation to reduce the feature map to 3x3, and then use multiple Conv2d layers with LayerNorm to extract features of this scale. Through experimentation, we have found that this straightforward approach provides the most effective results, which aligns with similar strategies discussed in the ViTDet(Li et al. 2022) model.

B. Training time after adding MUSE

Regarding the time complexity, as Table 9 shows, after incorporating MUSE, the training time on the MSR-VTT dataset increased to 17.3 GPU hours, which is a 10.9% increase compared to the original CLIP4clip baseline of 15.6 GPU hours. This training time was measured using 12 video frames as input. Considering the significant performance improvement and the relatively low memory overhead, we believe that the slight increase in training time is acceptable for the community, especially for training text-video retrieval models.

Method	R@1↑	R@5↑	R@10↑	Time(hr)↓
Baseline	42.6	70.8	79.9	**15.6**
+ MUSE	**44.8**	**71.6**	**82.1**	17.3

Table 9: Training time comparison after adding MUSE. The baseline model is CLIP4clip with 12 frame inputs.

C. GPU memory consumption.

We list the GPU memory consumption during training of the baseline model together with the models with MUSE or Transformer. As shown in Table 10, the memory required by MUSE is 12.6GB for each GPU when training with batch size 128 frame 12 on 8 A100s.

Method	R@1↑	R@5↑	R@10↑	Memory(GB)↓
Baseline	42.6	70.8	79.9	**9.84**
+ MUSE	**44.8**	**71.6**	**82.1**	12.6 (+2.76)
+ Transformer	43.0	71.1	80.0	36.8 (+26.96)

Table 10: GPU memory comparison. We list the memory consumption of the Baseline model (CLIP4clip), MUSE, and its Transformer counterpart.

D. Detailed illustration of ResMamba architecture

In our paper, we advance Mamaba to ResMamba with the following two considerations: 1) Stable training: Residual connections have been widely used in various tasks for stabling gradient back-propagation; 2) Feature retaining: This simple skip-connection maintains the un-traversed features in the fused features. Specifically, we implemented a gated network for output features to pass through before being added to the input features, it consists of a LayerNorm followed by a Linear layer with zero initialization. The ablations are shown in Table 11.

Method	R@1↑	R@5↑	R@10↑	MnR↓
w/o residual	43.9	**71.7**	81.6	**15.1**
w/ residual	**44.8**	71.6	**82.1**	15.6

Table 11: Ablations of residual architecture.