SMAUG: Sparse Masked Autoencoder for Efficient Video-Language Pre-training

Yuanze Lin[1] Chen Wei[2] Huiyu Wang[2] Alan Yuille[2] Cihang Xie[3]
[1] University of Washington [2] Johns Hopkins University [3] UC Santa Cruz

Abstract

*Video-language pre-training is crucial for learning powerful multi-modal representation. However, it typically requires a massive amount of computation. In this paper, we develop SMAUG, an efficient pre-training framework for video-language models. The foundation component in SMAUG is masked autoencoders. Different from prior works which only mask textual inputs, our masking strategy considers both visual and textual modalities, providing a better cross-modal alignment and saving more pre-training costs. On top of that, we introduce a space-time token sparsification module, which leverages context information to further select only "important" spatial regions and temporal frames for pre-training. Coupling all these designs allows our method to enjoy both competitive performances on text-to-video retrieval and video question answering tasks, and much less pre-training costs by **1.9**× or more. For example, our SMAUG only needs ∼**50** NVIDIA A6000 GPU hours for pre-training to attain competitive performances on these two video-language tasks across six popular benchmarks.*

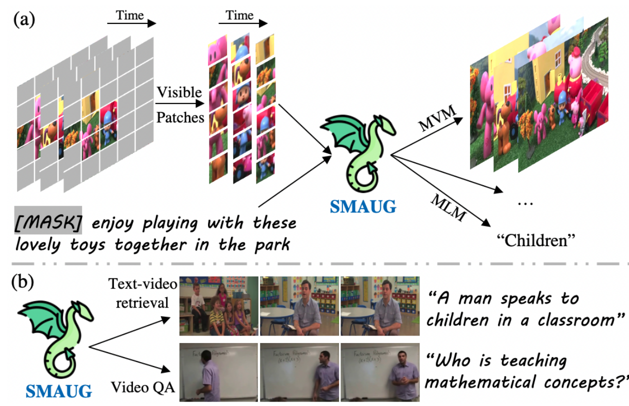

Figure 1: An overview of SMAUG (**S**parse **M**asked **A**utoencoder for video-lang**U**a**G**e pre-training). (a) During pre-training, we randomly mask out a large subset of individual frames' patches, and then utilize the *visible patches* and language sentences for video-text pre-training, which includes masked visual/language modeling (MVM/MLM) and *etc*. (b) The pre-trained models can then be fine-tuned on several down-stream video-language tasks, *e.g.*, text-to-video retrieval and video question answering (video QA).

1. Introduction

Recently, video-language pre-training [27, 29, 2, 67, 12, 38, 54] stands as the common practice to learn cross-modal representations on large-scale video-text datasets [47, 6, 2]. Such pre-trained models show strong transfer performances on a range of vision and language tasks, including visual question answering [62, 34], text-to-video retrieval [62, 1], visual reasoning [48] and video understanding [33, 4, 58]. Nonetheless, the corresponding training cost of these advanced video-language models is enormous. For example, the training of CLIP4Clip [38] needs ∼2 weeks with 8 GPUs, which therefore largely limits their explorations in a wider aspect. This invites us to ponder a thought-provoking but rarely explored question in this paper: *How can we still pre-train powerful video-language models while significantly reducing their pre-training cost?*

Interestingly, we note that the recent work Masked Autoencoders (MAE) [17], which establishes an efficient self-supervised paradigm for training models at scale, potentially offering a solution to the aforementioned question. In MAE, a large amount of image patches (*e.g.*, 75%) are masked. The heavy encoder only executes on a small portion of the visible patches, and the lightweight decoder reconstructs the other large portion of masked patches. This mask reconstruction process is a computationally efficient instantiation of masked visual modeling (*i.e.*, MVM), which has already been shown effective for helping video-language pre-training [13, 24]. Therefore, we conjecture that resorting to such MAE fashion can substantially mitigate the computational burden and still achieve satisfactory performances for video-language pre-training models.

Another interesting observation is that, even by masking out a significant portion of image patches as in MAE, the information could still be redundant [58, 11]. As argued in [44, 31], *not all patches are equally important*: an image could contain a significant amount of less informative visual patches (*e.g*., background patches), which scarcely or even negatively contribute to vision-language representation learning. What is worse, this issue could be severer in the *video*-language setting, as additionally, *not all frames are equally important* [26, 27]. For example, a video clip may contain a non-negligible portion of frames with just trivial noises (*e.g*., camera shake). Further eliminating these redundancies is expected to provide an extra speedup to video-language pre-training.

Based on the observations above, we present SMAUG, an efficient pre-training framework for video-language models. Our work is built upon MAE. We mask out a significant amount of space-time patches and let the autoencoder learn to reconstruct them. Next, we introduce a space-time token sparsification module to remove spatial and temporal redundancies: 1) we leverage the attention weights in the visual encoder to predict attentive or inattentive tokens to reduce spatial patches among individual frames. Attentive tokens are preserved while inattentive tokens are fused; 2) we propose a learnable Transformer-based network to pick up important video frames among the given video clip.

We evaluate SMAUG on two video-language tasks, including text-to-video retrieval and video question answering across six datasets. For text-to-video retrieval, the experiments are performed on MSRVTT [62], DiDeMo [1] and ActivityNet Captions [22]. For video question answering, MSRVTT-MC [64], MSRVTT-QA [60] and ActivityNet-QA [65] are used. SMAUG can achieve state-of-the-art or comparable performances over all six datasets. Meanwhile, the proposed method can achieve ∼**1.9**× video-language pre-training speedup. For example, SMAUG can finish video-language pre-training only with ∼**50** NVIDIA A6000 GPU hours.

2. Related Work

Video-and-language pre-training. The standard pipeline of video-language pre-training (*i.e*., first pre-train and then fine-tune) [54, 69, 12, 38, 37, 26] aims at learning a generalizable multi-modal feature representation for a range of downstream tasks, such as text-to-video retrieval [62, 1, 22, 10], video question answering [64, 60, 65, 25, 30], video captioning [57, 62, 68, 55], *etc*.

UniVL [37] proposes a unified video-language pre-training model for multi-modal generation and understanding. Clip4clip [38] transfers the image-text pre-trained model (*i.e*., CLIP [43]) for video-retrieval task in an end-to-end manner. Singularity [26] reveals that the video-language models pre-trained with a single video frame can still attain significant performances for video-and-language downstream tasks. Our work is directly motivated by Singularity [26], which also pre-trains models by leveraging single-frame and multiple-frame setups.

Masked visual modeling. The main goal of masked visual modeling (MVM) is to acquire effective visual representations by the first masking and then reconstructing process. The pioneering work, denoising autoencoders (DAE) [52, 53], learns representations by reconstructing the corrupted signals. Recently, iGPT [7] regards the pixels as tokens and predicts unknown pixels. MAE [17] masks out a subset of patches and learns to predict their original pixels. Other considerations of prediction targets include features [59] and discrete visual tokens [3].

In addition to image recognition, a set of works [11, 50, 56] further extend MAE into video recognition. In this paper, we focus on exploring the potential of MAE in enabling efficient vision-language pre-training.

Efficient vision transformer. Recent works have started investigating eliminating redundant computations in (cumbersome) Vision Transformer (ViT). DynamicViT [44] designs a lightweight prediction module first to estimate the importance scores of tokens, and then utilize the scores to prune redundant tokens hierarchically. EVIT [31] proposes to reorganize tokens by the attention from the class token, *i.e*., they preserve informative tokens while fusing uninformative ones. Motivated by EVIT [31], we also leverage attention from the class token to reduce spatial redundancies.

Temporal redundancy. Giving not all video frames equally contribute to video recognition, many works have been proposed to reduce the temporal redundancies among videos [58, 15, 16, 20, 21, 39, 49]. AR-Net [39] selects the optimal frame resolution conditioned on inputs for efficient video recognition by a decision policy. FrameExit [16] can automatically choose the number of frames to process, according to the complexities of the given videos. This paper proposes a Transformer-based network conditioned on the given video and language features to select informative video frames for reducing temporal redundancies.

3. Method

We first introduce the pre-training model architecture in Section 3.1. Second, we elaborate on the modules to reduce spatial and temporal redundancies in Section 3.2. Third, we describe the pre-training objectives in Section 3.3. Fourth, we list the pre-training datasets in Section 3.4 and implementation details in Section 3.5.

Overview. Given a video clip $V = [v_1, v_2, ..., v_N]$ with N video frames and a text sentence L, the video-language pre-training model needs to extract the visual and textual embeddings separately and feed the embeddings into the multimodal encoder to learn cross-modal representations. A decoder can optionally process the learned cross-modal representations to generate final outputs.

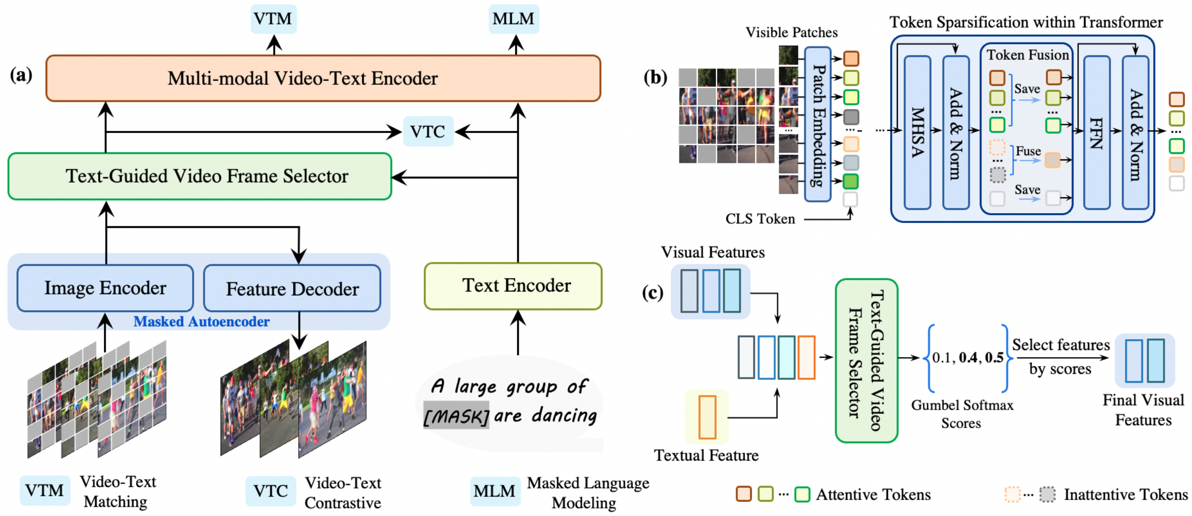

Figure 2: **An illustration of SMAUG architecture.** We use three video frames as an example. (a) The pre-training framework of our proposed SMAUG, we adopt MAE to extract visual features from *visible patches* and reconstruct the pixel values of masked patches. Note that we perform visual token sparsification (*i.e.*, (b)) in the image encoder, and the reconstruction loss for masked patches, *i.e.*, masked visual modeling (MVM), is performed in masked autoencoders. (b) The visual token sparsification module reduces spatial redundancies for *visible patches* of individual video frames, we only utilize single frame input for an explanation. (c) Text-guided video frame selector network takes visual and textual features as inputs and outputs the selected frames by the scores. It can further perform sparsification for video frames along the temporal dimension. The example of selecting two frames among the given video is used for explanation.

3.1. Model Architecture

As illustrated in Figure 2, the proposed SMAUG consists of two major components: *a*) a video-language pre-training model and *b*) space-time redundancy sparsification modules. The video-language pre-training model mainly contains *a*) a masked autoencoder, *b*) a text encoder, and *c*) a multi-modal video-text encoder for cross-modal representation fusion. The space-time redundancy sparsification modules aim to further reduce spatial and temporal redundancies among the visible patches.

3.1.1 Masked Autoencoder

We introduce MAE [17] into video-language pre-training. MAE contains an encoder $\mathcal{Q}_v$ and a decoder $\mathcal{D}_v$. We explain the involved components as follows:

Frame sampling. Following the setup of Singularity [26], we adopt both single-frame and multi-frame options for pre-training. Specifically, we randomly sample one video frame or multiple video frames from the given video clip V to pre-train the video-language models.

Patch embedding. Following ViT [8], we first patchify the sampled frames into non-overlapping patches [50, 11], which are then flattened and projected by a linear layer. The position embeddings [51] are added into the projected patches. In particular, the encoder (*i.e.*, ViT) of MAE [17] operates independently on each frame.

Masking. We randomly mask out a subset of embedded patches for each frame and use the remaining *visible* ones. Note that we perform tube masking [59] for multi-frame input, *i.e.*, the spatial locations of the masked patches are the same for all sampled frames. Although videos contain abundant space-time redundancies, masking out too many patches could lead to inaccurate alignment between visual and textual representations. In our experiments, we at most set the mask ratio to be 65%.

Feature encoding. The encoder $\mathcal{Q}_v$ operates only on the visible embedded patches and omits the masked ones following MAE [17]. The output of the encoder $\mathcal{F}_v$ is a sequence of visual embeddings. When the single-frame input is sampled for pre-training, the output $\mathcal{F}_v$ can be represented as:

$$\mathcal{F}_v = \{f_{cls}, f_1, ..., f_L\}, \tag{1}$$

in which $f_i \in \mathbb{R}^d$, L is the sequence length of visible patches, and d denotes the feature dimension. f_{cls} represents the visual `[CLS]` token. When sampling multiple frames for pre-training, we additionally adopt a temporal encoder $\mathcal{Q}_t$ to perform self-attention on frame-wise features across the temporal dimension. The output $\mathcal{F}_v$ now is:

$$\mathcal{F}_v^k = \{f_{cls}^k, f_1^k, ..., f_L^k\}, \quad k = 1, ..., K, \tag{2}$$

$$\mathcal{F}_v = \mathcal{Q}_t(\mathrm{Concat}(\mathcal{F}_v^1, ..., \mathcal{F}_v^K)), \tag{3}$$

where K is the number of sampled frames. $\mathcal{F}_v$ is adopted for cross-modal alignment.

Feature decoding. After that, we feed the embeddings of *visible patches* and the masked patches together into the decoder D_v for predicting the pixels of these masked ones. This process can be regarded as masked visual modeling (MVM), which has been adopted as one of the pre-training objectives. There exists other reconstruction targets [13] except for pixels, *e.g.*, features, depth maps, *etc*. We leave them in the future.

3.1.2 Text Encoder and Multi-modal Fusion

We then describe the details of the other encoders for textual embedding extraction and cross-modal fusion.

Text encoder. Given the text sentence L, the text encoder $\mathcal{Q}_l$ firstly segments it into a sequence of subwords [46], and inserts the special token (*e.g.*, `[CLS]` token) at the beginning of the subword sequence. This token sequence is then mapped into the text embeddings $\mathcal{F}_l$, which can be represented as:

$$\mathcal{F}_l = \{h_{cls}, h_1, ..., h_P\}, \tag{4}$$

in which $h_i \in \mathbb{R}^d$, h_{cls} is the text embedding of text `[CLS]` token and P means the number of text tokens.

Multi-modal video-text encoder. After obtaining the visual embedding $\mathcal{F}_v$ and text embeddings $\mathcal{F}_l$, we feed them into the multi-modal encoder $\mathcal{Q}_m$ to perform multi-modal fusion by cross-attention layers [27, 26, 19]. The learned cross-modal representations are used for video-text matching and masked language modeling objectives (*i.e.*, VTM and MLM) to pre-train the models.

3.2. Space-Time Token Sparsification

Although MAE has already randomly removed a large subset of frame patches, there still exist spatial and temporal redundancies among the remaining *visible patches* [44, 31, 58, 16], because MAE masks patches randomly without considering how informative they are. In order to make an informative decision, we introduce two context-dependent selection modules, for keeping informative tokens and removing uninformative ones. These modules can further save a large amount of pre-training costs (Table 10).

Visual token sparsification. A considerable amount of uninformative visual tokens (*e.g.*, background patches) will have little impact on the final performance when they are removed [31]. Since the MAE encoder $\mathcal{Q}_v$ inherits the structure of ViT, we can insert a token sparsification module, inspired by EVIT [31], in the 4^{th}, 7^{th} and 10^{th} layers of the ViT-B encoder $\mathcal{Q}_v$. The mechanism of token sparsification within a transformer layer is shown in Figure 2(b).

Specifically, the token sparsification module computes the average attention values among all heads from `[CLS]` with respect to the other tokens. The tokens whose corresponding attention values are top-k largest are regarded as attentive tokens, otherwise inattentive tokens. We retain the attentive tokens while fusing the inattentive tokens into a new token by taking the attention values as coefficients. The keeping rate of the tokens is defined as $\gamma = k/p$, where k and p mean the number of attentive tokens and total input tokens, and the attentive tokens have top $\gamma * 100\%$ average attention values. We explain the details in the Appendix.

Text-guided video frame selection. Similar to the image-level redundancy, videos, as collections of frames, are also redundant on the temporal axis. A fraction of frames could contain more informative contexts than others, especially when corresponding text sentences are given. We hereby aim to select the most context-relevant frames from a given clip to perform video-language tasks.

To this end, we propose a Transformer-based frame selector $\mathcal{S}$, which is illustrated in Figure 2(c). Specifically, it's exemplified by the case of selecting the features of top-2 essential frames. It first takes the visual and textual embeddings from Equation (3) and (4) respectively, and then outputs the features of essential frames. The frame selection process is denoted as:

$$\begin{aligned} \mathcal{I} &= S(\mathcal{F}_v; \mathcal{F}_l) \\ &= \{\mathcal{F}_v^{\mu_1}, ..., \mathcal{F}_v^{\mu_\kappa}\}, \quad i = 1, ..., \kappa. \end{aligned} \tag{5}$$

In Equation (5), $\mathcal{I} \in \mathbb{R}^{\kappa \times L' \times d}$ ($1 <= \kappa < K$) denotes the sparse collection of video frame features selected from the original visual embeddings $\mathcal{F}_v$, and L' is the sequence length of embedded patch embeddings. Since the frame selection process is discrete, we adopt the Gumbel-Softmax trick [18] to optimize the parameters of selector $\mathcal{S}$ in both the full pre-training stage and the task-specific fine-tuning stage. At inference time, the operation is discrete by selecting the top-k ($k = \kappa$) features as the final output. In this way, we obtain a sparse and efficient model.

3.3. Pre-training Objectives

Our models are pre-trained by using four objectives: (1) Video-Text Matching ($\mathcal{L}_{vtm}$): it predicts the matching scores of the given video-text pairs through the multi-modal video-text encoder's outputs. (2) Masked Language Modeling ($\mathcal{L}_{mlm}$): it fuses both visual and textual features by the multi-modal video-text encoder to predict the masked textual tokens. (3) Video-Text Contrastive ($\mathcal{L}_{vtc}$): it uses the pooled visual and textual representations to align parallel video-text pairs, so that they can have higher similarity scores. (4) Masked Visual Modeling ($\mathcal{L}_{mvm}$): it learns to predict the original pixels of masked visual patches.

These objectives have been widely used for pre-training video-language models [26, 12, 54, 13, 28, 9], we describe the details of them in the Appendix. The full pre-training objective $\mathcal{L}$ is the simple addition of these four objectives:

$$\mathcal{L} = \mathcal{L}_{vtm} + \mathcal{L}_{mlm} + \mathcal{L}_{vtc} + \mathcal{L}_{mvm}. \tag{6}$$

3.4. Pre-training Datasets

We adopt video-text and image-text data for pre-training. WebVid [2] is used as video-text data, which scrapes 2.5M video-text pairs from the web. The image-text data includes the combination of CC3M [47], CC12M [6], SBU Captions [41], Visual Genome (VG) [23] and COCO [32] datasets. Two subsets are employed for pre-training: 1) 5M corpus consisting of CC3M and WebVid. 2) 17M corpus that contains all the mentioned image-text and video-text datasets. Note that for the single frame setting, we sample only one frame from the video-text dataset, *i.e.*, WebVid.

3.5. Implementation Details

Network structures. For the masked autoencoder, we adopt ViT-B [8] as the encoder $\mathcal{Q}_v$, whose weights are initialized from CLIP's visual encoder (*i.e.*, CLIP-B/16) [43]. The decoder $\mathcal{D}_v$ is the stack of one linear layer, three Transformer blocks, and one linear layer. We adopt BERT$_{BASE}$ model as our text encoder $\mathcal{Q}_l$ and multi-modal video-text encoder $\mathcal{Q}_m$. In particular, the first 9 layers and the last three layers of BERT$_{BASE}$ are initialized as $\mathcal{Q}_l$ and $\mathcal{Q}_m$, respectively. The cross-attention layers [27, 26] of the multimodal encoder are learned from scratch. The frame selector $\mathcal{S}$ consists of two linear layers and two Transformer blocks [51]. The Transformer blocks have two heads and a hidden size of 2048.

Model pre-training. We use 4×NVIDIA A6000 GPUs to pre-train our models with AdamW optimizer [36] and an initial learning rate of $1\mathrm{e}^{-4}$. We randomly sample one frame for the single-frame setting and four frames for the multiple-frame setting. The total pre-training epochs are 10, and the learning rate is warmed up in the first epoch, followed by cosine decay [35] to $1\mathrm{e}^{-6}$ finally. The input size of the video frames is 224×224. The data augmentation includes random resized crop and flip operations. When using the single frame for pre-training, our model only takes about 13 hours to pre-train on the 5M corpus and two days on the 17M corpus, attaining better performance than ALPRO [28], which takes three days to pre-train the same epochs (*i.e.*, 10 epochs) on the 5M corpus with 16×A100 GPUs. It is worth mentioning that when using multiple video frames, *i.e.*, four frames, for pre-training, the model weights are initialized from single-frame pre-trained models, and the total training epoch for such setup is 5.

4. Experiment

4.1. Down-Stream Tasks and Datasets

Text-to-video retrieval. We first evaluate our approach on text-to-video retrieval task across three video-language datasets. The recall performance at K (R@K) is reported.

- **MSRVTT** [62] consists of 10K videos from YouTube, and each video has 20 textual captions. We follow the standard setting as [26, 28, 40, 64], *i.e.*, we fine-tune the pre-trained models with 7K videos and report the performances on the 1K test split [64].
- **DiDeMo** [1] includes 10K videos, which are collected from Flickr; there are 41K text descriptions total, and the train/val/test splits are adopted.
- **ActivityNet Captions** [22] consists of 20K YouTube videos which are paired with 100K captions. We evaluate the results on *val1* split.

For MSRVTT, we report the results of the standard text-to-video retrieval protocol, as for the other two datasets, we perform paragraph-to-video retrieval evaluation [26, 28], *i.e.*, we concatenate all the captions in the same video to one single paragraph for retrieval.

Video question answering. We also select video question answering for evaluation, the accuracies are reported, and three benchmarks are chosen:

- **MSRVTT-MC** [64] is a dataset for multiple-choice question answering, which consists of 3K videos and needs to select the best matching caption choice from 5 candidates for each video.
- **MSRVTT-QA** [60] is built on MSRVTT, consisting of 10K videos paired with 244K open-ended questions.
- **ActivityNet-QA** [65] collects 5.8K videos from ActivityNet [5] with 58K open-ended questions.

Fine-tuning setups. When fine-tuning the models on text-to-video retrieval task, we use the same model architecture except that we remove MLM and MVM objectives. The models are trained with an initial learning rate of $1\mathrm{e}^{-5}$, which is decreased to $1\mathrm{e}^{-6}$ by cosine decay; the training epochs are 5, 10 and 10 for MSRVTT, DiDeMo, and ActivityNet-Captions datasets. During the inference stage, we sample 12 frames from each video for MSRVTT and DiDeMo datasets and 32 frames per video for the ActivityNet Captions dataset.

To evaluate the performances on open-ended video question answering datasets (*e.g.*, MSRVTT-QA and ActivityNet-QA), an extra decoder is added after the multi-encoder, which generates the answers by taking the outputs from the multi-modal encoder. The models are fine-tuned with 10 epochs and tested with 12 frames. As for MSRVTT-MC, we follow the protocol of [27]. Specifically, the models trained on MSRVTT are leveraged to select the best matching choice with the highest retrieval scores as final predictions. Note that for all downstream tasks, we resize the video frames as 224×224, and the data augmentation is the same as the pre-training process.

Method	PT Datasets	#Frame	MSRVTT			DiDeMo			ActivityNet Cap		
			R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Pre-trained with >100M video-text pairs											
HT100M [40]	HT100M	16	14.9	40.2	52.8	-	-	-	-	-	-
HERO [29]	HT100M	310	20.5	47.6	60.9	-	-	-	-	-	-
MMT [14]	HT100M	1K/-/3K	26.6	57.1	69.6	-	-	-	28.7	61.4	94.5
AVLNet [45]	HT100M	-	27.1	55.6	66.6	-	-	-	-	-	-
SupportSet [42]	HT100M	-	30.1	58.5	69.3	-	-	-	-	-	-
VideoCLIP [61]	HT100M	960	30.9	55.4	66.8	-	-	-	-	-	-
VIOLET [12]	YT180M+5M	4	34.5	63.0	73.4	32.6	62.8	74.7	-	-	-
All-in-one [54]	HT100M+WebVid	9	34.4	65.4	75.8	32.7	61.4	73.5	22.4	53.7	67.7
Pre-trained with <100M video-text pairs											
ClipBERT [27]	COCO + VG	16/16/8	22.0	46.8	59.9	20.4	48.0	60.8	21.3	49.0	63.5
Frozen [2]	5M	4	31.0	59.5	70.5	31.0	59.8	72.4	-	-	-
ALPRO [28]	5M	8	33.9	60.7	73.2	35.9	67.5	78.8	-	-	-
Singularity [26]	5M	1	36.8	65.9	75.5	47.4	75.2	84.0	43.0	70.6	81.3
Singularity [26]	17M	1	41.5	68.7	77.0	53.9	79.4	86.9	47.1	75.5	85.5
Ours	**5M**	**1**	**40.6**	**67.6**	**77.5**	**49.2**	**76.7**	**85.6**	**44.8**	**72.2**	**82.7**
Ours	**17M**	**1**	**44.0**	**70.4**	**78.8**	**55.6**	**80.8**	**88.4**	**49.2**	**76.9**	**86.8**

Table 1: Fine-tuning results compared with existing video-language pre-training methods on text-to-retrieval tasks. The pre-training (PT) text-video datasets are HowTo100M (HT100M) [40], YT-Temporal-180M (YT180M)[66], MS-COCO (COCO) [32], Visual Genome (VG) [23], 5M corpus and 17M corpus. Note that 5M and 17M settings are illustrated in Section 3.4. For MSRVTT dataset, results using 9K training videos are **greyed out**. The methods using other modalities (*e.g.*, speech and audio) are highlighted by pink. "#Frame" denotes the number of final video frames for video-language model pre-training. For methods that adopt different input frames for different datasets, we use "/" to separate them.

Method	MSRVTT			DiDeMo		
	R@1	R@5	R@10	R@1	R@5	R@10
MMT [14]	-	6.9	-	-	-	-
AVLNet [45]	19.6	40.8	50.7	-	-	-
VideoCLIP [61]	10.4	22.2	30.0	16.6	46.9	-
Frozen [2]	18.7	39.5	51.6	21.1	46.0	56.2
ALPRO [28]	24.1	44.7	55.4	23.8	47.3	57.9
VIOLET [12]	25.9	49.5	59.7	23.5	49.8	59.8
Singularity [26]	28.4	50.2	59.5	36.9	61.1	69.3
Ours	**28.9**	**52.1**	**62.9**	**34.3**	**60.3**	**69.4**

Table 2: Zero-shot results compared with existing video-language pre-training methods on text-to-retrieval tasks. The pre-training text-video datasets and the number of input video frames are the same as Table 1. For a clear comparison, we display the results of Singurity [26] and our SMAUG, whose number of the final video frames is 1, and the pre-training corpus is 5M.

4.2. Comparison with State-of-the-arts

Here, we show the performances of the proposed approach, the default masking ratios of MAE and masked language modeling (MLM) are 50% and 15%, and the keeping rate γ is 0.8. The number of input frames keeps unchanged during the pre-training and fine-tuning stages.

Text-to-video retrieval. In Table 1, we show the compared results on text-to-video retrieval task. When the final video frame number is 1 and using a smaller scale of pre-trained data (*e.g.*, 5M and 17M), our method can still significantly surpass existing methods, even compared with other approaches (*e.g.*, HERO [29], MMT [14] and AVLNet [45]) that adopt other modalities, such as audio, speech, *etc*.

As for MSRVTT dataset, when the pre-training corpus is 5M and the final input frame number is 1, SMAUG can achieve **3.8%** relative improvement on R@1 compared with Singularity [26], when pre-trained on the 17M corpus, SMAUG can also lead to **2.5%** performance gain on R@1 compared with Singularity. These results can demonstrate the effectiveness of our approach. Note that our pre-training is also much faster compared with Singularity [26]. The analysis of pre-training costs is shown in Section 4.4.

In Table 2, we show the zero-shot results to compare our approach with existing methods. Our SMAUG can achieve comparable performances compared with Singularity [26]. The results can demonstrate that even with **50%** mask ratios for masked autoencoder, the pre-trained model can still learn meaningful cross-modal representations for video-language tasks.

Video question answering. We display the results of video question answering in Table 3. When using 5M and 17M settings for pre-training, our proposed approach can still perform better than Singularity [26].

We can also observe that, when pre-training the models on the 17M corpus, our method can still surpass the performances of VIOLET [12] on MSRVTT-QA and MSRVTT-MC benchmarks, note that we use less video-text data for pre-training (17M *v.s.* YT180M+5M).

4.3. Ablation Study

In this section, we ablate the influences of the proposed components. The pre-training corpus is 5M. We report the fine-tuning results on the text-video-retrieval task, and the

Method	PT Datasets	#Frame	MSRVTT-QA	ActivityNet-QA	MSRVTT-MC
Pre-trained with >100M video-text pairs					
JustAsk [63]	HT69M	640	41.5	38.9	-
MERLOT [66]	YT180M	5	43.1	41.4	90.9
VideoCLIP [61]	HT100M	960	-	-	92.1
VIOLET [12]	YT180M+5M	4	43.9	-	91.9
Pre-trained with <100M video-text pairs					
ClipBERT [27]	COCO + VG	16	37.4	-	88.2
ALPRO [28]	5M	16	42.1	-	-
Singularity [26]	5M	1	42.7	41.8	92.0
Singularity [26]	17M	1	43.5	43.1	92.1
Ours	**5M**	**1**	**43.4**	**42.7**	**92.7**
Ours	**17M**	**1**	**44.5**	**44.2**	**92.9**

Table 3: Results compared with existing video-language pre-training methods on video question answering task. The pre-training text-video datasets are HowTo100M (HT100M) [40], YT-Temporal-180M (YT180M)[66], MS-COCO (COCO) [32], Visual Genome (VG) [23], HowToVQA69M (HT69M) [63], 5M corpus and 17M corpus. Note that 5M and 17M settings are illustrated in Section 3.4. "#Frame" denotes the number of final video frames for video-language model pre-training.

Masking Ratio	PT Time	MSRVTT R@1	MSRVTT R@5	MSRVTT R@10
0%	74.9 hours	**40.7**	**66.6**	**76.7**
10%	70.8 hours	40.5	66.3	76.5
25%	64.6 hours	40.1	65.7	76.0
50%	50.4 hours	39.3	65.4	75.6
65%	**44.3 hours**	38.1	64.2	75.0

Table 4: Ablation study on using different masking ratios.

Keeping Rate	PT Time	MSRVTT R@1	MSRVTT R@5	MSRVTT R@10
0.6	**44.8 hours**	37.5	64.2	74.6
0.7	47.6 hours	38.2	64.7	75.3
0.8	50.4 hours	39.3	65.4	75.6
0.9	53.8 hours	**39.8**	**65.9**	**76.0**

Table 5: Ablation study on using different keeping rates.

Frame Selection	PT Time	MSRVTT R@1	MSRVTT R@5	MSRVTT R@10
Single-frame selection				
1	**50.4 hours**	39.3	65.4	75.6
4→1	75.3 hours	**40.6**	**67.6**	**77.5**
Multiple-frame selection				
4→2	**77.8 hours**	41.2	67.8	77.9
4→3	80.2 hours	41.5	68.0	78.0
4	82.5 hours	41.7	68.3	78.3
8→4	100.3 hours	**42.8**	**69.2**	**79.5**

Table 6: Ablation study on selecting different frames by the proposed selector $\mathcal{S}$.

Method	MSRVTT R@1	MSRVTT R@5	MSRVTT R@10
Singularity* [26]	36.8	65.9	75.5
Ours*	**38.9**	**67.0**	**76.8**
Ours+	**40.6**	**67.6**	**77.5**

Table 7: Ablation study on different visual encoders. * and + mean adopting BEiT-Base [3] and CLIP-B/16 [43].

MSRVTT dataset is chosen. Except for the ablation studies of Table 6, 8 and 10, the number of the final input frames is 1 for the experiments. *"PT Time"* denotes the total time for model pre-training, we report the GPU hours of using a single A6000 GPU. The results highlighted in blue are used to specify the default settings.

Impact of masking ratios. To figure out the influence of masking ratios of MAE for video-language pre-training, we show the results in Table 4, when the masking ratio is 50%, the pre-training time can be efficiently reduced from **70.8** hours, whose masking ratio is 10%, to **50.4** hours, while losing only **1.2%** on R@1, which can demonstrate the potential of extending MAE for video-language pre-training.

Effect of keeping rates. The results of adopting different keeping rates for visual token sparsification (VTS) are shown in Table 5. When the keeping rate is 0.8 (*i.e.*, 80%), the pre-training time can be reduced from **53.8** hours, whose keeping rate is 0.9, to **50.4** hours, while only losing **0.5%** on R@1. The saving of pre-training time is promising, considering the masked autoencoder already removes plenty of visual patches and the number of the input frames is only 1.

Effectiveness of frame selection. We display the results of frame selection (FS) in Table 6. When selecting one frame among 4 video frames (*i.e.*, 4→1), the performance can be improved from **39.3%** to **40.6%** on R@1. When multiple frames are selected, for example, selecting 2 frames among 4 video frames (*i.e.*, 4→2), the loss on R@1 is only **0.5%** compared with pre-training using 4 video frames, while reducing the pre-training time from **82.5** hours to **77.8** hours. Note that pre-training with the four-frame input can lead to a **2.4%** improvement in R@1 but increase **32.1** pre-training hours compared to the single-frame input setting.

Impact of different visual encoders. In Table 7, we report results of using different visual encoders, Singularity [26] which adopts BEiT-Base [3] by default is compared, SMAUG can achieve better performances than Singularity.

Impact of different strategies for the proposed modules. In Table 8, we report the results of using different strate-

Strategy	MAE	VTS	FS	MSRVTT R@1	MSRVTT R@5	MSRVTT R@10
Random	✓			39.8	66.5	76.8
Tube (Ours)	✓			**41.2**	**67.8**	**77.9**
Random		✓		39.5	66.4	76.6
Ours		✓		**41.2**	**67.8**	**77.9**
Random			✓	38.9	65.3	75.4
CLIP-based			✓	39.7	66.3	76.5
Ours			✓	**41.2**	**67.8**	**77.9**

Table 8: Ablation study on using different strategies.

VTS	Inference Time (Per Example)	MSRVTT R@1	MSRVTT R@5	MSRVTT R@10
✗	0.21 seconds	39.3	65.4	75.6
✓	**0.16 seconds**	39.1	65.3	75.4

Table 9: Ablation study on adopting VTS for inference.

Caption: A man is talking

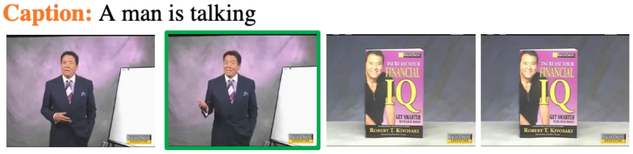

Figure 3: **An example of frame selection.** The frame denoted by the *green* box means the final selected frame.

gies for our proposed modules, where we can see that ours consistently outperforms Random or CLIP-based strategy.

Impact of VTS on inference efficiency. Since VTS can benefit models' inference, we fine-tune the models and report their inference time in Table 9. By using VTS, the inference time per example can be reduced from **0.21** seconds to **0.16** seconds with only **0.2%** performance drop in R@1.

Impact of different components. In Table 10, we show the results of adopting different components to figure out their influences. The final pre-training time can be consistently reduced from **144.5** hours to **77.8** hours (**1.9**×speedup), while only losing **1.4%** on R@1, which can show the efficiencies of the proposed components.

4.4. Analysis of SMAUG

Pre-training Datasets and Costs. The pre-training efficiency of our proposed method can be observed in Table 11. We report the pre-training time of Singularity [26] by using their official codes. When adopting 5M and 17M corpus for pre-training, our approach can consistently use less pre-training time while achieving better performances. For example, when pre-trained on the 17M corpus, our approach can obtain **2.5%** performance gain on R@1 while saving **87.1** pre-training hours compared with Singularity, demonstrating the strong scalability of our method.

Visualization. Finally, we showcase the example of frame selection by selector $\mathcal{S}$ in Figure 3. It can be observed that there're abundant video frames in a video clip which not correspond to the given caption, removing these chaotic frames are reasonable. Our approach can accurately select the most essential frame according to the given caption.

MAE	VTS	FS	PT Time	MSRVTT R@1	MSRVTT R@5	MSRVTT R@10
✗	✗	✗	144.5 hours	42.6	68.8	79.2
✓	✗	✗	93.5 hours	42.0	68.4	78.7
✓	✓	✗	82.5 hours	41.6	67.9	78.3
✓	✓	✓	**77.8 hours**	41.2	67.8	77.9

Table 10: Ablation study on using different components of SMAUG on the multiple-frame setting (*i.e.*, "4→2").

Method	PT Time	MSRVTT R@1	MSRVTT R@5	MSRVTT R@10
Singularity* [26]	83.4 hours	36.8	65.9	75.5
Ours*	**75.3 hours**	**40.6**	**67.6**	**77.5**
Singularity+ [26]	285.3 hours	41.5	68.7	77.0
Ours+	**198.2 hours**	**44.0**	**70.4**	**78.8**

Table 11: Pre-training with different scales of the video-text corpus. "*" and "+" mean the pre-trained models with 5M and 17M corpus respectively. We report the results pre-trained with the single-frame setting.

Caption: Aerial shot of tractor raking grass for combine to silage

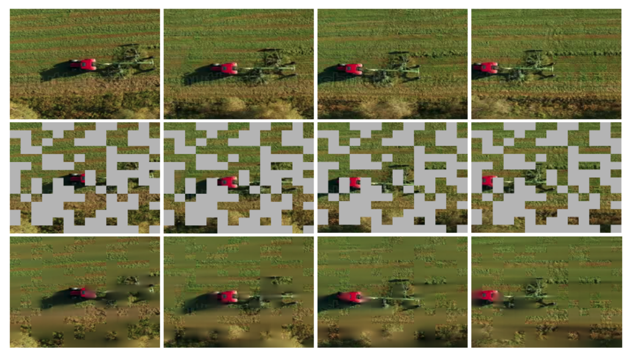

Figure 4: **The examples of pixel prediction for masked patches.** The top row represents the original frames, the middle row means the masked frames, and the bottom row denotes the pixel reconstruction for masked patches.

In Figure 4, we display the qualitative examples of pixel reconstruction for masked patches. Even with a masking ratio of 50%, the model can still produce reliable predictions for masked patches, which can further demonstrate mask autoencoders can help us achieve satisfactory performances and efficiently mitigate the computational burden of video-language model pre-training.

5. Conclusion

In this paper, we propose an efficient video-language pre-training method, SMAUG, which is built on masked autoencoders. We additionally develop a space-time sparsification module to remove the spatial and temporal redundancies among the remaining visible patches. Our SMAUG can achieve state-of-the-art or comparable performances on two video-language tasks across six popular benchmarks, while obtaining 1.9× pre-training time speedup.

Acknowledgement. CW and AY is supported by ONR N00014-21-1-2690 and Amazon award to JHU AI2AI.

References

[1] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pages 5803–5812, 2017. 1, 2, 5

[2] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1728–1738, 2021. 1, 5, 6

[3] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 2, 7

[4] Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. Revisiting the" video" in video-language understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2917–2927, 2022. 1

[5] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015. 5

[6] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pretraining to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3558–3568, 2021. 1, 5

[7] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020. 2

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3, 5

[9] Zi-Yi Dou, Yichong Xu, Zhe Gan, Jianfeng Wang, Shuohang Wang, Lijuan Wang, Chenguang Zhu, Pengchuan Zhang, Lu Yuan, Nanyun Peng, et al. An empirical study of training end-to-end vision-and-language transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18166–18176, 2022. 4

[10] Maksim Dzabraev, Maksim Kalashnikov, Stepan Komkov, and Aleksandr Petiushko. Mdmmt: Multidomain multimodal transformer for video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3354–3363, 2021. 2

[11] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. *arXiv preprint arXiv:2205.09113*, 2022. 1, 2, 3

[12] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. Violet: End-to-end video-language transformers with masked visual-token modeling. *arXiv preprint arXiv:2111.12681*, 2021. 1, 2, 4, 6, 7

[13] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. An empirical study of end-to-end video-language transformers with masked visual modeling. *arXiv preprint arXiv:2209.01540*, 2022. 1, 4

[14] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *European Conference on Computer Vision*, pages 214–229. Springer, 2020. 6

[15] Ruohan Gao, Tae-Hyun Oh, Kristen Grauman, and Lorenzo Torresani. Listen to look: Action recognition by previewing audio. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10457–10467, 2020. 2

[16] Amir Ghodrati, Babak Ehteshami Bejnordi, and Amirhossein Habibian. Frameexit: Conditional early exiting for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15608–15618, 2021. 2, 4

[17] 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. 1, 2, 3

[18] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 4

[19] Haojun Jiang, Yuanze Lin, Dongchen Han, Shiji Song, and Gao Huang. Pseudo-q: Generating pseudo language queries for visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15513–15523, 2022. 4

[20] Hanul Kim, Mihir Jain, Jun-Tae Lee, Sungrack Yun, and Fatih Porikli. Efficient action recognition via dynamic knowledge propagation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13719–13728, 2021. 2

[21] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6232–6242, 2019. 2

[22] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715, 2017. 2, 5

[23] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 5, 6, 7

[24] Gukyeong Kwon, Zhaowei Cai, Avinash Ravichandran, Erhan Bas, Rahul Bhotika, and Stefano Soatto. Masked vision and language modeling for multi-modal representation learning. *arXiv preprint arXiv:2208.02131*, 2022. 1

[25] Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. Hierarchical conditional relation networks for video question answering. In *Proceedings of the IEEE/CVF con-*

ference on computer vision and pattern recognition, pages 9972–9981, 2020. 2

[26] Jie Lei, Tamara L Berg, and Mohit Bansal. Revealing single frame bias for video-and-language learning. *arXiv preprint arXiv:2206.03428*, 2022. 2, 3, 4, 5, 6, 7, 8

[27] 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. 1, 2, 4, 5, 6, 7

[28] Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven CH Hoi. Align and prompt: Video-and-language pre-training with entity prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4953–4963, 2022. 4, 5, 6, 7

[29] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. *arXiv preprint arXiv:2005.00200*, 2020. 1, 6

[30] Yicong Li, Xiang Wang, Junbin Xiao, Wei Ji, and Tat-Seng Chua. Invariant grounding for video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2928–2937, 2022. 2

[31] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022. 1, 2, 4

[32] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 5, 6, 7

[33] Yuanze Lin, Xun Guo, and Yan Lu. Self-supervised video representation learning with meta-contrastive network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8239–8249, 2021. 1

[34] Yuanze Lin, Yujia Xie, Dongdong Chen, Yichong Xu, Chenguang Zhu, and Lu Yuan. Revive: Regional visual representation matters in knowledge-based visual question answering. *arXiv preprint arXiv:2206.01201*, 2022. 1

[35] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[37] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020. 2

[38] 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. 1, 2

[39] Yue Meng, Chung-Ching Lin, Rameswar Panda, Prasanna Sattigeri, Leonid Karlinsky, Aude Oliva, Kate Saenko, and Rogerio Feris. Ar-net: Adaptive frame resolution for efficient action recognition. In *European Conference on Computer Vision*, pages 86–104. Springer, 2020. 2

[40] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640, 2019. 5, 6, 7

[41] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems*, 24, 2011. 5

[42] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander Hauptmann, Joao Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning. *arXiv preprint arXiv:2010.02824*, 2020. 6

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2, 5, 7

[44] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021. 1, 2, 4

[45] Andrew Rouditchenko, Angie Boggust, David Harwath, Brian Chen, Dhiraj Joshi, Samuel Thomas, Kartik Audhkhasi, Hilde Kuehne, Rameswar Panda, Rogerio Feris, et al. Avlnet: Learning audio-visual language representations from instructional videos. *arXiv preprint arXiv:2006.09199*, 2020. 6

[46] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015. 4

[47] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018. 1, 5

[48] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. *arXiv preprint arXiv:1811.00491*, 2018. 1

[49] Ximeng Sun, Rameswar Panda, Chun-Fu Richard Chen, Aude Oliva, Rogerio Feris, and Kate Saenko. Dynamic network quantization for efficient video inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7375–7385, 2021. 2

[50] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *arXiv preprint arXiv:2203.12602*, 2022. 2, 3

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia

Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3, 5

[52] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 2

[53] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010. 2

[54] Alex Jinpeng Wang, Yixiao Ge, Rui Yan, Yuying Ge, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. All in one: Exploring unified video-language pre-training. *arXiv preprint arXiv:2203.07303*, 2022. 1, 2, 4, 6

[55] Bairui Wang, Lin Ma, Wei Zhang, and Wei Liu. Reconstruction network for video captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7622–7631, 2018. 2

[56] Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiyang Dai, Mengchen Liu, Yu-Gang Jiang, Luowei Zhou, and Lu Yuan. Bevt: Bert pretraining of video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14733–14743, 2022. 2

[57] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4581–4591, 2019. 2

[58] Yulin Wang, Yang Yue, Yuanze Lin, Haojun Jiang, Zihang Lai, Victor Kulikov, Nikita Orlov, Humphrey Shi, and Gao Huang. Adafocus v2: End-to-end training of spatial dynamic networks for video recognition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20030–20040. IEEE, 2022. 1, 2, 4

[59] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14668–14678, 2022. 2, 3

[60] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1645–1653, 2017. 2, 5

[61] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *arXiv preprint arXiv:2109.14084*, 2021. 6, 7

[62] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016. 1, 2, 5

[63] 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. 7

[64] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 471–487, 2018. 2, 5

[65] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9127–9134, 2019. 2, 5

[66] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models. *Advances in Neural Information Processing Systems*, 34:23634–23651, 2021. 6, 7

[67] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5579–5588, 2021. 1

[68] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2

[69] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8746–8755, 2020. 2