SViTT: Temporal Learning of Sparse Video-Text Transformers

Yi Li*[1] Kyle Min[2] Subarna Tripathi[2] Nuno Vasconcelos[1]

[1]UC San Diego [2]Intel Labs

{yil898, nvasconcelos}@ucsd.edu {kyle.min, subarna.tripathi}@intel.com

Abstract

Do video-text transformers learn to model temporal relationships across frames? Despite their immense capacity and the abundance of multimodal training data, recent work has revealed the strong tendency of video-text models towards frame-based spatial representations, while temporal reasoning remains largely unsolved. In this work, we identify several key challenges in temporal learning of video-text transformers: the spatiotemporal trade-off from limited network size; the curse of dimensionality for multi-frame modeling; and the diminishing returns of semantic information by extending clip length. Guided by these findings, we propose ***SViTT****, a sparse video-text architecture that performs multi-frame reasoning with significantly lower cost than naïve transformers with dense attention. Analogous to graph-based networks,* ***SViTT*** *employs two forms of sparsity: edge sparsity that limits the query-key communications between tokens in self-attention, and node sparsity that discards uninformative visual tokens. Trained with a curriculum which increases model sparsity with the clip length,* ***SViTT*** *outperforms dense transformer baselines on multiple video-text retrieval and question answering benchmarks, with a fraction of computational cost. Project page:* http://svcl.ucsd.edu/projects/svitt.

1. Introduction

With the rapid development of deep neural networks for computer vision and natural language processing, there has been growing interest in learning correspondences across the visual and text modalities. A variety of vision-language pretraining frameworks have been proposed [12, 22, 29, 34] for learning high-quality cross-modal representations with weak supervision. Recently, progress on visual transformers (ViT) [5, 16, 32] has enabled seamless integration of both modalities into a unified attention model, leading to image-text transformer architectures that achieve state-the-art performance on vision-language benchmarks [1, 27, 44].

Progress has also occurred in *video*-language pretraining by leveraging image-text models for improved frame-based reasoning [4, 9, 18]. Spatial modeling has the advantage of efficient (linear) scaling to long duration videos. Perhaps due to this, single-frame models have proven surprisingly effective at video-text tasks, matching or exceeding prior arts with complex temporal components [9, 24]. However, spatial modeling creates a bias towards static appearance and overlooks the importance of temporal reasoning in videos. This suggests the question: Are temporal dynamics not worth modeling in the video-language domain?

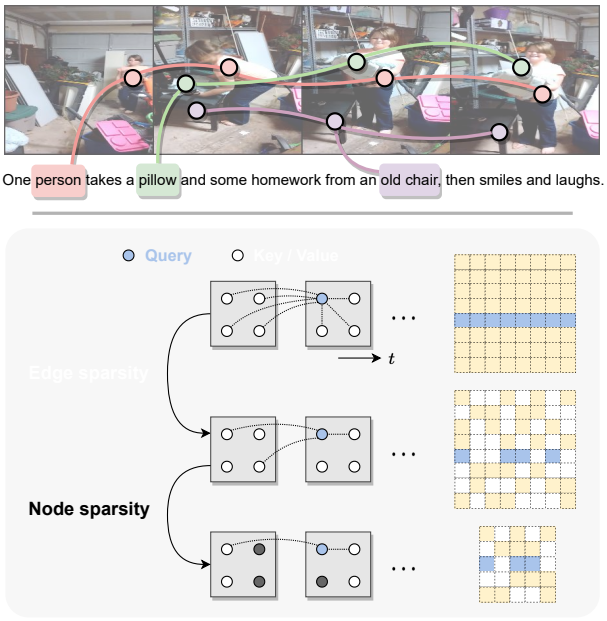


Figure 1. We propose **SViTT**, a *sparse* video-text transformer for efficient modeling of temporal relationships across video frames. **Top**: Semantic information for video-text reasoning is highly localized in the spatiotemporal volume, making dense modeling inefficient and prone to contextual noises. **Bottom**: **SViTT** pursues *edge* sparsity by limiting query-key pairs in self-attention, and *node* sparsity by pruning redundant tokens from visual sequence.

Upon a closer investigation, we identify a few key challenges to incorporating multi-frame reasoning in video-language models. First, limited model size implies a trade-off between spatial and temporal learning (a classic example being 2D/3D convolutions in video CNNs [46]). For any given dataset, optimal performance requires a careful bal-

*Work done during an internship at Intel Labs.

ance between the two. Second, long-term video models typically have larger model sizes and are more prone to overfitting. Hence, for longer term video models, it becomes more important to carefully allocate parameters and control model growth. Finally, even if extending the clip length improves the results, it is subject to diminishing returns since the amount of information provided by a video clip does not grow linearly with its sampling rate. If the model size is not controlled, the computational increase may not justify the gains in accuracy. This is critical for transformer-based architectures, since self-attention mechanisms have a quadratic memory and time cost with respect to input length. In summary, model complexity should be adjusted adaptively, depending on the input videos, to achieve the best trade-off between spatial representation, temporal representation, overfitting potential, and complexity. Since existing video-text models lack this ability, they either attain a suboptimal balance between spatial and temporal modeling, or do not learn meaningful temporal representations at all.

Motivated by these findings, we argue that video-text models should learn to allocate modeling resources to the video data. We hypothesize that, rather than uniformly extending the model to longer clips, the allocation of these resources to the relevant spatiotemporal locations of the video is crucial for efficient learning from long clips. For transformer models, this allocation is naturally performed by pruning redundant attention connections. We then propose to accomplish these goals by exploring transformer sparsification techniques. This motivates the introduction of a *Sparse Video-Text Transformer* (**SViTT**) inspired by graph models. As illustrated in Fig. 1, **SViTT** treats video tokens as graph vertices, and self-attention patterns as edges that connect them. We design **SViTT** to pursue sparsity for both: *edge* sparsity aims at reducing query-key pairs in attention module while maintaining its global reasoning capability; *node* sparsity reduces to identifying informative tokens (e.g., corresponding to moving objects or person in the foreground) and pruning background feature embeddings. To address the diminishing returns for longer input clips, we propose to train **SViTT** with *temporal sparse expansion*, a curriculum learning strategy that increases clip length and model sparsity, in sync, at each training stage.

SViTT is evaluated on diverse video-text benchmarks from video retrieval to question answering, comparing to prior arts and our own dense modeling baselines. First, we perform a series of ablation studies to understand the benefit of sparse modeling in transformers. Interestingly, we find that both nodes (tokens) and edges (attention) can be pruned drastically at inference, with a small impact on test performance. In fact, token selection using cross-modal attention improves retrieval results by 1% without re-training.

We next perform full pre-training with the sparse models and evaluate their downstream performance. We observe that **SViTT** scales well to longer input clips where the accuracy of dense transformers drops due to optimization difficulties. On all video-text benchmarks, **SViTT** reports comparable or better performance than their dense counterparts with lower computational cost, outperforming prior arts including those trained with additional image-text corpora.

The key contributions of this work are: 1) a video-text architecture **SViTT** that unifies edge and node sparsity; 2) a sparse expansion curriculum for training **SViTT** on long video clips; and 3) empirical results that demonstrate its temporal modeling efficacy on video-language tasks.

2. Related Work

Video-language pretraining. Vision-language pretraining has been widely adopted for various video-text downstream tasks. VideoBERT [45] was an early effort, using video-text pretraining for action classification and video captioning. Recently, the massive-scale instructional video dataset HowTo100M [39] has motivated many approaches to video-text pretraining [4, 19, 24, 38, 56, 57]. Frozen [4] proposed to pretrain a space-time transformer on a combination of video and image data to enable zero-shot text-to-video retrieval. ATP [9] and Singularity [24] showed strong performance using image-based models, highlighting the importance of spatial modeling for video-language tasks. In this work, we pursue an alternative route of efficient *temporal* modeling across multiple video frames.

Sparse transformers. The self-attention of naïve transformers [16, 48] has quadratic complexity making them inefficient for modeling long sequences. Different forms of sparse attention have been studied to improve text [7, 13, 55], image [15, 32, 47], and video modeling [3, 8, 33], although the sparse patterns are typically predetermined and do not adapt to the input semantics. Several works have also considered speeding up vision transformers by adaptively reducing the number of input tokens [11, 30, 37, 41, 42, 53]. For example, DynamicViT [41] proposed to drop visual tokens with a dedicated module that identifies and prunes less informative ones. TokenLearner [42] introduces a learnable module to adaptively generate a small subset of tokens from input frames. EViT [30] progressively reduces the number of tokens based on their attention scores, fusing inattentive tokens into a new token to preserve input information. Unlike prior works that focus on visual modeling on images or videos alone, we study the sparsity of video-language transformers which can benefit from cross-modal attention.

3. Exploiting Sparsity in Video Transformers

In this section, we formulate video transformers as graph models (Sec. 3.1) and present a set of approaches towards sparse video modeling, exploiting the redundancy of edges (Sec. 3.2) and nodes (Sec. 3.3). We combine these into a unified sparse framework for video-text learning (Sec. 3.4).

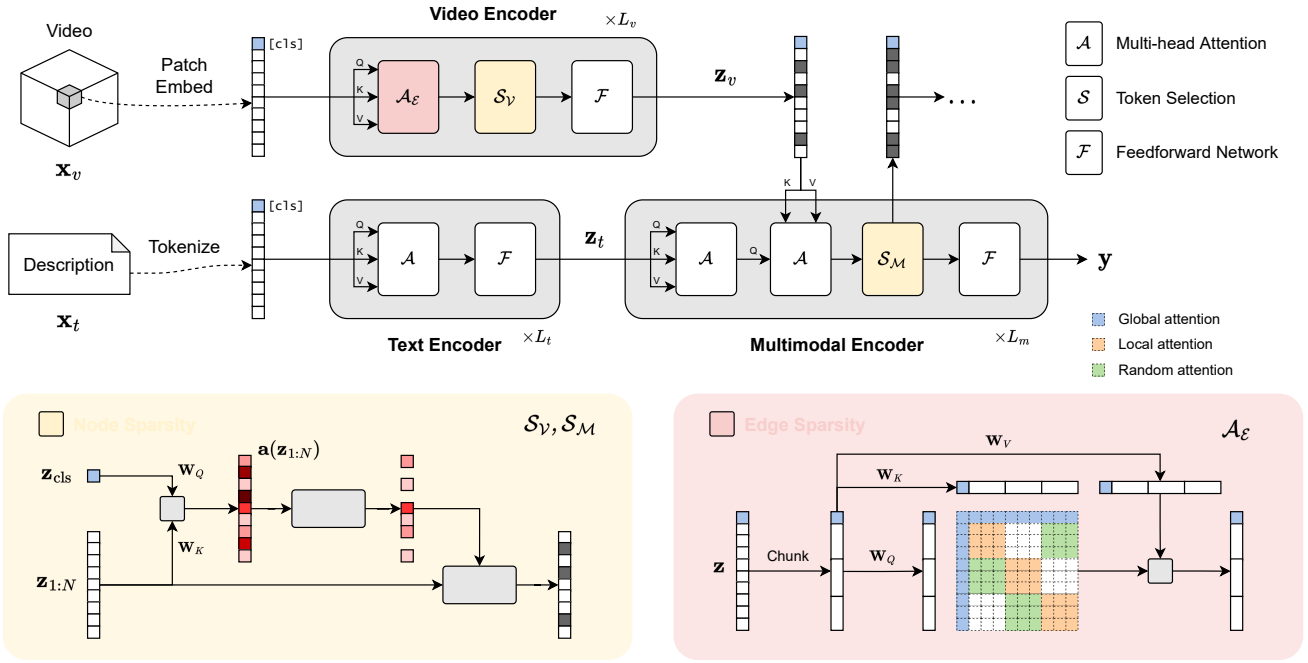


Figure 2. **Model Architecture.** **SViTT** improves modeling efficiency of conventional video-text transformers through two key components: *node* sparsity and *edge* sparsity. **Edge sparsification** $\mathcal{A}_\mathcal{E}$ computes sparse self-attention of input visual sequence $\mathbf{z}$, where each query token attends to a small subset of key and value tokens, with connectivity $\mathcal{E}$ specified by global, local, and random attention. **Node sparsification** $\mathcal{S}_\mathcal{V}$ uses global attention scores from $\mathcal{A}_\mathcal{E}$ to prune uninformative tokens, removing them from the computational graph of subsequent layers; $\mathcal{S}_\mathcal{M}$ uses text-to-video attention in the multimodal encoder to further reduce the length of visual sequence.

3.1. Video Transformers are Graph Models

Visual transformers [16] are deep neural networks that model images or videos as sequences of local pixel patches, through a combination of patchwise feature transformation and self-attention. Inspired by transformer architectures for language models [48], video transformers encode input clips into a sequence of spatiotemporal patches, flattened and linearly projected to a d-dimensional embedding space:

$$\mathbf{Z}^{(0)} = \left(\mathbf{z}_{\text{cls}}^{(0)}, \mathbf{z}_1^{(0)}, \ldots, \mathbf{z}_N^{(0)}\right) = f^{\text{tok}}(\mathbf{x}_{1:T}) \in \mathbb{R}^{(N+1)\times d} \tag{1}$$

where $N = T'H'W'$ is the volume of the 3D patch grid, and $\mathbf{z}_{\text{cls}}^{(0)}$ denotes a special class token responsible for instance-level prediction. The tokenized sequence is processed by a cascade of transformer blocks $f^{(l)}$

$$\mathbf{Z}^{(l)} = f^{(l)}(\mathbf{Z}^{(l-1)}), \quad l = 1, \ldots, L \tag{2}$$

each of which computes the self-attention $\mathcal{A}$ between input tokens, followed by a feed-forward network $\mathcal{F}$:[1]

$$f^{(l)}(\mathbf{Z}) = \mathcal{F}(\mathcal{A}(\mathbf{Z}\mathbf{W}_K^T, \mathbf{Z}\mathbf{W}_Q^T, \mathbf{Z}\mathbf{W}_V^T)), \tag{3}$$

$$\mathcal{A}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \sigma(\mathbf{Q}\mathbf{K}^T)\mathbf{V} \tag{4}$$

[1]Attention heads, residual connections and normalization terms are omitted for brevity, although we use the conventional implementation [48].

We interpret the transformer architecture as a special case of *graph* networks [6], with *nodes* representing tokenized video patches and *edges* connecting pairs of tokens for which self-attention is computed. Specifically, consider a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ defined by the vertices $\mathcal{V} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ corresponding to spatiotemporal video patches, and edges $\mathcal{E} \subseteq \{1, \ldots, N\}^2$ connecting pairs of nodes. The self-attention of (4) can be generalized such that node $\mathbf{z}_i$ attends to $\mathbf{z}_j$ only if $(i, j) \in \mathcal{E}$,

$$\mathcal{A}_\mathcal{E}(\mathbf{q}_i, \mathbf{K}, \mathbf{V}) = \sum_{j:(i,j)\in\mathcal{E}} \mathbf{a}_{ij}\mathbf{v}_j, \tag{5}$$

$$\mathbf{a}_{ij} = \frac{e^{\langle \mathbf{q}_i, \mathbf{k}_j \rangle}}{\sum_{j:(i,j)\in\mathcal{E}} e^{\langle \mathbf{q}_i, \mathbf{k}_j \rangle}} \tag{6}$$

Under this interpretation, the transformer architecture with full attention resembles a *complete* graph, where $\mathcal{E}$ includes every pair of vertices. This dense attention mechanism endows (5) with quadratic memory and time complexity w.r.t. sequence length N, making naïve transformers notoriously inefficient to train and expensive to deploy, especially for longer video clips. We argue that due to the inherent sparsity of information in video data, a large portion of the graph can be pruned dynamically without significant performance loss, leading to a *sparse* graph model of significantly lower cost for training and inference.

3.2. Edge Sparsity: Local & Random Attention

Prior natural language processing models, such as BigBird [55], have explored the idea of restricting the number of key-value pairs each query token attends to, which reduced the number of edges in $\mathcal{E}$. We use a similar procedure to create a video transformer with *edge sparsity*, utilizing a combination of *local*, *random* and *global* attention.

Local attention. Regional tokens $\{\mathbf{z}_i\}_{i=1}^N$ are first chunked into $N_b = \lceil N/G \rceil$ contiguous blocks of size G[2]. Tokens of one block k attend to a local neighborhood of K_l blocks $\{k-\Delta, \dots, k+\Delta\}$, where $\Delta = (K_l - 1)/2$ is the maximum range of local attention,

$$(k, k') \in \mathcal{E}, \quad \forall k, k' : |k' - k| \leq \Delta \tag{7}$$

This preserves the modeling of interactions between local features (objects, people, textures) that does not require long-range attention. In the case of $K_l = 1$, local reduces to diagonal attention, where each block only attends to itself.

Random attention. Beyond *local* attention, each block also attends to K_r other blocks sampled randomly from the input sequence,

$$(k, k') \in \mathcal{E}, \quad \forall k' \in \mathcal{N}(k) \tag{8}$$

where $\mathcal{N}(k)$ is a random subset of $\{k' \mid |k' - k| > \Delta\}$ of size K_r. This enables the transformer to model long-range visual relationships while avoiding the quadratic cost.

Global attention. Class token $\mathbf{z}_{\text{cls}}$ always attends to/from *regional* tokens $\mathbf{z}_i$, i.e. the link between $\mathbf{q}_{\text{cls}}$ and $\mathbf{k}_i$, as well as $\mathbf{q}_i$ and $\mathbf{k}_{\text{cls}}$, are always retained:

$$(\text{cls}, i), (i, \text{cls}) \in \mathcal{E}, \quad i = 1, \dots, N \tag{9}$$

This enables $\mathbf{z}_{\text{cls}}$ to capture global video context, even when the rest of tokens do not attend globally.

In summary, we retain all attention connections for local-to-global and global-to-local vertices, and limit local-to-local edges to $(K_l + K_r)G$ per token. Edge sparsity has a *linear* asymptotic cost of $O((K_l + K_r)GN)$, a significant improvement over dense attention at $O(N^2)$. However, this approach has two critical limitations of this strategy. First, the sparse patterns are predetermined and do not adapt dynamically to the input sequence. Second, the connections that remain in $\mathcal{E}$ are not determined by the video semantics. In result, connections between pairs of tokens of low semantic affinity (low attention values) may be preserved, impairing the efficiency of the sparse attention mechanism. To enable more aggressive sparsification, we introduce a second mechanism, which is dynamic, guided by video semantics, and applied to the *nodes* of the graph.

[2]Padding is added when N is not divisible by G.

3.3. Node Sparsity: Dynamic Token Pruning

A large percentage of the tokens of a video transformer corresponds to *contextual* regions, which contain little temporal dynamics and are only weakly related to the prediction target (e.g. background content uninformative of the activities of subjects in the foreground). While edge sparsification improves the efficiency of self-attention, it lacks both the flexibility and the semantic sensitivity to account for the uneven distribution of information across video patches.

To introduce these properties, we propose a node sparsification strategy, based on the dynamic pruning of tokens. We leverage a combination of observations. First, video semantics are summarized by the class tokens $\mathbf{z}_{\text{cls}}$, which contain a global representation of the information of interest for classification. Second the global-to-local edges survive the edge sparsification, through (9), making the attention weights from the $\mathbf{z}_{\text{cls}}$ to all regional tokens $\mathbf{z}_i$,

$$\mathbf{a}(\mathbf{z}_i) = \frac{e^{\langle \mathbf{q}_{\text{cls}}, \mathbf{k}_i \rangle}}{\sum_{j=1}^N e^{\langle \mathbf{q}_{\text{cls}}, \mathbf{k}_j \rangle}}, \quad i = 1, \dots, N \tag{10}$$

available for node sparsification. Since the class token is used for video-level predictions, $\mathbf{a}(\mathbf{z}_i)$ quantifies the contribution of feature $\mathbf{z}_i$ to the main task. This implies that nodes $\mathbf{z}_i$ of low $\mathbf{a}(\mathbf{z}_i)$ are not informative and can be ignored [30].

Node pruning then reduces to keeping the $N' = \lceil qN \rceil$ tokens of largest class attention

$$\mathcal{S}_{\mathcal{V}}(\mathbf{Z}; q) = \{\mathbf{z}_i \mid i \in \text{topk}(\mathbf{a}(\mathbf{Z}), \lceil qN \rceil)\} \tag{11}$$

where hyperparameter q denotes *keep rate* and $\text{topk}(\mathbf{L}, m)$ selects the m largest entries of vector $\mathbf{L}$. This procedure is repeated multiple times throughout the video encoder (keep rate $q^{(l)}$ at layer l), progressively reducing the length of input sequences. Importantly, the pruning procedure is dynamic and ensures that semantically uninformative vertices in the attention graph $\mathcal{G}$ are removed along with all edges they are associated with.

Cross-modal sparsity. Node sparsification can be naturally extended to video-language learning. For this, we propose to extend the token selection mechanism discussed above to a *cross-modal* setting, where video and text tokens $\mathbf{Z}_v, \mathbf{Z}_t$ are modeled jointly in a multimodal encoder. In this case, we replace the query $\mathbf{q}_{\text{cls}}$ of (10) with the class token of text sequence $\mathbf{q}_{\text{cls}}^{(t)}$, obtaining cross-modal attention

$$\mathbf{a}_m(\mathbf{z}_i^{(v)}) = \frac{e^{\langle \mathbf{q}_{\text{cls}}^{(t)}, \mathbf{k}_i^{(v)} \rangle}}{\sum_{j=1}^N e^{\langle \mathbf{q}_{\text{cls}}^{(t)}, \mathbf{k}_j^{(v)} \rangle}}, \quad i = 1, \dots, N \tag{12}$$

and subsequently the token sparsification function

$$\mathcal{S}_{\mathcal{M}}(\mathbf{Z}_v; q) = \{\mathbf{z}_i^{(v)} \mid i \in \text{topk}(\mathbf{a}_m(\mathbf{Z}_v), \lceil qN \rceil)\} \tag{13}$$

Multimodal node sparsity $\mathcal{S}_{\mathcal{M}}$ is applied on top of visual sparsity $\mathcal{S}_{\mathcal{V}}$. We expect cross-modal sparsification to create

additional room for sparsity over standalone visual modeling. While the visual encoder can identify salient actors and objects from background patches from the input clip, only the text semantics provide direct guidance for the video-text model to focus on regions relevant to the task of interest.

With node sparsity, the compute cost of subsequent layers is improved to $O(q^2N^2)$ using dense attention or $O(q(K_l + K_r)GN)$ with edge sparsity, and the reduction accumulates with multiple sparse layers.

3.4. Hybrid Sparse Transformers

We propose to combine *edge* and *node* sparsity into a unified framework, **SViTT**, as illustrated in Fig. 2. **SViTT** is built on top of existing video-language transformer architectures that combine separate video and text encoders with a cross-modal transformer. The visual encoder blocks perform sparse self-attention in the following steps:

- *edge sparsification*: given the random graph $\mathcal{E}$ of (7)-(9) compute attention weights $\mathcal{A}_\mathcal{E}$, using (5);
- *node sparsification:* select the nodes $\mathcal{S}_\mathcal{V}$ to retain, using (10)-(11) with keep rate $q < 1$).

After video and text embeddings $\mathbf{z}_v, \mathbf{z}_t$ are derived from their respective encoders, a multimodal transformer is applied to aggregate features across modalities. It follows the design of the text encoder, except for a cross-attention module that is applied after each self-attention block, where text queries $\mathbf{q}_t$ attends to key-value pairs $\mathbf{k}_v, \mathbf{v}_v$ extracted by the video encoder. The text-to-video attention of (12)-(13) is then used to select the nodes $\mathcal{S}_\mathcal{M}$ to retain, further reducing the number of video tokens for subsequent layers.

4. Temporal Sparse Expansion

In this section, we introduce a new training strategy for **SViTT**. We motivate for progressive model training with increasing clip length and sparsity in Sec. 4.1, and detail our pretraining procedure in Sec. 4.2.

4.1. Sparsity vs. Clip Length

The key insight behind the design of **SViTT** is the *diminishing return* of clip length. In general, a $2\times$ longer sequence does not contain twice the semantic information about the video, due to the redundancy of adjacent frames. This leads to a lower percentage of informative patches with denser frame sampling.

Due to this redundancy, it is possible to use higher sparsity for models with longer clips. This can be implemented by reducing the keep rate q of node sparsification, and the percentage of key/value blocks to attend to in edge sparsification $(K_l/N_b, K_r/N_b)$. For the latter, since the total number of blocks $N_b = \lceil N/G \rceil$ increases with number of frames T (assuming a fixed block size G), it suffices to keep the parameters K_l, K_r constant.

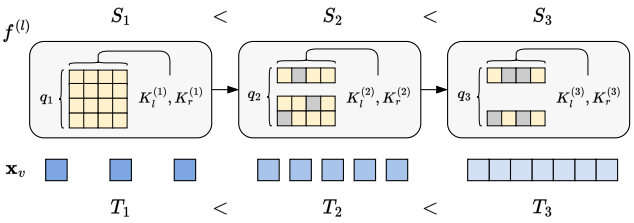


Figure 3. **Temporal Sparse Expansion.** We propose a multi-stage curriculum for training **SViTT**. At each stage, the node and edge sparsity S of video-text transformer increases with clip length T.

4.2. Temporal Expansion

Pretraining video-text transformers on long clips is time-consuming and leads to suboptimal models. Instead, we follow a learning strategy similar to Frozen [4], where the model is initially pretrained with shorter clips, and the number of frames increases as training progresses. However, when expanding the clip length, we increase the model sparsity to simultaneously 1) account for the redundancy of information, and 2) limit the growth of computational cost.

Fig. 3 depicts the expansion process proposed for video-text training. In the initial training stage $j = 1$, a dense video-text model is pretrained on clip length T_1. Denoting the sparsity hyperparameters of **SViTT** at stage j by $S_j = (q_j, K_l^{(j)}, K_r^{(j)})$, we create a learning curriculum with progressively increasing clip length and sparsity, by enforcing the constraints

$$T_1 < T_2 < \ldots; \tag{14}$$

$$q_1 > q_2 > \ldots; \tag{15}$$

$$\frac{K_l^{(1)} + K_r^{(1)}}{T_1} > \frac{K_l^{(2)} + K_r^{(2)}}{T_2} > \ldots \tag{16}$$

In practice, we use a decreasing token keep rate (15) and keep local and random attention block numbers fixed, i.e. $K_l^{(j)} = K_l$ and $K_r^{(j)} = K_r$, to satisfy (16).

5. Results

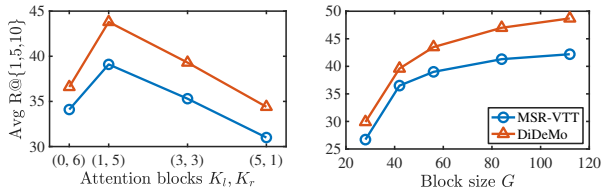
In this section we present experimental results of **SViTT** on vision-language modeling. We briefly introduce the experimental setup in Sec. 5.1, and perform several ablation studies on the design choices involving model sparsification and training in Sec. 5.2. We then demonstrate the performance of **SViTT** on various vision-language tasks in Sec. 5.3 and include additional qualitative analysis.

5.1. Experimental Setup

Architecture. Our implementation of video-text transformer is based on *Singularity* [24]. The model has a two-tower structure, with separate encoders for vision and language. The video encoder f_v is a 12-layer BEiT-B [5] initialized with ImageNet weights and inflated for video inputs. This differs from [24] which embeds each frame independently and applies late temporal fusion on extracted

Attn. blocks K_l, K_r, G	Keep rate q_v, q_m	# Edges (M)	DiDeMo R1	R5	R10	Mean
—	—	7.47	28.8	53.1	63.0	**48.3**
Edge sparsity						
(1, 3, 56)	—	2.14	20.7	41.7	50.5	**37.6**
(1, 5, 56)	—	3.21	26.0	48.6	56.8	**43.8**
Node sparsity						
—	(0.7, 1)	3.99	26.9	51.9	61.3	**46.7**
—	(0.7, 0.1)	3.97	27.6	53.1	62.9	**47.9**
Hybrid sparsity						
(1, 3, 56)	(0.7, 0.1)	1.48	19.9	40.5	50.6	**37.0**
(1, 5, 56)	(0.7, 0.1)	2.22	24.5	47.6	58.6	**43.6**

Table 1. **Ablation on Edge and Node Sparsity**. We evaluate the same *dense* video-text model under different sparsification modes.



(a) Number of local / random blocks. (b) Attention block size.

Figure 4. **Ablation on Edge Sparsity.** We evaluate dense model using different local/random blocks (K_l, K_r) and block size G.

features. The text encoder f_t is a pretrained BERT [14] model, whose last 3 layers are modified to implement the multimodal encoder f_m, with *cross-modal* attention based on visual tokens as key-value pairs, which we sparsify as described in Sec. 3.3.

Pre-training. **SViTT** is pre-trained on 2.5M video-text pairs from the WebVid dataset [4]. Since our goal is to investigate how to improve the effectiveness of the *temporal* learning of the video modality, we do not train with additional image-text corpora as done in [4, 18, 24].

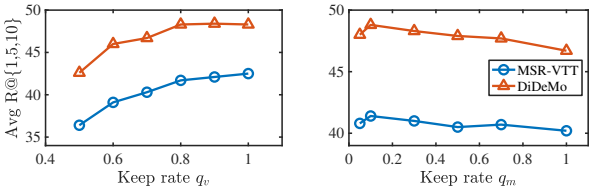
The `[cls]` tokens of the video and text encoders are first linearly projected to a joint embedding space, producing feature vectors $\mathbf{z}_v = f_v(\mathbf{x}_v)$ and $\mathbf{z}_t = f_t(\mathbf{x}_t)$, respectively. Following prior work, we use the InfoNCE loss [40] to align these feature vectors. The output of multimodal encoder $\mathbf{y} = f_m(\mathbf{z}_v, \mathbf{z}_t)$ is optimized with video-text matching (VTM) and masked language modeling (MLM) losses commonly found in VLP literature [12, 18, 24, 27, 28].

Downstream tasks. Trained video-text models are evaluated on two multimodal tasks: *Text-to-video retrieval* and *video question answering*. Video retrieval is evaluated on MSR-VTT [51], DiDeMo [2], Charades [43] and Something-Something v2 [20, 24], by top-K recalls ($K \in \{1, 5, 10\}$) and their numeric average. Question answering is evaluated on MSRVTT-QA [49], ActivityNet-QA [10,54] and AGQA [21], with top-1 accuracy of the answers.

Training details are given in the Appendix.

T	Spars.	FLOPs (G)	Mem. (GB)	DiDeMo R1	R5	R10	Mean
4	Dense	139.9	0.96	28.8	53.1	63.0	48.3
	Edge	135.9	0.80	28.3	51.9	61.4	47.2
	Node	95.8	0.61	**29.9**	**54.8**	**63.9**	**49.6**
	Hybrid	**93.9**	**0.54**	29.3	53.7	63.2	48.8
8	Dense	291.3	3.14	29.6	54.1	64.1	49.3
	Edge	271.8	1.57	29.8	54.9	65.4	50.1
	Node	197.6	1.77	30.4	55.7	66.0	50.7
	Hybrid	**166.4**	**0.92**	**31.0**	**57.2**	**66.3**	**51.5**
16	Dense	627.8	10.66	Untrainable			
	Edge	543.6	3.02	**31.6**	55.1	64.6	50.5
	Node	370.9	4.39	Untrainable			
	Hybrid	**296.2**	**1.57**	31.4	**57.3**	**67.8**	**52.2**

Table 2. **Ablation on Training Sparse Models**. We compare the zero-shot performance, inference GFLOPs, and training memory (per sample) of sparse models to the dense attention baseline.



(a) Visual sparsification. (b) Multimodal sparsification.

Figure 5. **Ablation on Node Sparsity.** We evaluate the pre-trained dense model using different keep rates q at test time.

5.2. Ablation Studies

We start by training a video-text transformer with clip length of 4 frames and *dense* attention, and measure its zero-shot performance on downstream tasks after *sparsifying* its video encoder while keeping its weights unchanged.

Edge sparsification. We first apply edge sparsification with different number of local blocks K_l, random blocks K_r, and block size G. As shown in Fig. 4a, under a limited budget of 6 local or random attention blocks per query token, using 1 local block and 5 random blocks provides the best trade-off. Increasing the local attention window K_l hurts long-term modeling capacity and degrades retrieval; while removing the single local block responsible for diagonal attention also impairs performance. This suggests that while query tokens should always attend to their respective blocks, there is no benefit in attending to neighboring blocks. We thus fix $K_l = 1$ for the rest of experiments.

We next vary the block size G of sparse attention. Fig. 4b shows that larger sizes have stronger retrieval performance, as expected. However, they also make self-attention less sparse (more costly to compute). $G = 56$ provides a good balance between performance and complexity. Tab. 1 summarizes the test performance of two edge sparsity configs: $(K_l, K_r, G) = (1, 3, 56)$ and $(1, 5, 56)$. While both underperform the dense model, we will later demonstrate that the

Frames T	Sparsity T_S	DiDeMo R1	R5	R10	Mean
4	$4_{.80}$	28.0	50.7	59.2	46.0
	$1_0 \rightarrow 4_{.80}$	**29.3**	**53.7**	**63.2**	**48.8**
8	$8_{.91}$	27.3	51.4	63.8	47.5
	$4_{.80} \rightarrow 8_{.91}$	**31.0**	**57.2**	**66.3**	**51.5**
16	$16_{.96}$	27.5	52.4	63.2	47.7
	$4_{.80} \rightarrow 8_{.91} \rightarrow 16_{.96}$	**31.4**	**57.3**	**67.8**	**52.2**

Table 3. **Ablation on Temporal Sparse Expansion**. Sparsity T_S indicates training on T frames while removing $S \times 100\%$ attention edges of dense transformer.

gap can be closed and even reversed by training the sparse model with the proposed temporal expansion curriculum.

Node sparsification. We next incorporate node sparsity into the video-text transformer. This includes *visual* sparsification (keep rate q_v) using the self-attention of video encoder f_v to progressively prune the input tokens[3], and *multimodal* sparsification (keep rate q_m) using the text-to-video attention of cross-modal encoder f_m to further drop visual tokens unrelated to the text query. Fig. 5a shows that the dense model is quite robust to token pruning, even without sparse training. Using visual keep rate $q_v \geq 0.8$ has minimal impact on test results, and performance only starts to drop rapidly at $q_v = 0.5$, at which point only $1/8$ of input tokens are retained after three rounds of sparsification.

Even more surprisingly, the subsequent multimodal sparsification step, using a keep rate of $q_m = 0.1$, *improves* zero-shot performance by 1%. This shows that the 90% redundant visual tokens not only add unnecessary complexity to the model, but also introduce noise that harms retrieval performance. The fact that the optimal q_m is much lower than q_v also suggests that text modality provides crucial semantic guidance for identifying relevant visual patches, at a much higher accuracy than visual modeling alone.

Hybrid sparsification. Combining the best sparse settings for edges and nodes, we obtain the hybrid sparsification strategy for **SViTT**. As shown in Tab. 1, compared to edge sparsity, the introduction of node sparsity ($q_v = 0.7, q_m = 0.1$) only impacts recall scores marginally, while saving computations on a large portion of visual tokens throughout the network.

Training sparse transformers. We next perform *full pretraining* with the sparse models and compare their performance and efficiency to the dense transformer baseline. Tab. 2 summarizes the results obtained with different input clip lengths and types of sparsity. At 4 frames, edge sparsity has small benefit due to the relatively short sequence length and node sparsity performs the best. However, at 8 frames, we start to see clear advantages to edge sparsity in memory

[3]We follow [30] to prune visual tokens at layer #4, #7, and #10.

Method	PT	T	MSR-VTT R1	MSR-VTT Mean	DiDeMo R1	DiDeMo Mean
VideoCLIP [50]	100M	—	10.4	20.9	16.6	—
Frozen [4]	5M	4	23.2	41.5	21.1	41.1
ALPRO [26]	5M	8	24.1	41.4	23.8	43.0
VIOLET [18]	5M	4	25.9	45.0	23.5	44.4
Singularity [24]	5M	1	*28.4*	*46.0*	*36.9*	*55.8*
Singularity*	2M	1	21.1	38.7	23.3	40.8
		4	24.4	40.0	26.4	44.1
		8	24.3	41.0	25.8	45.5
SViTT Dense	2M	8	**26.0**	43.6	29.6	49.3
SViTT Hybrid	2M	8	25.4	**43.8**	**31.0**	**51.5**

Table 4. **Zero-shot Text-to-video Retrieval.** Results reported in prior works are marked in gray; * indicates our reproduced results. **PT** = # video-text pairs for pre-training, T = # frames per clip.

complexity, and both edge and node sparsity outperform the dense transformer baseline. Combining both types of sparsity performs the best, while only requiring 60% the FLOPs and 30% the memory of the dense model. At 16 frames, the dense and node sparsity models are no longer trainable due to their quadratically increasing cost. The models with edge sparsity, however, are able to fit into GPU memory thanks to the linear complexity from sparse attention. A 16-frame **SViTT** with hybrid sparsity requires similar computation to an 8-frame dense model and only half of its training memory, while achieving 3% higher recall.

Progressive training. We next study the impact of temporal sparse expansion on the training of **SViTT** models on longer clips. Tab. 3 compares the performance of models trained using temporal expansion (i.e. initialized from checkpoints pretrained on fewer frames and lower sparsity) to standard single-stage training. For single-stage training, performance does not improve substantially with clip length. This is in contrast to the proposed sparse expansion, where using 8 instead or 4 frames results in a gain of 2.7%. This suggests that the models have learned to exploit the temporal relationships between video frames. For a given clip length, sparse expansion also substantially improves upon single stage performance, from 2.8 points for $T = 4$ to 4.5 points for $T = 16$. In fact, sparse expansion training with a shorter clip length (e.g. 4 frames) can outperform single stage training with a larger length (16 frames).

5.3. Main Results

We compare **SViTT** to state-of-the-art models in text-to-video retrieval and video question answering.

Text-to-video retrieval. Video-text retrieval is evaluated under zero-shot and fine-tuning settings. Tab. 4 shows the zero-shot results on MSR-VTT and DiDeMo. Compared to our reproduced models of Singularity on WebVid-2M, which aggregate frame-level features using a temporal transformer encoder, the spatiotemporal transformer

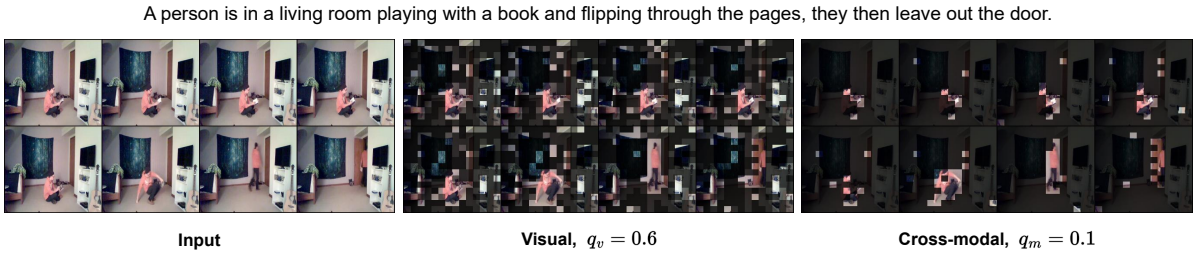


Figure 6. **Qualitative Results.** **SViTT** isolates informative regions from background patches to facilitate efficient temporal reasoning.

Method		PT	T	Charades		SSv2	
				R1	Mean	R1	Mean
Frozen [4]		5M	32	11.9	25.1	—	—
CLIP4Clip [35]		400M	12	13.9	27.1	43.1	65.1
ECLIPSE [31]		400M	32	15.7	30.3	—	—
MKTVR† [36]		400M	42	16.6	34.7	—	—
Singularity [24]		5M	1	—	—	36.4	58.9
			4	—	—	44.1	66.6
SViTT	Dense	2M	8	16.0	32.7	43.6	66.1
	Hybrid			**17.7**	**35.7**	**49.8**	**69.3**

Table 5. **Text-to-video Retrieval with Fine-tuning.**

Method		PT	T	MSRVTT	ANet	AGQA
HME [17]		—	20	33.0	—	47.7
HCRN [23]			128	35.5	—	47.4
ClipBERT [25]		0.2M	16	37.4	—	—
ALPRO [26]		5M	16	42.1	—	—
Just Ask [52]		69M	640	41.5	38.9	—
MERLOT [56]		180M	5	43.1	41.4	—
VIOLET [18]		185M	4	**43.9**	—	49.2
Singularity [24]		5M	1	42.7	41.8	—
SViTT	Dense	2M	8	42.7	42.3	50.2
	Hybrid			43.0	**43.2**	**52.7**

Table 6. **Video Question Answering.**

of **SViTT** produces stronger results on both downstream datasets. This highlights the importance of temporal modeling in earlier layers of video-text transformers. **SViTT** with hybrid sparsity has similar performance to the dense model on MSR-VTT but significantly outperforms in on DiDeMo, which contains longer videos with localized activities.

For Charades and SSv2, we evaluate text-to-video retrieval with fine-tuning, as shown in Tab. 5. Both datasets are action-centric, posing a greater challenge to the temporal reasoning of video-text models. **SViTT** with hybrid sparsification dominates the dense variant by $\sim 3\%$ on both datasets, a more substantial gap than observed for MSR-VTT and DiDeMo. This confirms our hypothesis that exploiting node and edge sparsity reduces the dependency of models on contextual regions, forcing them to focus on the temporal dynamics of person and objects in the foreground.

Video question answering. We next evaluate the cross-modal modeling of **SViTT** on MSRVTT-QA, ActivityNet-QA and AGQA. As shown in Tab. 6, the hybrid sparsity version of **SViTT** outperforms the dense transformer baseline on all three datasets, thanks to its more efficient temporal modeling. On MSRVTT-QA, the accuracy gap is small between dense and sparse transformer (0.3%), and **SViTT** marginally underperforms baselines pretrained with fewer number of frames (MERLOT [56], VIOLET [18]). This is likely due to the nature of MSRVTT, which consists of short video clips and questions biased towards spatial cues, allowing temporal modeling little benefit over spatial transformers pretrained on massive image & video data. On ActivityNet-QA and AGQA, which both contain longer clips and temporally challenging questions, the sparse modeling of **SViTT** proves advantageous, with 0.9% and 2.3% boost in accuracy respectively, beating all baseline methods.

Qualitative analysis. To show how **SViTT** efficiently identifies and concentrates its computation on informative spatiotemporal regions of the input clips, we visualize the outcome of node sparsification in Fig. 6. Using visual sparsification in video encoder f_v, **SViTT** learns to isolate foreground entities from the majority of background patches, enabling the model to perform sparse video-text inference on longer temporal context. Cross-modal attention in multimodal encoder f_m provides an even stronger signal for isolating the regions of interest of each video clip, validating the importance of text semantics in visual sparsification.

6. Conclusion

This work introduced **SViTT**, a sparse video-text transformer for efficient reasoning over a long temporal context. By interpreting visual transformers as graph networks, we proposed to optimize their *edge* and *node* sparsity, using a combination of sparse block attention, visual token pruning and text-guided token selection. We further introduced a temporal expansion strategy for training **SViTT**, which aims to gradually increase model sparsity with clip length. **SViTT** showed strong performance and efficiency compared to dense transformers, with a larger gap when frame number increases. On video retrieval and question answering benchmarks, **SViTT** achieved state-of-the-art results using only video data, without extra image-text pretraining.

Acknowledgements. This work was funded in part by NSF award IIS-2041009.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022. 1

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

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

[4] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1728–1738, 2021. 1, 2, 5, 6, 7, 8

[5] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEit: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022. 1, 5

[6] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018. 3

[7] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. 2

[8] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021. 2

[9] 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, 2

[10] 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. 6

[11] Tianlong Chen, Yu Cheng, Zhe Gan, Lu Yuan, Lei Zhang, and Zhangyang Wang. Chasing sparsity in vision transformers: An end-to-end exploration. *Advances in Neural Information Processing Systems*, 34:19974–19988, 2021. 2

[12] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer, 2020. 1, 6

[13] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. 2

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 6

[15] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12124–12134, 2022. 2

[16] 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. In *International Conference on Learning Representations*, 2021. 1, 2, 3

[17] Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. Heterogeneous memory enhanced multimodal attention model for video question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1999–2007, 2019. 8

[18] 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, 6, 7, 8

[19] 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. 2

[20] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017. 6

[21] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. Agqa: A benchmark for compositional spatio-temporal reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11287–11297, 2021. 6

[22] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. 1

[23] Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. Hierarchical conditional relation networks for video question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9972–9981, 2020. 8

[24] Jie Lei, Tamara L Berg, and Mohit Bansal. Revealing single frame bias for video-and-language learning. *arXiv preprint arXiv:2206.03428*, 2022. 1, 2, 5, 6, 7, 8

[25] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for

video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7331–7341, 2021. 8

[26] 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. 7, 8

[27] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021. 1, 6

[28] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2046–2065, 2020. 6

[29] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 1

[30] 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. In *International Conference on Learning Representations*, 2022. 2, 4, 7

[31] Yan-Bo Lin, Jie Lei, Mohit Bansal, and Gedas Bertasius. Eclipse: Efficient long-range video retrieval using sight and sound. *arXiv preprint arXiv:2204.02874*, 2022. 8

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 1, 2

[33] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211, 2022. 2

[34] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019. 1

[35] 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 and captioning. *Neurocomputing*, 508:293–304, 2022. 8

[36] Avinash Madasu, Estelle Aflalo, Gabriel Ben Melech Stan, Shao-Yen Tseng, Gedas Bertasius, and Vasudev Lal. Improving video retrieval using multilingual knowledge transfer. *arXiv preprint arXiv:2208.11553*, 2022. 8

[37] Lingchen Meng, Hengduo Li, Bor-Chun Chen, Shiyi Lan, Zuxuan Wu, Yu-Gang Jiang, and Ser-Nam Lim. Adavit: Adaptive vision transformers for efficient image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12309–12318, 2022. 2

[38] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889, 2020. 2

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

[40] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 6

[41] 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. 2

[42] Michael Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: Adaptive space-time tokenization for videos. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 12786–12797. Curran Associates, Inc., 2021. 2

[43] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526. Springer, 2016. 6

[44] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15638–15650, 2022. 1

[45] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473, 2019. 2

[46] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 1

[47] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12894–12904, 2021. 2

[48] 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. 2, 3

[49] 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. 6

[50] 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. 7

[51] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016. 6

[52] 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. 8

[53] Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10809–10818, 2022. 2

[54] 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. 6

[55] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297, 2020. 2, 4

[56] 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. 2, 8

[57] 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