Supplemental Material

A. Implementation Details

A.1. Model Architecture

**Sparse configurations.** The sparsity of SViT is controlled by the following hyperparameters:

- Visual token keep rate $q_v^{(l)}$ and multimodal token keep rate $q_m^{(l)}$ per layer $l$ for node sparsity;
- Local attention blocks $K_l$, random attention blocks $K_r$ and block size $G$ shared across layers for edge sparsity.

Tab. A1 lists the configurations for each stage of pre-training and the corresponding sparsity $s$, computed as the percent of reduction in edges of sparsified attention graph $G$ from that of a dense transformer. For the $l$th layer of visual encoder $f_v$, the number of edges is given by

$$|E_v^{(l)}| = N_v^{(l)}(K_l + K_r)G$$

where input length $N_v^{(l)} = q_v^{(l-1)}N_v^{(l-1)}$. For multimodal layers $f_m$, the edge count is

$$|E_m^{(l)}| = N_m^{(l)}N_t$$

where $N_t$ denotes text length and $N_m^{(l)} = q_m^{(l-1)}N_m^{(l-1)}$. Therefore an SViT model with $L_v = 12$ visual layers and $L_m = 3$ multimodal layers has overall edge sparsity

$$S(q_v, q_m, K_l, K_r) = 1 - \frac{\sum_{l=1}^{L_v}|E_v^{(l)}| + \sum_{l=1}^{L_m}|E_m^{(l)}|}{L_vN_v^2 + L_mN_tN_v}$$

**Temporal expansion.** Transformer architectures do not require fixed input lengths as its operations are either point-wise (e.g. FFN) or permutation equivariant (e.g. MHSA). This makes the temporal expansion (Sect. 4) of input clips a mostly trivial process, except for the position embeddings, which does depend on spatiotemporal dimensions of inputs. Following prior work on training video transformers with image models, we inflate the 2D positional embedding $P = [p_{cls}, p_{1,1}, \ldots, p_{H,W}] \in \mathbb{R}^{(HW+1) \times d}$ into a 3D embedding tensor

$$P' = [p_{cls}, p_{1,1,1}, \ldots, p'_{T,H,W}] \in \mathbb{R}^{(THW+1) \times d}$$

for inputs of $T$ frames, by duplicating the local embeddings $p_{h,w}$ along the temporal dimension:

$$p'_{t,h,w} = p_{h,w}, \quad \forall t, h, w$$

Likewise, expansion of clip length from $T_1$ to $T_2$ can be performed by temporally resizing the positional embedding, e.g. through nearest neighbors interpolation:

$$p'_{t,h,w} = p_{\lfloor t \frac{T_1}{T_2} + \frac{1}{2} \rfloor, h,w}, \quad \forall t, h, w$$

The BEiT backbone of visual encoder uses relative position bias $[22]$ in every self-attention layer, which encodes a scalar added to each entry of the similarity matrix depending on the relative position between query and key patches:

$$A(Q, K, V) = \sigma(QK^T + B)V, \quad B_{(h,w),(h',w')} = R_{h'-h,w'-w}$$

<table>
<thead>
<tr>
<th>Frames</th>
<th>Attn. blocks</th>
<th>Keep rate</th>
<th>Edges (M)</th>
<th>Sparsity</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>(0.7, 0.1)</td>
<td>1.48</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(0.6, 0.1)</td>
<td>2.60</td>
<td>0.91</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>(0.5, 0.1)</td>
<td>4.61</td>
<td>0.96</td>
<td></td>
</tr>
</tbody>
</table>

Table A1. **SViT Configurations.** We report hyperparameters controlling the edge and node sparsity for different clip lengths $T$, as well as the overall sparsity as computed by Eq. (3).
where $\mathbf{R} \in \mathbb{R}^{(2H-1) \times (2W-1)}$ are learnable parameters. When expanding the input to multi-frame clips, we again inflate the relative position bias to the temporal dimension:

$$B'(t,h,w) = B'(t',h',w') = B'_t - t, B'_h - h, B'_w - w,$$

$$\mathbf{R'} \in \mathbb{R}^{(2T-1) \times (2H-1) \times (2W-1)}$$

$\mathbf{R'}$ is initialized by interpolating $\mathbf{R}$ temporally, identical to the procedure for absolute positional embedding $\mathbf{P'}$.

### A.2. Datasets

#### Pre-training

**SViTT** is pre-trained on WebVid-2M [2] with 2.5 million video-text pairs scraped from the Internet. While alternative datasets exist for video-language pre-training such as HowTo100M [20] and YT-Temporal [30], we choose WebVid as it has higher caption quality, covers a wide range of scenes, and can be trained with a reasonable amount of resource.

#### Text-to-video retrieval

We evaluate text-to-video retrieval on 4 datasets: MSR-VTT [27], DiDeMo [1], Charades [24] and Something-Something v2 [8]. MSR-VTT and DiDeMo are video-text datasets commonly used in prior work; Charades and SSv2 were initially collected for video action recognition, with an emphasis on human-object interactions and temporal modeling, but also includes text descriptions for each video clip.

#### Video question answering

Video question answering is evaluated on MSRVTT-QA [25], ActivityNet-QA [28] and AGQA 2.0 [10], annotated on top of the videos from MSR-VTT [27], ActivityNet [5] and Charades [24] respectively. MSRVTT-QA consists of mostly descriptive questions which can be solved without intricate temporal reasoning. ActivityNet-QA focuses on human actions and spatiotemporal relation between objects, posing a greater challenge beyond frame-based reasoning. AGQA contains difficult questions involving the composition of actions, testing the generalization capacity of video-text models.

Tab. A2 summarizes the statistics of all aforementioned datasets.

### A.3. Training Details

#### Pre-training tasks

**SViTT** is pre-trained on three losses following prior art in VLP [6, 7, 12, 14, 15].

- Video-text contrastive (VTC) applies InfoNCE loss between the video embeddings $\mathbf{Z}_v$ and text embeddings $\mathbf{Z}_t$ extracted at $[\text{cls}]$ locations of their respective encoder $f_v$ and $f_t$:

$$\mathcal{L}_{VTC} = \ell_c(\mathbf{Z}_v, \mathbf{Z}_t) + \ell_c(\mathbf{Z}_t, \mathbf{Z}_v),$$

$$\ell_c(\mathbf{X}, \mathbf{Y}) = - \sum_{i=1}^{B} \log \frac{e^{(\mathbf{X}_i, \mathbf{Y}_i)}/\tau} { \sum_{j=1}^{B} e^{(\mathbf{X}_i, \mathbf{Y}_j)}/\tau}$$

- Video-text matching (VTM) learns a binary classifier on top of the $[\text{cls}]$ output of multimodal encoder $f_m$ to discriminate between paired and misaligned video-text pair, optimized by binary cross entropy:

$$\mathcal{L}_{VTM} = - \sum_{i=1}^{B} \left( \log(f_m(\mathbf{Z}_c,i, \mathbf{Z}_t,i)) ight) + \log(1 - f_m(\mathbf{Z}_v,i, \mathbf{Z}_t,i)) \right)$$

where $i' \neq i$ is a randomly selected negative sample.

- Masked language modeling (MLM) requires the multimodal encoder $f_m$ to predict randomly masked out text tokens conditioned on the rest of text and video sequence, through a cross-entropy loss:

$$\mathcal{L}_{MLM} = - \sum_{i=1}^{B} \sum_{j \in J} \log y_{i,j}$$

where $[x_{i}]_{i,j}$ is a one-hot vector denoting the word at location $j$ of example $i$, $y_{i,j}$ is the classifier output predicting the word at the same location, and $J$ is the set of masked indices.

We use equal weights for all three losses.

#### Downstream tasks

We follow the downstream evaluation setup of Singularity [12] for the most part. Text-to-video retrieval is performed by ranking all candidate videos $x_v$ of the test set by their matching scores to text query $x_t$. For video QA, a transformer decoder is applied on top of multimodal encoder $f_m$ to generate the answer.

---

1 Linear projection on top of $z_v, z_t$ omitted.
## B. Additional Results & Analysis

### B.1. Retrieval Metrics

We include full retrieval results with Recall@{1, 5, 10} in Tab. A3 (zero-shot) and Tab. A5 (fine-tuned).

### B.2. Video-Text Backbone

In addition to the Singularity baseline with BEiT-B backbone used in the main paper, we also evaluate SViTT on a simpler structure from Frozen [2]. This is also a two-tower model with separate video and text encoders $f_v, f_t$, but unlike most vision-language transformers, does not contain a cross-modal encoder on top. Frozen is trained solely on the InfoNCE loss between video and text embeddings, and uses their cosine similarity to perform retrieval. While the cross-modal node sparsification does not apply to this framework, visual node sparsity and edge sparsity can still be applied to the visual encoder $f_v$ to enable temporal learning across frames.

The original Frozen model uses a divided space-time attention similar to TimeSformer [4], where temporal attention is added to a pre-trained ViT and initialized as identity mapping. During early experiments, however, we find that the temporal module with zero-init fails to learn meaningful attention across frames, with query and key matrices stuck at zero weights. We opted to remove the temporal attention modules and make the spatial attention global instead (i.e. each token attends to every token from the video clip, instead of just those from the same frame).

Tab. A6 shows the performance of SViTT applied to the Frozen model. Similar to the results in the main paper, our dense spatiotemporal transformer with the above modifica-

<table>
<thead>
<tr>
<th>Method</th>
<th>PT</th>
<th>Frames</th>
<th>Sparsity</th>
<th>MSR-VTT R1</th>
<th>R5</th>
<th>R10</th>
<th>Mean</th>
<th>DiDeMo R1</th>
<th>R5</th>
<th>R10</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>VideoCLIP [26]</td>
<td>100M</td>
<td>—</td>
<td></td>
<td>10.4</td>
<td>22.2</td>
<td>30.0</td>
<td><strong>20.9</strong></td>
<td>16.6</td>
<td>46.9</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Frozen [2]</td>
<td>5M</td>
<td>4</td>
<td></td>
<td>23.2</td>
<td>44.6</td>
<td>56.6</td>
<td><strong>41.5</strong></td>
<td>21.1</td>
<td>46.0</td>
<td>56.2</td>
<td><strong>41.1</strong></td>
</tr>
<tr>
<td>ALPRO [13]</td>
<td>5M</td>
<td>8</td>
<td>—</td>
<td>24.1</td>
<td>44.7</td>
<td>55.4</td>
<td><strong>41.4</strong></td>
<td>23.8</td>
<td>47.3</td>
<td>57.9</td>
<td><strong>43.0</strong></td>
</tr>
<tr>
<td>VIOLET [7]</td>
<td>5M</td>
<td>4</td>
<td></td>
<td>25.9</td>
<td>49.5</td>
<td>59.7</td>
<td><strong>45.0</strong></td>
<td>23.5</td>
<td>49.8</td>
<td>59.8</td>
<td><strong>44.4</strong></td>
</tr>
<tr>
<td>Singularity [12]</td>
<td>5M</td>
<td>1</td>
<td></td>
<td>28.4</td>
<td>50.2</td>
<td>59.5</td>
<td><strong>46.0</strong></td>
<td>36.9</td>
<td>61.1</td>
<td>69.3</td>
<td><strong>55.8</strong></td>
</tr>
<tr>
<td>Singularity*</td>
<td>2M</td>
<td>4</td>
<td>—</td>
<td>21.1</td>
<td>42.1</td>
<td>53.0</td>
<td><strong>38.7</strong></td>
<td>23.3</td>
<td>45.4</td>
<td>53.7</td>
<td><strong>40.8</strong></td>
</tr>
<tr>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td>24.4</td>
<td>43.8</td>
<td>51.7</td>
<td><strong>40.0</strong></td>
<td>26.4</td>
<td>48.7</td>
<td>57.3</td>
<td><strong>44.1</strong></td>
</tr>
<tr>
<td><strong>SViTT</strong></td>
<td>2M</td>
<td>8</td>
<td>Dense</td>
<td>26.0</td>
<td>47.7</td>
<td>57.1</td>
<td><strong>43.6</strong></td>
<td>29.6</td>
<td>54.1</td>
<td>64.1</td>
<td><strong>49.3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Hybrid</td>
<td>25.4</td>
<td>48.4</td>
<td>57.5</td>
<td><strong>43.8</strong></td>
<td>31.0</td>
<td>57.2</td>
<td>66.3</td>
<td><strong>51.5</strong></td>
</tr>
</tbody>
</table>

Table A3. Zero-shot Text-to-video Retrieval. Results reported in prior works marked in gray; * indicates our reproduced results.

### Training Hyperparameters

<table>
<thead>
<tr>
<th>Task</th>
<th>Frames $T$</th>
<th>Pre-training</th>
<th>Video-text Retrieval</th>
<th>Video QA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Epochs</td>
<td>4, 8, 16</td>
<td>10</td>
<td>15</td>
<td>5 (1 for AGQA)</td>
</tr>
<tr>
<td>Warm-up</td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Batch size</td>
<td>512, 336, 192</td>
<td>$3 \times 10^{-5}$, $1 \times 10^{-5}$, $5 \times 10^{-6}$</td>
<td>64, 48, 32</td>
<td>128, $5 \times 10^{-5}$</td>
</tr>
<tr>
<td>Learning rate</td>
<td></td>
<td>0.02</td>
<td>$1 \times 10^{-5}$</td>
<td>0.02</td>
</tr>
<tr>
<td>Weight decay</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text length</td>
<td>32</td>
<td></td>
<td>32 (64 for DiDeMo)</td>
<td>25 (Q), 5 (A)</td>
</tr>
<tr>
<td>Attn. blocks $(K_v, K_r, G)$</td>
<td>(1, 3, 56)</td>
<td>(0.7, 0.1) (0.6, 0.1) (0.5, 0.1)</td>
<td>(0.7, 0.1) (0.6, 0.1) (0.5, 0.1)</td>
<td>(0.7, 0.1) (0.6, 0.1) (0.5, 0.1)</td>
</tr>
<tr>
<td>Keep rate $(q_v, q_m)$</td>
<td>(0.7, 0.1) (0.6, 0.1) (0.5, 0.1)</td>
<td>(0.7, 0.1) (0.6, 0.1) (0.5, 0.1)</td>
<td>(0.7, 0.1) (0.6, 0.1) (0.5, 0.1)</td>
<td></td>
</tr>
</tbody>
</table>

Table A4. Training Hyperparameters.
Table A5. Text-to-video Retrieval with Fine-tuning. † denotes concurrent work.

<table>
<thead>
<tr>
<th>Method</th>
<th>PT</th>
<th>Frames</th>
<th>Sparsity</th>
<th>Charades R1</th>
<th>Charades R5</th>
<th>Charades R10</th>
<th>Ssv2-Label R1</th>
<th>Ssv2-Label R5</th>
<th>Ssv2-Label R10</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frozen [2]</td>
<td>5M</td>
<td>32</td>
<td>11.9</td>
<td>28.3</td>
<td>35.1</td>
<td>25.1</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>CLIP4Clip [18]</td>
<td>400M</td>
<td>12</td>
<td>13.9</td>
<td>30.4</td>
<td>37.1</td>
<td>27.1</td>
<td>43.1</td>
<td>71.4</td>
<td>80.7</td>
<td>65.1</td>
</tr>
<tr>
<td>ECLIPSE [16]</td>
<td>400M</td>
<td>32</td>
<td>15.7</td>
<td>32.9</td>
<td>42.4</td>
<td>30.3</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>MKTVR† [19]</td>
<td>400M</td>
<td>42</td>
<td>16.6</td>
<td>37.5</td>
<td>50.0</td>
<td>34.7</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>Singularity [12]</td>
<td>5M</td>
<td>1</td>
<td>—</td>
<td>—</td>
<td>36.4</td>
<td>43.1</td>
<td>71.4</td>
<td>80.7</td>
<td>65.1</td>
<td></td>
</tr>
<tr>
<td><strong>SViTT</strong></td>
<td>2M</td>
<td>8</td>
<td>Dense</td>
<td>16.0</td>
<td>34.9</td>
<td>47.2</td>
<td>32.7</td>
<td>43.6</td>
<td>72.6</td>
<td>66.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Hybrid</td>
<td>17.7</td>
<td>39.5</td>
<td>49.8</td>
<td>35.7</td>
<td>47.5</td>
<td>76.3</td>
<td>69.3</td>
</tr>
</tbody>
</table>

Table A6. Zero-shot Retrieval with SViTT on Frozen Baseline.

Table A7. Ablation on Token Ordering. We compare the standard SViTT trained with flattened video tokens and reordering using space-filling curves.

<table>
<thead>
<tr>
<th>Order</th>
<th>MSR-VTT R5</th>
<th>MSR-VTT R10</th>
<th>DiDeMo R5</th>
<th>DiDeMo R10</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>21.0</td>
<td>43.0</td>
<td>29.1</td>
<td>53.5</td>
<td>63.1</td>
</tr>
<tr>
<td>Morton</td>
<td>20.6</td>
<td>40.6</td>
<td>27.3</td>
<td>51.9</td>
<td>61.9</td>
</tr>
<tr>
<td>Hilbert</td>
<td>20.3</td>
<td>40.9</td>
<td>27.9</td>
<td>52.5</td>
<td>62.5</td>
</tr>
</tbody>
</table>

Figure A1. Node Sparsity for 4- and 8-frame Models. Model of longer clip length is more robust to node sparsification.

B.3. Video Sparsity vs. Clip Length

To demonstrate the claim that video sparsity increases with clip length, we evaluate dense models trained with clip length 4 and 8 under different levels of sparsity. As shown in Fig. A1, the 8-frame model is more robust to token pruning with lower keep rates. On DiDeMo, it outperforms 4-frame model by 4% at \( q_v = 0.5 \), while the two models differ by under 2% under dense evaluation. This reveals that longer clips contain greater level of redundancy, and should be modeled with higher sparsity (as done in this work).

B.4. Chunking Strategy

In edge sparsification, the flattened video sequence \( z_{1:N} \) is chunked into subsequences of length \( G \). While this strategy is straightforward and common in language transformers [3, 29], it breaks the spatiotemporal continuity of video data. We investigate an alternative to naïve chunking, by reordering the input tokens using space-filling curves such as Morton [21] and Hilbert [11] curves. This ensures that neighboring tokens in the flattened sequence are close to each other in the original multidimensional space, leading to more localized chunks.

However, early experiments showed no benefit of space-filling token order over naïve flattening, as shown in Tab. A7. This is possibly because video encoders are initialized from image transformers, and block attention with re-ordering prevents video tokens from attending to other spatial locations from the same frame. We leave the study of an optimal chunking strategy for 3D inputs for future work.

B.5. Temporal Probing

To measure the sensitivity of the learned video-text model to temporal cues, we perform an evaluation with shuffled input frames. Tab. A8 shows a performance drop...
of SViT models on retrieval (SSv2) and video QA (ActivityNet) tasks, indicating that the video-text models have learned to reason about the temporal dynamics of video clips. The difference $\Delta$ between normal and shuffled inputs is more prominent on hybrid sparse models, possibly because they attend more to the foreground which contains more temporal variations. Notably, this behavior does not hold for the Singularity model, whose performance is unaffected by frame order. This suggests that late temporal aggregation after spatial global pooling is insufficient to capture spatiotemporal relations across video frames.

B.6. Qualitative Results

Figs. A2 and A3 visualizes the node sparsification patterns generated by visual encoder $f_v$ and multimodal encoder $f_m$. While visual sparsification alone can significantly reduce the number of tokens during forward pass, we find that the cross-modal attention map aligns better with regions of interest in each clip, enabling greater node sparsity in video-text modeling.

C. Limitations & Future Work

While SViT shows great potential towards building long-term video-text models, we recognize that learning temporal relationships from videos would not be possible without high-quality pre-training data. We find that WebVid-2M exists a strong tendency towards spatial appearances: Many videos consist of only simple motions (running, talking etc.), and captions are often highly correlated to the static background. Given this, we suspect that further increasing the clip length beyond 16 frames per video is unlikely to make a significant difference in modeling performance. Building on top of the sparse video-text architecture in this work, future studies can focus on pre-training on video-language datasets and tasks that require aggregating information over a longer period of time span, e.g. narrated egocentric videos over long episodes [9], where SViT may provide larger gains over frame-based approaches and dense spatiotemporal transformers.

References


Figure A2. **Qualitative Results.** We visualize node sparsity patterns generated by visual ($q_v = 0.6$) and cross-modal encoder ($q_m = 0.1$).


A person is standing, grasping their phone, then begins to tidy up the sofa, and begins to sneeze.

A person is undressing and throwing their clothes on the sofa, then sitting on the edge of the bed. The person removes their socks.

A person drinking a glass of water walks down the stairs at the bottom they sit down to take off their shoes and sneeze while untying the laces.

Figure A3. Qualitative Results (continued).


