HiVLP: Hierarchical Interactive Video-Language Pre-Training

Bin Shao, Jianzhuang Liu, Renjing Pei, Songcen Xu, Peng Dai, Juwei Lu, Weimian Li, Youliang Yan
Huawei Noah’s Ark Lab
{shaobin3,liu.jianzhuang,peirenjing,xusongcen, peng.dai,juwei.lu,liweimian,yanyouliang}@huawei.com

Abstract

Video-Language Pre-training (VLP) has become one of the most popular research topics in deep learning. However, compared to image-language pre-training, VLP has lagged far behind due to the lack of large amounts of video-text pairs. In this work, we train a VLP model with a hybrid of image-text and video-text pairs, which significantly outperforms pre-training with only the video-text pairs. Besides, existing methods usually model the cross-modal interaction using cross-attention between single-scale visual tokens and textual tokens. These visual features are either of low resolutions lacking fine-grained information, or of high resolutions without high-level semantics. To address the issue, we propose Hierarchical interactive Video-Language Pre-training (HiVLP) that efficiently uses a hierarchical visual feature group for multi-modal cross-attention during pre-training. In the hierarchical framework, low-resolution features are learned with focus on more global high-level semantic information, while high-resolution features carry fine-grained details. As a result, HiVLP has the ability to effectively learn both the global and fine-grained representations to achieve better alignment between video and text inputs. Furthermore, we design a hierarchical multi-scale vision contrastive loss for self-supervised learning to boost the interaction between them. Experimental results show that HiVLP establishes new state-of-the-art results in three downstream tasks, text-video retrieval, video-text retrieval, and video captioning.

1. Introduction

Recently, the framework of pre-training with large-scale uncurated data and then fine-tuning on some specific downstream tasks has attracted much attention. It firstly emerges in the field of Natural Language Processing (NLP), such as BERT [10], GPT [41] and T5 [42], which are pre-trained on a large corpus of web-scraped dataset and then fine-tuned on a wide variety of NLP downstream tasks. Recently, the framework of pre-training with large-scale uncurated data and then fine-tuning on some specific downstream tasks has attracted much attention. It firstly emerges in the field of Natural Language Processing (NLP), such as BERT [10], GPT [41] and T5 [42], which are pre-trained on a large corpus of web-scraped dataset and then fine-tuned on a wide variety of NLP downstream tasks. Hereafter, it is transferred rapidly to the computer vision area. For example, CLIP [40], ALIGN [16], Florence [58] and BLIP [20] all use more than 100 million open-domain image-text pairs in Image-Language Pre-training (ILP). However, most of the existing Video-Language Pre-training (VLP) works [47, 22, 51, 64, 30] use either a small-scale dataset (e.g., YouCookII [63] with 14K video-text pairs) or a large-scale dataset with less diversity (e.g., Howto100M [34] sourced from 1.22M videos). To solve this problem, we use a larger-scale dataset with 114M image-text pairs and a dataset with 2.5M video-text pairs to pre-train our model. We show that diversity is more important than the total amount of training pairs, and a small set of image-text pairs can achieve much better performance than using millions of video-text pairs. We believe this is a significant way to enhance VLP models and alleviate the

![Figure 1](image-url)
cost of the collection of video-text pairs.

In VLP and ILP, existing works [47] [22] [64] [21] [56] often use cross-attention to model the cross-modal interaction between visual features and text features. However, they usually adopt only the single-scale and low-resolution visual features (i.e., \( \frac{1}{16} \) scale of the input) for cross-attention (CA) blocks, as shown in Figure 1a). This scheme fails to obtain fine-grained interaction with text features and limits the performance of the pre-training model. For fine-grained interactions, [33] injects the high-resolution visual features (i.e., \( \frac{1}{4} \) scale of the input) to CA blocks as shown in Figure 1b), but it does not have high-level semantics. To overcome these limitations, we propose Hierarchical interactive Video-Language Pre-training (HiVLP) that efficiently uses a Hierarchical Visual Feature Group (HVFG) for multi-modal cross-attention. As shown in Figure 1c), HVFG includes different scales of visual features, where the low-resolution ones with high-level semantics are beneficial for global representation and the high-resolution ones with detailed information are useful for fine-grained interaction. Especially, HVFG is able to achieve much better accuracy because of such a multi-scale.

Many works [47] [64] [51] use self-supervised learning to assist the video-language pre-training by reconstructing the masked frame tokens. However, it may introduce noise to interactions between visual and textual features for the masked frame tokens [32]. In this paper, we propose a Multi-level Vision Contrastive (MVC) loss for our HiVLP by applying a global-to-local contrast learning to every scale in HVFG. The MVC loss does not damage the visual tokens and helps the multi-level alignment between visual and textual features.

Our contributions can be summarized as follows:

- To the best of our knowledge, our HiVLP is the first work that uses a hierarchical interaction for video-language pre-training. It is able to effectively learn both the global and fine-grained representations for better alignment between visual and textual features.

- We design a multi-level vision contrastive (MVC) loss for self-supervised learning that can sufficiently mine multi-level visual information to help video-language pre-training.

- We reveal that diversity is more important than the amount of training pairs, and using more diverse image-text pairs benefits a lot for VLP.

- Our HiVLP unifies video-language understanding and generation. It achieves state-of-the-art results in text-video retrieval, video-text retrieval, and video captioning.

2. Related Work

Image-Language Pre-Training (ILP). CLIP [40] is the pioneering work that collects large-scale web data (400M image-text pairs) and achieves competitive zero-shot performance on a variety of downstream tasks [39] [14]. ALIGN [16] is pre-trained with a larger-scale dataset (1.8B) obtaining better performance. FILIP [57] is pre-trained with 300M image-text pairs and designs a cross-modal late interaction mechanism for fine-grained contrastive learning. The key behind their success is that they take the advantages of large-scale datasets which are currently not available in VLP. To deal with this problem, we train our HiVLP model with a hybrid of video-text and image-text pairs.

Video-Language Pre-Training (VLP). Existing VLP works either use a pre-trained S3D [50] to extract visual features as vision input [50] [52] [51] to speed up the training process, or firstly perform pre-training on video-text pairs and then transfer the model to video-language generation tasks [47] [64] [22]. However, both these two training approaches limit the model performance because they are not trained end-to-end [24]. Our HiVLP jointly trains the model with image-text and video-text pairs end-to-end, and unifies both video-language understanding and generation in one framework. FiT [4] also involves pre-training with both image-text and video-text pairs, but can not do video-language generation.

Self-Supervised Learning (SSL). To effectively use datasets, many works [47] [64] [51] use self-supervised learning to assist VLP. VideoBERT [47] tokenizes video frames by hierarchical vector quantization, and then performs SSL by predicting the masked visual tokens. ActBERT [64] predicts the action and object words of masked video tokens. VLM [51] masks either all video tokens or all text tokens, and then uses tokens from one modality to recover masked tokens. However, the existing methods may damage the visual tokens, which introduces noise into cross-attention between visual and textual tokens [32]. For better visual representation learning and avoiding damaging visual tokens, we introduce the MVC loss which maximizes the mutual information between multi-scale global and local representations, and improves the multi-level alignment between visual and textual features.

Hierarchical Interaction. As far as we know, there is no related work about hierarchical interaction in VLP. The most related work is VinVL [59] in ILP. VinVL [59] uses an object detector to extract different sizes of objects as visual features to do cross-attention with textual features. The method of VinVL is complicated and the accuracy of its visual features is limited by the object detector. However, HiVLP makes different scales of visual features interact with textual features without the need of an object detector, which is more effective.
3. Approach

We firstly describe the detailed architecture of HiVLP. Then we present the hierarchical interaction HVFG. Finally, the pre-training objectives are given.

3.1. Architecture

As illustrated in Figure 2(a), HiVLP consists of a vision encoder, a momentum vision encoder, a text encoder, a multi-modal video-text encoder, and a multi-modal video-text decoder. The vision encoder and the text encoder extract image and text features respectively, and do not need to interact with each other for a fast approximate nearest neighbor search in inference [33]. The multi-modal video-text encoder re-ranks the top-k video-text pairs by an additional MLP head that predicts whether they are matched or not. The multi-modal video-text decoder is used to perform the video-language generation (e.g., video captioning). The momentum vision encoder has the same architecture as the vision encoder, and the weights are updated by a momentum-based moving average strategy as in MoCo [15].

The main component of the vision encoder is a hierarchical backbone Swin Transformer [29], followed by a light feature pyramid network (FPN) [25]. Besides, we add a temporal self-attention layer in the last block of Swin Transformer to capture the temporal information, as shown in Figure 2(c). In Figure 2(b), the light FPN uses the output features of $\frac{1}{32}$, $\frac{1}{16}$, and $\frac{1}{8}$ scales from Swin Transformer as inputs, and then gradually upsamples those features for feature fusion. Note that to reduce computation overhead, we do not use $\frac{1}{4}$ scale of visual features in HVFG.

The text encoder’s architecture is the same as BERT [10], which consists of 12 transformer layers. The multi-modal video-text encoder has 12 cross-attention (CA) blocks each with a bi-directional self-attention layer. The multi-modal video-text decoder consists of 12 CA blocks each with a casual self-attention layer.
3.2. Hierarchical Interaction

Existing works often inject single-scale visual features to CA blocks to interact with textual features. For example, ALBEF \cite{21} and BLIP \cite{20} use visual features of $\frac{1}{16}$ scale, which fail to do fine-grained cross-attention. In contrast, \cite{33} adopts features of $\frac{1}{4}$ scale, but these over detailed features without high-level semantics. In our work, we propose a hierarchical interaction mechanism between visual and textual features via injecting HVFG into the CA blocks as shown in Figure 2(a). HVFG includes visual features of three scales $\left\{\frac{1}{4}, \frac{1}{16}, \frac{1}{32}\right\}$. The visual features of $\frac{1}{4}$ scale contain fine-grained information, and those of $\frac{1}{32}$ scale carry high-level semantics. Therefore, HVFG is able to obtain both global and fine-grained representations to boost the interaction between visual and textual features.

3.3. Pre-Training Objectives

The input of the vision encoder is an image or video clip $X \in R^{B \times M \times 3 \times H \times W}$, which consists of $M$ frames with the resolution $H \times W$, where $M = 1$ for images and $B$ is the batch size. The input to the text encoder is a batch of tokenized sequences of words $T \in N^{B \times L}$, where $L$ is the max length of a caption.

In our method, an input video clip $X$ is split into non-overlapping patches $\{x_{i}\}^{B M N}_{i=1}$, $x_{i} \in R^{3 \times P \times P}$, where $P \times P$ is input path size and $N = \frac{H \times W}{P \times P}$. The patches are tokenized by a linear embedding layer which is followed by a linear embedding layer of Swin Transformer.

**Vision-Text Contrastive (VTC) Loss.** VTC aims to pull positive pairs of vision and language representations together and push negative pairs far away. Like ALBEF \cite{21} using a momentum distillation for vision-language pre-training, we introduce two queues to store the most recent $K$ vision-text representations pairs from momentum encoders. Formally, the video-to-text contrastive loss is defined as:

$$L_{V TC} = \frac{1}{B} \sum_{i=1}^{B} \log \frac{\exp(s(f_i, \hat{f}_j^+) / \rho)}{\exp(s(f_i, \hat{f}_j^+) / \rho) + \exp(s(f_i, f_j^-) / \rho)},$$

where $\rho$ is the temperature, $\hat{f}_j^+$ and $\hat{f}_j^-$ are the text features of the positive and negative text samples for the $i$-th vision input $X_i$, respectively, and $f_i$ is the visual feature of $\frac{1}{32}$ scale from $X_i$. $s(f_i, \hat{f}_j^+)$ denotes the cosine similarity between $f_i$ and $\hat{f}_j^+$ that are matched, and $s(f_i, f_j^-)$ denotes the similarity between $f_i$ and $f_j^-$ that are not matched. Symmetrically, the text-to-video contrastive loss is:

$$L_{VT} = \frac{1}{B} \sum_{i=1}^{B} \log \frac{\exp(s(t_i, \hat{f}_j^+) / \rho)}{\exp(s(t_i, \hat{f}_j^+) / \rho) + \exp(s(t_i, f_j^-) / \rho)},$$

Multi-level Vision Contrastive (MVC) Loss. As shown in Figure 3(a), MVC aims to pull the global representation of a scale closer to those local patch representations from the same vision input in different views of a scale. The light FPN has three scales $L_1$, $L_2$, and $L_3$ (see Figure 2(b) and Figure 3(a)). The global feature $f_{ij}$ of scale $j$ of the $i$-th visual input is obtained by the average of all the patch features at $L_j$ (Figure 3(b)). The local features $f_{ijk}$, $k = 1, 2, 3, ..., s_j$ are generated as shown in Figure 3(b) where $f_{ijk}$ is the average feature pooling result of a local window such as $2 \times 2$, and $s_j$ is the number of the local averaged features at $L_j$. Finally, the MVC loss is formulated as:

$$L_{MVC}^j = \frac{\exp(s(f_{ij}, \hat{f}_{ijk}^+) / \rho)}{\exp(s(f_{ij}, \hat{f}_{ijk}^+) / \rho) + \exp(s(f_{ij}, f_{ijk}^-) / \rho)},$$

$$L_{MVC} = \frac{1}{3B} \sum_{i=1}^{B} \sum_{j=1}^{3} -\log L_{MVC}^j,$$

where $\hat{f}_{ij}$ and $\hat{f}_{ijk}$ correspond to $f_{ij}$ and $f_{ijk}$, respectively, but from the other augmentation of the visual input.

Figure 3. (a) Multi-level global-to-local contrastive loss. (b) Generation of global and local features.
Vision-Text Matching (VTM) Loss. Through a MLP layer, VTM is used to judge whether a video-/image-text pair is matched or not with the cross entropy function. There are \( B \) positive pairs from the batch, and \( 2B \) negative pairs are obtained according to the hard negative mining strategy in [21]. This loss is defined as:

\[
L_{VTM} = -\frac{1}{3B} \sum_{i=1}^{3B} \left[ y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \right],
\] (6)

where \( y_i \) is the ground-truth, and \( p_i \) is the output of the MLP layer of the \( i \)-th pair.

Language Modeling (LM) Loss. We inject the hierarchical visual representations into the multi-modal video-text decoder to generate captions. Let \( i \)-th input text be \( T_i = [t_i^1, t_i^2, ..., t_i^L] \). This loss is defined as:

\[
L_{LM} = -\frac{1}{B} \sum_{i=1}^{B} \sum_{l=1}^{L} \log(p(t_i^l | t_i^{l-1}, ..., t_i^1, f_{i1}, f_{i2}, f_{i3})),
\] (7)

where \( p(t_i^l | t_i^{l-1}, ..., t_i^1, f_{i1}, f_{i2}, f_{i3}) \) is the output probability of the multi-modal video-text decoder for the \( l \)-th token \( t_i^l \) given the previously fed tokens \( t_i^{l-1}, ..., t_i^1 \) and the multi-level visual features \( f_{i1}, f_{i2}, f_{i3} \).

Total Loss. Finally, we have the total loss to train HiVLP:

\[
L_{total} = \alpha L_{VTC} + \beta L_{MVC} + \gamma L_{VTM} + \delta L_{LM},
\] (8)

where \( \alpha, \beta, \gamma, \) and \( \delta \) are the weight coefficients of the losses.

4. Experiments

In this section, we first describe the pre-training and downstream datasets, and the implementation details. Then we show the experimental and comparison results, and finally ablate our model.

4.1. Datasets

Pre-Training Datasets. Following FiT [4], we involve two popular datasets (CC-3M [46] and WebVid-2M [4]) for pre-training HiVLP. CC-3M consists of 3M image-text pairs and WebVid-2M includes 2.5M video-text pairs, in total 5.5M image-/video-text pairs. Besides, we use LAION100M, D-14M, and WebVid-2M (total 116.5M image-/video-text pairs) as the larger-scale dataset to train our model, resulting in HiVLP*. LAION100M is a subset of LAION [45]. D-14M is a combined dataset of image-text pairs from COCO [26], Visual Genome [17], CC3M [46], CC12M [6], and SBU captions [35].

Downstream Task Datasets. There are four downstream datasets used in our work for video retrieval. (i) MSR-VTT [53] consists of 10K YouTube videos with 200K human-annotated descriptions. Similar to previous methods [4], we use the split of 9K for training and 1K for testing. (ii) MSVD [7] is a smaller dataset with 1970 videos and 78800 sentences, about 40 descriptions per video. Like [4], we use the split of 1200, 100, and 670 videos for training, validation and testing, respectively. (iii) DiDeMo [3] has 10K videos, each of which is described by multiple sentences, resulting in 40K sentences. For a fair comparison, we follow the setting in [4], where all descriptions for one video are concatenated into a single sentence. (iv) LSMDC [43] contains 128K clips. Like [4], we use 7408 clips for validation and 1000 clips for testing. For video captioning, we evaluate our model on MSR-VTT and MSVD. Following the split of [30], on MSR-VTT, we use 6.5K training videos and 2.9K testing videos; on MSVD, we use 1.2K training videos and 670 testing videos.

4.2. Implementation Details

Our model HiVLP is pre-trained on 16 GPUs (32G memory). Our vision encoder is initialized by Swin-B [29] pre-trained on ImageNet-21k [9]. The light FPN is randomly initialized. The text encoder, multi-modal video-text encoder, and multi-modal video-text decoder are BERTs initialized from BERTbase [10]. HiVLP is pre-trained for 25 epochs, the first 20 epochs only with image-text pairs with a batch size of 30 and the last 5 epochs with both image-text and video-text pairs. Each video clip consists of 4 frames. We use AdamW optimizer with a weight decay of 0.02, and the learning rate is initialized as \( 10^{-5} \) and is warmed up to \( 10^{-4} \) after 3,000 training iterations. We then decrease the learning rate by the cosine decay strategy to \( 10^{-5} \). For the hyper-parameters in Equation 8 we set \( \alpha, \beta, \gamma \) to 1.0 and \( \delta \) to 0.001. The input resolution is 224 × 224 with data augmentations used in [21]. And we use 160 GPUs (16G memory) to train HiVLP* with a batch size of 18 per GPU. For speeding up the training of HiVLP*, we sample 1 frame from each clip.

For the downstream task of text-video retrieval, we sample 4 frames per video for training. Since the multi-modal video-text encoder filters top-\( k \) candidates during inference \( (k \) is set to 128). We sample 8 frames per video for testing on MSR-VTT, MSVD, and LSMDC. Because the videos in DiDeMo are longer, we sample 10 frames for testing from it. For the downstream task of video captioning, we evaluate our model on MSR-VTT and MSVD, with 8 random frames per video for training and 16 frames per video for testing. For all downstream tasks, the initial learning rate is set to \( 5 \times 10^{-6} \), and the weight decay, the batch size, and the total number of epochs are set to 0.05, 64, and 10, respectively.
### Table 1. Comparisons with SOTA text-to-video retrieval methods on MSR-VTT. R@K: Recall@K; MedR: Median Rank; MR: Mean Rank; L-115M: LAION115M; L-100M: LAION100M.

<table>
<thead>
<tr>
<th>Method</th>
<th>Pre-Training Datasets</th>
<th>#Pairs</th>
<th>R@1↑</th>
<th>R@5↑</th>
<th>R@10↑</th>
<th>MedR↓</th>
<th>MR↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero-Shot</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SupportSet</td>
<td>HowTo100M</td>
<td>100M</td>
<td>8.7</td>
<td>23.0</td>
<td>31.1</td>
<td>31.0</td>
<td>20.9</td>
</tr>
<tr>
<td>HD-VILA</td>
<td>HD-VILA-100M</td>
<td>100M</td>
<td>14.4</td>
<td>31.6</td>
<td>41.6</td>
<td>17.5</td>
<td>29.2</td>
</tr>
<tr>
<td>FiT</td>
<td>CC3M, WebVid-2M</td>
<td>5.5M</td>
<td>18.7</td>
<td>39.5</td>
<td>51.6</td>
<td>10.0</td>
<td>36.6</td>
</tr>
<tr>
<td>ALPRO</td>
<td>CC3M, WebVid-2M</td>
<td>5.5M</td>
<td>24.1</td>
<td>44.7</td>
<td>55.4</td>
<td>8.0</td>
<td>41.4</td>
</tr>
<tr>
<td>CLIP</td>
<td>WT400M</td>
<td>400M</td>
<td>30.6</td>
<td>54.4</td>
<td>64.3</td>
<td>4.0</td>
<td>49.8</td>
</tr>
<tr>
<td>Florence</td>
<td>FLD-900M</td>
<td>900M</td>
<td>37.6</td>
<td>63.8</td>
<td>72.6</td>
<td>-</td>
<td>58</td>
</tr>
<tr>
<td>BLIP</td>
<td>L-115M, D-14M</td>
<td>239M</td>
<td>43.3</td>
<td>65.6</td>
<td>74.7</td>
<td>2.0</td>
<td>61.2</td>
</tr>
<tr>
<td>HiVLP</td>
<td>CC3M, WebVid-2M</td>
<td>5.5M</td>
<td>26.4</td>
<td>47.3</td>
<td>55.7</td>
<td>7.0</td>
<td>43.1</td>
</tr>
<tr>
<td>HiVLP*</td>
<td>L-100M, D-14M, WebVid-2M</td>
<td>5.5M</td>
<td>26.4</td>
<td>47.3</td>
<td>55.7</td>
<td>7.0</td>
<td>43.1</td>
</tr>
</tbody>
</table>

### Table 2. Text-to-video results on the MSVD dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>R@1↑</th>
<th>R@5↑</th>
<th>R@10↑</th>
<th>MR↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE</td>
<td>19.8</td>
<td>49.0</td>
<td>63.8</td>
<td>44.2</td>
</tr>
<tr>
<td>SupportSet</td>
<td>28.4</td>
<td>60.0</td>
<td>72.9</td>
<td>53.8</td>
</tr>
<tr>
<td>FiT</td>
<td>33.7</td>
<td>64.7</td>
<td>76.3</td>
<td>57.3</td>
</tr>
<tr>
<td>CLIP4Clip</td>
<td>46.2</td>
<td>76.1</td>
<td>84.6</td>
<td>68.9</td>
</tr>
<tr>
<td>HiVLP</td>
<td>39.1</td>
<td>68.1</td>
<td>77.2</td>
<td>61.5</td>
</tr>
<tr>
<td>HiVLP*</td>
<td>50.2</td>
<td>78.9</td>
<td>85.8</td>
<td>71.6</td>
</tr>
</tbody>
</table>

### Table 3. Text-to-video results on the DiDeMo dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>R@1↑</th>
<th>R@5↑</th>
<th>R@10↑</th>
<th>MR↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClipBERT</td>
<td>20.4</td>
<td>44.5</td>
<td>56.7</td>
<td>40.5</td>
</tr>
<tr>
<td>FiT</td>
<td>34.6</td>
<td>65.0</td>
<td>74.7</td>
<td>58.1</td>
</tr>
<tr>
<td>ALPRO</td>
<td>35.9</td>
<td>67.5</td>
<td>78.8</td>
<td>60.7</td>
</tr>
<tr>
<td>CLIP4Clip</td>
<td>43.4</td>
<td>70.2</td>
<td>80.6</td>
<td>64.7</td>
</tr>
<tr>
<td>HiVLP</td>
<td>40.5</td>
<td>68.7</td>
<td>77.1</td>
<td>62.1</td>
</tr>
<tr>
<td>HiVLP*</td>
<td>48.1</td>
<td>73.8</td>
<td>82.5</td>
<td>68.1</td>
</tr>
</tbody>
</table>

### 4.3. Comparisons with State-of-the-Art Methods

In this section, we compare HiVLP with state-of-the-art (SOTA) methods on three popular video-language downstream tasks (text-video retrieval, video-text retrieval, and video captioning).

**Text-Video Retrieval.** We use MSR-VTT, MSVD, LSMDC, and DiDeMo datasets to evaluate text-video retrieval. We report zero-shot and fine-tuning results (Table 1) on MSR-VTT. For zero-shot retrieval, pre-trained with the same small amount of data, HiVLP outperforms FiT by a large margin (7.4% on R@1). CLIP, Florence, BLIP, and HiVLP* use large-scale datasets for pre-training, with the numbers of pairs 400M, 900M, 239M, and 116.5M, respectively. BLIP performs best among previous works. However, even with much fewer training pairs (116.5M vs. 239M), our HiVLP* outperforms BLIP on all the metrics except obtaining the same MedR. Note that BLIP uses both filtered and synthetic captions in LAION115M and D-14M, while we only use the filtered captions in LAION100M and D-14M.

For fine-tuning comparison, HiVLP/HiVLP* is fine-tuned with the VTC and VTM losses. In Table 4 we see that...
<table>
<thead>
<tr>
<th>Method</th>
<th>MSR-VTT</th>
<th>MSVD</th>
<th>LDMDC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R@1</td>
<td>R@5</td>
<td>R@10</td>
</tr>
<tr>
<td>CLIP4Clip-meanP</td>
<td>43.1</td>
<td>70.5</td>
<td>81.2</td>
</tr>
<tr>
<td>CLIP4Clip-seqTransf</td>
<td>42.7</td>
<td>70.9</td>
<td>80.6</td>
</tr>
<tr>
<td>X-Pool</td>
<td>44.4</td>
<td>73.3</td>
<td>84.0</td>
</tr>
<tr>
<td>HiVLP</td>
<td>51.5</td>
<td>75.9</td>
<td>83.1</td>
</tr>
</tbody>
</table>

Table 5. Comparison with SOTA methods on MSR-VTT, MSVD, and LDMDC for video-text retrieval. + indicates using both CLIP’s image and text encoders as its encoders.

<table>
<thead>
<tr>
<th>Method</th>
<th>#Test Frames</th>
<th>MSVD</th>
<th>MSR-VTT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B4↑</td>
<td>M↑</td>
<td>R↑</td>
</tr>
<tr>
<td>SibNet [27]</td>
<td>30</td>
<td>54.2</td>
<td>34.8</td>
</tr>
<tr>
<td>SAAT [62]</td>
<td>28</td>
<td>46.5</td>
<td>33.5</td>
</tr>
<tr>
<td>STG-KD [36]</td>
<td>16</td>
<td>52.2</td>
<td>36.9</td>
</tr>
<tr>
<td>PMI-CAP [8]</td>
<td>32</td>
<td>54.6</td>
<td>36.4</td>
</tr>
<tr>
<td>ORG-TRL [61]</td>
<td>28</td>
<td>54.3</td>
<td>36.4</td>
</tr>
<tr>
<td>OpenBook [60]</td>
<td>28</td>
<td>66.3</td>
<td>42.4</td>
</tr>
<tr>
<td>HiVLP</td>
<td>16</td>
<td>67.1</td>
<td>45.3</td>
</tr>
<tr>
<td>HiVLP*</td>
<td>16</td>
<td>68.3</td>
<td>45.1</td>
</tr>
</tbody>
</table>

Table 6. Comparison with SOTA methods on MSVD and MSR-VTT for video captioning.

Figure 4. Diversity vs. amount of training pairs.

HiVLP uses the same pre-training datasets as FiT but again outperforms it by a large margin (10.1% on Rank@1). With only about 29% amount of pre-training pairs in CLIP4Clip, HiVLP* exceeds CLIP4Clip on all the metrics significantly. Tables 2, 3, and 4 show the comparison on another three datasets, where HiVLP* outperforms the other methods by a large margin.

**Video-Text Retrieval.** In Table 5 we use HiVLP* to compare with the SOTA methods which are built on CLIP. Our method can beat these CLIP-based methods with fewer pre-training pairs (116.5M vs. 400M).

**Video Captioning.** We also evaluate our model on MSR-VTT and MSVD for video captioning. Four popular metrics BLEU4 [37] (B4), METEOR [5] (M), ROUGE-L [23] (R), and CIDEr [48] (C) are employed. We fine-tune our model with the LM loss for the video-language generation task.

As shown in Table 6, HiVLP works best on all the metrics except one. Note that SwinBERT adopts 64 frames for testing.

### 4.4. Ablation Study

We randomly choose 1/10 image-text pairs from CC3M, resulting in 300k image-text pairs to do ablation study. We surprisingly find that our model pre-trained with only 300k image-text pairs can beat the models SupportSet and HD-VILA pre-trained respectively on Howto100M and HD-VILA-100M, when they are transferred to the MSR-VTT dataset for text-to-video zero-shot retrieval (45.3 vs. 31.1 vs. 41.6 on R@1).

**Image-Text Pairs to Enhance VLP.** As shown in Figure 4 when the number of image-text pairs is increased from 3M to 114M but with the same set of video-text pairs, the Rank@1 on the MSR-VTT dataset raises a lot (from 26.4 to 43.5); when only using much fewer image-text pairs of CC3M than video-text pairs of HD-VILP (0.3M vs. 100M), the Rank@1 performance is much better (17.4 vs. 14.4). It reveals that image-text pairs can bring more diversity than the same amount or more video-text pairs, and more image-text pairs benefit a lot. We believe this is a significant way to enhance the performance of VLP models.

**MVC Loss Design.** We use the HiVLP without the MVC loss as the baseline model. As shown in Table 7 the visual SSL both G2G and G2L-1 are able to improve the performance of the baseline. Using the G2L-1 contrastive loss is better than the G2G contrastive loss, because G2L-1 can dig out local information while G2G cannot. And using more number of different scales of visual features in the MVC loss (i.e., G2L-2 and G2L-3) can further boost the align-
5. Conclusion

We have presented HiVLP, a novel hierarchical interactive video-language pre-training framework. Different from previous methods that input single-scale visual features to cross-attention blocks, HiVLP injects a hierarchical vision feature group (HVFG) to effectively use both global and fine-grained visual features for interaction with textual features. Additionally, our HiVLP is pre-trained with multi-level self-supervised learning that can further improve the model performance. We also reveal that VLP models benefit a lot from the diversity of image-text pairs. Extensive experimental results of downstream tasks (text-video retrieval, video-text retrieval, and video captioning) on 4 popular benchmark datasets show that HiVLP is able to achieve better performance than previous SOTA methods overall by a large margin.

Acknowledgements

We gratefully acknowledge the support of MindSpore [1], CANN (Compute Architecture for Neural Networks) and Ascend AI Processor used for this research.
References


[34] Antoine MICHE, Dimitri ZHUOIK, Jean-Baptiste ALAYRAC, Makarand TAPASWII, Ivan LAPTEV, and Josef SIVIC. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In ICCV, 2019.


