Revitalize Region Feature for Democratizing Video-Language Pre-training

Guanyu Cai\textsuperscript{1,2}, Yixiao Ge\textsuperscript{3}, Alex Jinpeng Wang\textsuperscript{2}, Rui Yan\textsuperscript{2}, Xudong Lin\textsuperscript{4}, Ying Shan\textsuperscript{3}, Lianghua He\textsuperscript{1,}\textsuperscript{*}, Xiaohu Qie\textsuperscript{3}, Jianping Wu\textsuperscript{3}, and Mike Zheng Shou\textsuperscript{2}

\textsuperscript{1}Tongji University
\textsuperscript{2}Show Lab, National University of Singapore
\textsuperscript{3}ARC Lab, Tencent PCG
\textsuperscript{4}Columbia University
\textsuperscript{5}Tsinghua University

Abstract

Recent dominant methods for video-language pre-training (VLP) learn transferable representations from the raw pixels in an end-to-end manner to achieve advanced performance on downstream video-language tasks. Despite the impressive results, VLP research becomes extremely expensive with the need for massive data and a long training time, preventing further explorations. In this work, we revitalize region features of sparsely sampled video clips to significantly reduce both spatial and temporal visual redundancy towards democratizing VLP research at the same time achieving state-of-the-art results. Specifically, to fully explore the potential of region features, we introduce a novel bidirectional region-word alignment regularization that properly optimizes the fine-grained relations between regions and certain words in sentences, eliminating the domain/modality disconnections between pre-extracted region features and text. Extensive results of downstream text-to-video retrieval and video question answering tasks on seven datasets demonstrate the superiority of our method on both effectiveness and efficiency, e.g., our method achieves competing results with 80\% fewer data and 85\% less pre-training time compared to the most efficient VLP method so far \cite{29}. The code will be available at https://github.com/CuthbertCai/DemoVLP.

1 Introduction

Video-language pre-training (VLP) \cite{29,30,37} that jointly learns video and language representations in a self-supervised manner has become the most popular practice to cope with video-language tasks, such as text-to-video retrieval \cite{28,34} and video question answering \cite{3}. Recently, end-to-end methods \cite{29,29,54} that learn video representations from raw pixels have dominated due to their strong performance on downstream tasks. Despite significant progress, these methods are quite data-hungry due to a large number of model parameters and uncurated raw inputs. The pre-training stage turns out to be inefficient and expensive with massive pre-training data and long pre-training time, making it difficult for researchers to pursue research in VLP.

Previous work \cite{29} attempts to lower the barrier for VLP via removing visual redundancy. They point out that video clips with sparsely sampled frames are sufficient enough to capture key semantics for pre-training, since adjacent frames often contain similar scenes. The effort enables more efficient...
People are having car trouble.

He is a boy with a backpack. He rode bicycle to school.
He is going to help another boy learn to ride bicycle.

(a) Region-Word Alignment.

(b) MSRVTT Retrieval.

(c) Pre-training Time.

Figure 1: Our main motivation and results. (a) Our motivation is to reason the detailed correspondence between salient regions and words. (b) and (c) demonstrate that our method significantly improves text-to-video retrieval while reducing pre-training time to a large extent.

VLP with competitive downstream performances. Besides the temporal visual redundancy, we argue that, in contrast to the text with highly abstract semantics, each frame of the video clips also has heavy spatial redundancy.

Towards this end, we further propose to remove the redundant spatial information in sparsely sampled video clips via the claim that a frame is actually worth around 30 objects (based our experiments in Section 4.6). Specifically, we revitalize offline region features that were all the rage in image-language tasks [28, 34] to encourage efficient VLP. Region features are generally pre-extracted by a pre-learned object detector [3]. Rather than the dense and continuous visual signal of the raw pixels, the region features are sparsely distributed with the compact information of salient visual contents, which are the most useful for video-text understanding. We further advocate “less is more” for one more step towards democratizing VLP research.

As is known, methods using off-the-shelf features [28, 34] have been phased out in visual-language tasks due to the inferior downstream performances. Previous work [29] attributes the unsatisfactory pre-training performance of pre-extracted features to their disconnections with the current domain and language modality. We would like to clarify that such disconnections can be properly eliminated by imposing fine-grained cross-modality alignment regularization.

Specifically, besides the common late fusion regularization on the global visual-text representations [4], we introduce a novel bidirectional region-word alignment regularization under the observation that objects extracted from video frames are naturally associated with certain words in the corresponding sentences. For instance, as demonstrated in Fig. 1, the keywords “people”, “car” and “bicycle” share high-level semantics with cropped regions (highlighted with bounding boxes), respectively. To model and promote such a detailed cross-modality relationship, we build bidirectional connections between extracted regions and words.

As demonstrated in Fig. 1, the keywords “people”, “car” and “bicycle” share high-level semantics with cropped regions (highlighted with bounding boxes), respectively. To model and promote such a detailed cross-modality relationship, we build bidirectional connections between extracted regions and words. In the Region→Word manner, we estimate the region-to-sentence similarity resorting to the similarities between each region and all the words in a sentence. The average region-to-sentence similarity over all the regions of a video clip is treated as the video-to-sentence similarity, which is further maximized for positive pairs. Similarly, the Word→Region manner is conducted to measure and optimize the sentence-to-video similarity according to the similarities between each word and the corresponding regions. We surprisingly find that the proposed fine-grained region-word alignment constraints can also be seamlessly integrated into end-to-end VLP methods [4], achieving promising performance gains.

In summary, our contributions are three-fold:

- We revitalize region features towards democratizing VLP via removing both temporal and spatial visual redundancy. Specifically, our efficient VLP model can maintain state-of-the-art performance on multiple downstream tasks with 80% fewer data and 85% less pre-training time than ClipBERT [29], which is the most efficient end-to-end VLP method so far.

- We clarify that the inferior performance of off-the-shelf features in previous attempts [30, 56, 45, 53, 17] lies in the sub-optimal learning regularization. We tackle the challenge with a newly proposed bidirectional region-word constraint, which optimizes fine-grained visual-text relations and properly eliminates the domain/modality disconnections of the region features.
Figure 2: Overview of the proposed method. Given region features of a video and word tokens of a caption, we encode them via the video and language encoder respectively. A global-local alignment is proposed to connect video and language: i) Video-Sentence Alignment maximizes the similarity score of video and language’s [CLS] token; ii) Region-Word Alignment builds up a fine-grained connection between each region feature and word feature.

- Our method shows competitive results on seven downstream tasks, including video-to-text retrieval and video question answering. We surprisingly observe that the introduced region-word alignment regularization can also effectively boost the end-to-end method [4] with noticeable improvements.

2 Related Work

Video-Language Pre-training. Early VLP methods [30, 56, 45, 53, 17, 48] introduce pretrained models on other tasks to pre-extract video representations. Some of them [30, 56, 45] utilize action recognition backbones [15, 19] to pre-extract video representations. These backbones are designed with 2D [20] and 3D [19] CNNs to capture spatial and temporal information in videos. Others [53, 35, 17, 48] fuse multiple “Experts” that are trained on different modalities, such as audio classification [21], OCR [17], image classification [22] and so on, to fully exploit cross-modal high-level semantics in videos. Recently, end-to-end models [37, 29, 3, 54, 16] are proposed. Some [37, 29, 54] utilize CNNs to extract video features, others [4, 16] replace CNNs with ViT [10] to build a pure Transformer-based VLP model.

Region Features in Image-Language Pre-training. Region features extracted by an object detector [3] are adopted by image-language pre-training (ILP) methods [8, 33]. They represent an image to a set of region features that focus on salient regions that are a much more natural basis for attention [12, 44]. Region features enable the interaction between image and text at the object level. Recent ITP methods [33, 8] attempt to form fine-grained alignment between image and text based on region features.

Fine-Grained Alignment between Vision and Language. Fine-grained alignment between visual and textual contents has been explored in image-language retrieval for several years. Some studies [28, 34, 31, 7, 49, 52] propose different attention mechanisms to match visual regions and captions. Recent ITP methods [33, 8] also explore the alignment between regions and words. Considering the low efficiency and large GPU memory consumption for video processing, previous VLP methods [46, 45, 42, 17] adopt simple contrastive loss or binary classification to align video and language. Recently, TACo [51] has attempted to build up a fine-grained alignment. However, it still suffers from so large computation cost that simplifies its original design. In this work, benefiting from our efficient pipeline, we propose a novel region-word alignment method to better connect the video and language.
3 Method

The framework of our method is illustrated in Fig. 2. A video and its corresponding caption are taken as input. The video is initially embedded to region features by an off-the-shelf object detector before being fed into a video encoder. The corresponding caption is encoded to language features by a language encoder. During the pre-training, the video and caption features are optimized towards the objective of global-local alignment. After pre-training, the model can be transferred to a variety of downstream video-language tasks with fine-tuning.

3.1 Model Architecture.

**Input.** Our method takes paired video $V$ and sentence $T$ as raw inputs. Similar to previous VLP methods, $T$ is tokenized into word tokens $\{w_i\}_{i=0}^{L}$, where $L$ denotes the number of tokens in $T$ and $w_0$ denotes the [CLS] token. These word tokens are inputted to language encoder. Video frames $\{f_1, f_2, ..., f_M\}$ are sampled from $V$, where $M$ denotes the number of frames. Given frames $\{f_1, f_2, ..., f_M\}$ of a video, objects and salient regions in each frame are detected by Faster R-CNN pre-trained on Visual Genome. Pooled RoI features of these regions are concatenated together as features $\{o_n \in \mathbb{R}^d\}_{n=1}^N$ of the whole video, where $N$ denotes the number of regions. Video encoder takes these RoI features as the inputs.

In this work, we use pre-extracted region features rather than fine-tuning the detector end-to-end for efficient pre-training. Tuning the detector requires extra computational and memory overhead, and it is also infeasible to acquire bounding box annotations in video-language datasets.

**Video Encoder.** We add a [CLS] token $o_0$ to represent the whole video so that the input becomes $\{o_n\}_{n=0}^N$. To encode location information for each region feature, we use a linear layer $FC$ to project 7-dimensional location vectors $\{l_n = [x_1, y_1, x_2, y_2, w, h, w \cdot h]\}_{n=0}^N$ (normalized top/left/bottom/right coordinates, width, height and area) to $\{l'_n \in \mathbb{R}^d\}_{n=0}^N$:

$$l'_n = FC(l_n)$$  \hspace{1cm} (1)

Learned temporal position embeddings $P \in \mathbb{R}^{M \times d}$ are used to model temporal clues. Video Encoder encodes $\{o_n\}_{n=0}^N$ with a Transformer-based network $E_V$:

$$\{r_n\}_{n=0}^N = E_V(\{o_n + l'_n + P_m\}_{n=0}^N)$$  \hspace{1cm} (2)

where $r_n$ denotes the output region feature, and $m$ denotes $o_n$ is extracted from $m$-th frame. We introduce VIT [10] as $E_V$. Video Encoder makes region features interplay with each other, and outputs refined region features $\{r_n\}_{n=0}^N$, where $r_0$ is encoded from [CLS] token and treated as the global video feature.

**Language Encoder.** Similar to previous VLP methods, Language Encoder is based on BERT [9]. Word tokens are encoded to $\{t_i \in \mathbb{R}^d\}_{i=0}^L$:

$$\{t_i\}_{i=0}^L = E_V(\{w_i\}_{i=0}^L)$$  \hspace{1cm} (3)

where $t_i$ denotes the output language feature and we adopt DistillBERT as $E_L$, which has 40% less parameters than BERT [9] and preserves over 95% of BERT’s performance. The outputs of Language Encoder $\{t_i\}_{i=0}^L$ are then aligned with region features by our proposed pre-training objective.

3.2 Reduce Visual Redundancy.

To democratize video-language pre-training, we attempt to remove both temporal and spatial redundancy. We involve two training strategies to ensure the pre-training time is acceptable.

**Temporal Visual Redundancy - Frame Sparse Sampling.** To save pre-training time and reduce GPU memory consumption, we adopt a sparse frame sampling strategy similar to ClipBERT. During the pre-training period, only a single frame is sampled randomly for each video. When fine-tuning on downstream tasks, a dense sampling strategy is adopted to capture more visual information from videos. We observe that such extremely sparse sampling still achieves competing results on downstream tasks in our framework.
Where \( \alpha \) denotes the temperature coefficient. The local alignment also attempts to maximize video-to-language and language-to-video similarity, but in a fine-grained way. We take video-to-language as an example to illustrate the details of region-word alignment. For \( i \)-th video with region features \( \{ r_n \}_{n=1}^N \) and \( j \)-th sentence with word features \( \{ t_l \}_{l=1}^L \), \( n \)-th region feature firstly attends to each word feature to pick up the most relevant words according to attention weights:

\[
a_{n,l} = \frac{\exp(\langle r_n, t_l \rangle)}{\sum_{k=1}^L \exp(\langle r_n, t_k \rangle)}
\]

where \( \langle \cdot, \cdot \rangle \) denotes the cosine similarity, \( a_{n,l} \) denotes the similarity between the \( n \)-th region feature and \( l \)-th word feature. Intuitively, semantics in videos are usually associated with nouns, verbs and adjectives \([51, 34]\). Other types of words, e.g., prepositions, are not “concrete” enough in videos \([51]\). Thus, to make the region only attend to the most relevant words and ignore other noisy words, we refine \( a_{n,l} \) as follows:

\[
a'_{n,l} = \mathbb{1}[a_{n,l} - \frac{1}{L} \sum_{k=1}^L a_{n,k}] \cdot a_{n,l}
\]

where \( \mathbb{1}[x] = 1 \) if \( x > 0 \) else 0. \( a'_{n,l} \) is the refined similarity between the \( n \)-th region feature and \( l \)-th word feature. Attention weights of irrelevant words are set to 0. By conducting this operation, we ensure the model only focuses on the relevant words with respect to a video.

Given the similarity between region and word features, an attended sentence feature with respect to \( n \)-th region feature is calculated as follows:

\[
\alpha_n = \sum_{l=1}^L a'_{n,l} t_l
\]

where \( \alpha_n \) denotes the attended sentence feature. Thus, the similarity of \( i \)-th video and \( j \)-th caption can be computed as:

\[
S_{i,j} = \frac{1}{N} \sum_{n=1}^N \langle r_n, \alpha_n \rangle
\]
| Method          | Data | GPU Hrs | R@1   |
|-----------------|------|---------|-------|
| Frozen [4]      | 5.8M | 4800    | 31.0  |
| UniVL [36]      | 132M | 2496    | 21.2  |
| HERO [30]       | 7.6M | 8064    | 20.5  |
| VIOLET [16]     | 185.8M | 2240  | 34.5  |
| ClipBERT [29]   | 5.6M | 768     | 22.0  |
| Ours (4F/1.0)   | 5.8M | 1600    | 36.3  |
| Ours (1F/1.0)   | 5.8M | 800     | 36.0  |
| Ours (1F/0.5)   | 2.9M | 416     | 36.4  |
| Ours (1F/0.2)   | 1.2M | 104     | 34.6  |

Table 1: Comparing the pre-training efficiency with existing video-language pre-training methods. 4F means that 4 frames per video are sampled for pre-training. 0.2 means that only 20% pre-training data are used.

The video-to-language loss is calculated with contrastive loss:

\[
\mathcal{L}_{v2l}^{\text{local}} = -\frac{1}{B} \sum_{i} \log \frac{\exp(S_{i,i}/\sigma)}{\sum_{j} \exp(S_{i,j}/\sigma)}
\]  

(10)

where \(\sigma\) denotes the temperature coefficient.

The language-to-video loss \(\mathcal{L}_{l2v}^{\text{local}}\) is calculated in the similar way as \(\mathcal{L}_{v2l}^{\text{local}}\). Given the similarity \(a_{l,n}\) between \(l\)-th word and \(n\)-th region, the attended video vector with respect to \(l\)-th word is calculated:

\[
\beta_{l} = \sum_{n=1}^{N} \mathbb{1}[a_{l,n} = \frac{1}{L} \sum_{n} a_{l,n}] \cdot a_{l,n} \cdot r_{n}
\]  

(11)

where \(\beta_{l}\) denotes the attended video feature. The similarity of \(j\)-th caption and \(i\)-th video is:

\[
S_{j,i} = \frac{1}{L} \sum_{l=1}^{L} \langle t_{l}, \beta_{l} \rangle
\]  

(12)

The language-to-video loss \(\mathcal{L}_{l2v}^{\text{local}}\) is then given as follows:

\[
\mathcal{L}_{l2v}^{\text{local}} = -\frac{1}{B} \sum_{j} \log \frac{\exp(S_{j,j}/\sigma)}{\sum_{j} \exp(S_{j,i}/\sigma)}
\]  

(13)

**Overall Pre-training Objective.** Combining the above four losses, the overall pre-training objective is:

\[
\mathcal{L} = \mathcal{L}_{v2l}^{\text{global}} + \mathcal{L}_{l2v}^{\text{global}} + \mathcal{L}_{v2l}^{\text{local}} + \mathcal{L}_{l2v}^{\text{local}}
\]  

(14)

### 3.4 Complexity Analysis.

Previous offline-feature-based VLP methods [30, 36, 51] are criticized for excessive demand on memory and computation. However, we claim that our method is more efficient that recent end-to-end methods [4, 29, 16] overall. Although we need to pay extra time to extract region features, it is negligible compared with the pre-training time that our method saves, especially we only need to extract them once but can reuse them for plenty of times. When it comes to downstream tasks, the detector only needs to extract features for queries, as we can assume gallery features are pre-extracted and stored in practical. Such extra time is also acceptable considering the performance improvements.

We give a detailed complexity analysis as follow.

**Pre-training.** We compare the pre-training time and R@1 on MSRVTT retrieval of our method with different settings and other methods. Because different methods use different number of GPUs, we introduce GPU Hours, i.e., \(\text{number of GPUs} \times \text{pre-training time}\), to measure the computational cost. As the results shown in Table 1, our method is significantly more efficient than other methods.
Our method achieves the best performance, even when we only use 20% pre-training data and extremely sample only 1 frame per video. Compared with the most efficient VLP method so far, i.e., ClipBERT [29], our method save 85% pre-training time. The reason why the model pretrained with 50% data performs better than those pretrained with more data is that finetuning results may not be strictly correlated with the data scale. Zero-shot results reflect the impact of data scale more precisely. As shown in Fig. 4 and 6, zero-shot results degrade slightly as the number of frames and the size of data decreases.

Besides the whole pre-training, we also explore the running time of a training loop. We compare our method with Frozen [4] and VIOLET [16] to show the efficiency of our method. As the results shown in Fig. 5, our method requires much less time than Frozen [4] and VIOLET [16]. The forward time of our method is only 43.8% of Frozen’s and 9.7% of VIOLET’s. The backward time of our method is only 34.5% of Frozen’s and 9.9% of VIOLET’s.

**Downstream Retrieval.** We test the inference time of each model components on text-to-video retrieval. We build up two galleries: i) test split of LSMDC that contains 1000 samples; ii) 12000 samples collected from WebVid’s train split. In practical, samples in gallery are pre-extracted, thus we record the running time of detection, feature extraction of a query and ranking.

As shown in Table 2, detection costs around 100ms for a video. Feature extraction of texts and videos takes very little time. As the gallery expands, ranking time becomes the major consumption. Considering the gallery capacity is usually larger than 12000 in practical, the extra time for detection is acceptable.

### Table 2: Inference time of each model component. Because the length of sentence and size of video frames are different in WebVid and LSMDC, inference time on language and video encoders is different.

| Components      | 12000 Samples | 1000 Samples |
|-----------------|---------------|--------------|
|                 | Text→Video    | Video→Text   | Text→Video | Video→Text |
| Language Encoder| 5.6ms         | -            | 4.8ms      | -          |
| Video Encoder   | -             | 17.6ms       | -          | 16.4ms     |
| Detector        | -             | 103.2ms      | -          | 103.2ms    |
| Rank            | 372.2ms       | 345.8ms      | 31.8ms     | 30.0ms     |

4 Experiments

#### 4.1 Pre-training Datasets

Following the recent works [4, 16], we involve a video-language dataset and an image-language dataset to pre-train our model. i) WebVid2.5M (WebVid) consists of 2.5M video-language pairs collected from web. WebVid contains manually generated captions that are well-formed sentences and well-aligned with videos. ii) Google Conceptual Captions (CC3M) consists of 3.3M image-language pairs. Images and raw textual descriptions are harvested from the web following a similar process to WebVid.

#### 4.2 Downstream Tasks

We evaluate our method on text-to-video retrieval and video question answering, across 7 downstream tasks. For **Text-to-Video Retrieval**, we report results on i) MSRVTT; ii) DiDeMo; iii) LSMDC and iv) MSVD. We adopt the common R@K (K=1, 5, 10) metric to measure the performance of retrieval.

For **Video Question Answering**, we evaluate the performance on multiple-choice and open-ended settings: i) MSRVTT Multiple Choice; ii) MSRVTT-QA and iii) MSVD-QA. The metric of video question answering tasks is accuracy. More details are provided in Supplementary.

#### 4.3 Implementation Details

We implement our method with Pytorch and train all models on Tesla V100 GPUs with a batch size of 128 per GPU. We use the Adam optimizer with different learning rate schedule on pre-training and
downstream fine-tuning. For pre-training, we train our model in 50 epochs using an initial learning rate of $1 \times 10^{-5}$. The learning rate decays to 1/10 of the previous one at 30 and 40 epochs. For fine-tuning, all experiments on downstream tasks are trained in 10 epochs. We densely sample 8 frames for each video. The initial learning rate is $1 \times 10^{-5}$ and decays to 1/10 at 2, 4 and 8 epochs.

4.4 Analysis of Region-Word Alignment

**Effectiveness.** To demonstrate the effectiveness of our proposed region-word alignment (RWA), we perform our method with and without RWA. For the model without RWA (i.e., Base), the pre-training objective is modified into $\mathcal{L} = \mathcal{L}_{\text{global}}^{vl} + \mathcal{L}_{\text{global}}^{lv}$. Furthermore, we compare RWA to other fine-grained alignment methods, e.g., TACo [51] and FILIP [52].

The results on MSRVTT retrieval are shown in Table 3a. Specifically, RWA improves Base by 13.5 on R@1. Compared to TACo [51] and FILIP [52], RWA brings 6.2 and improvements on R@1, respectively. TACo and FILIP only align each region with the most relevant word, however, a region can correspond to multiple words. Such cases are covered by RWA, thus, RWA brings more gain than TACo and FILIP.

| Method            | Text→Video |   |
|-------------------|------------|---|
|                   | R@1       | R@5 | R@10 |
| Base              | 22.5       | 48.1 | 59.5   |
| TACo [51]         | 29.6       | 59.7 | 72.7   |
| Base+TACo         | 29.8       | 60.1 | 70.7   |
| Base+FILIP        | 27.8       | 55.0 | 65.1   |
| Ours (Base+RWA)   | 36.0       | 61.0 | 71.8   |

(a) Comparison with other visual-language alignment methods on MSRVTT retrieval.

| Method            | Text→Video |   |
|-------------------|------------|---|
|                   | R@1       | R@5 | R@10 |
| Frozen            | 31.0       | 59.5 | 70.5   |
| Frozen+RWA       | 35.1       | 60.7 | 70.9   |
| Zero-shot         | 18.7       | 39.5 | 51.6   |
| Frozen+RWA       | 21.7       | 42.2 | 52.2   |

(b) Results on MSRVTT retrieval when cooperating with methods on MSRVTT retrieval.

**Generalization.** We observe that RWA can seamlessly adapt to end-to-end methods. To prove it, we add RWA to an end-to-end VLP methods, i.e., Frozen [4], which divides the video frame into 196 non-overlapping patches as the input. The results on MSRVTT retrieval are shown in Table 3a. RWA improves R@1 of Frozen by 4.1 and 3.0 for finetuning and zero-shot settings, respectively. The gain demonstrates that RWA can generalize to other end-to-end VLP methods.

4.5 Comparisons with State-of-the-art

We compare our method with state-of-the-art methods on text-to-video retrieval and video question answering tasks. All results in this section is based on the model that samples 1 frame per video during pre-training.

**Text-to-Video Retrieval.** Table 4 summarizes the results on text-to-video retrieval. Across all four tasks, our method achieves significant improvement compared with previous methods in both finetuning and zero-shot settings. On LSMDC, our method outperforms the AVLNet [42] by 8.2 on R@1. Similarly, on MSRVTT, our method surpasses Frozen [4] by 5.0 and 5.3 on R@1 for finetuning and zero-shot settings. For DiDeMo and MSVD, our method brings more than 10 improvement compared with existing best methods. Another advantage of our method is that our zero-shot performance significantly surpasses other methods. This means that our method builds up a good alignment between video and language during pre-training and generalizes well on different datasets.

**Video Question Answering.** Table 5 summarizes the results on video question answering tasks. On MSRVTT MC, our method obtains accuracy of 92.4%, which outperforms prior state-of-the-art VideoCLIP [50] by 0.5%. On MSRVTT-QA and MSVD-QA, our method achieves accuracy of 38.3% and 39.5%. Compared to ClipBERT [29], our method brings 0.9% improvement on MSRVTT-QA. It also surpasses DualVGR [47] by 0.5% on MSVD-QA.
### Table 4: Comparisons with state-of-the-art results on video-language retrieval.

| Method      | R@1 | R@5 | R@10 |
|-------------|-----|-----|------|
| JSFusion [53] | 9.1 | 21.2 | 34.1 |
| MEE [38]    | 9.3 | 25.1 | 33.4 |
| CE [35]     | 11.2 | 26.9 | 34.8 |
| MMT [17]    | 12.9 | 29.2 | 38.8 |
| AVLNet [42] | 17.0 | 38.0 | 48.6 |
| Dig [48]    | 15.8 | 34.1 | 43.6 |
| Frozen [4]  | 15.0 | 34.1 | 39.8 |
| VTMCE [1]   | 14.9 | 33.2 | -    |
| MDMMT [11]  | 18.8 | 38.5 | 47.9 |
| **Ours**    | 25.2 | 45.5 | 54.5 |
| **Zero-shot** | 14.3 | 25.8 | 32.0 |

(a) LSMDC retrieval

| Method      | R@1 | R@5 | R@10 |
|-------------|-----|-----|------|
| MMT [17]    | 26.6 | 57.1 | 69.6 |
| ActBERT [56] | 16.3 | 42.8 | 56.9 |
| SupportSet [41] | 30.1 | 58.5 | 69.3 |
| AVLNet [42] | 27.1 | 55.6 | 66.6 |
| TACo [51]   | 29.6 | 59.7 | 72.7 |
| ClipBERT [29] | 22.0 | 46.8 | 59.9 |
| Frozen [4]  | 31.0 | 59.5 | 70.5 |
| **Ours**    | 36.0 | 61.0 | 71.8 |

(b) MSRVTT retrieval

| Method      | R@1 | R@5 | R@10 |
|-------------|-----|-----|------|
| S2VT [46]   | 11.9 | 33.6 | -    |
| FSE [55]    | 13.9 | 36.0 | -    |
| CE [35]     | 16.1 | 41.1 | 54.4 |
| ClipBERT [29] | 20.4 | 44.5 | 56.7 |
| Frozen [4]  | 31.0 | 59.8 | 72.4 |
| **Ours**    | 41.4 | 67.6 | 77.6 |
| **Zero-shot** | 21.1 | 46.0 | 56.2 |
| **Frozen**  | 29.6 | 53.0 | 65.1 |

(c) DiDeMo retrieval

| Method      | R@1 | R@5 | R@10 |
|-------------|-----|-----|------|
| VSE [25]    | 12.3 | 30.1 | 42.3 |
| VSE++ [13]  | 15.4 | 39.6 | 53.0 |
| MCues [40]  | 20.3 | 47.8 | 61.1 |
| CE [35]     | 19.8 | 49.0 | 63.8 |
| SupportSet [41] | 28.4 | 60.0 | 72.9 |
| Frozen [4]  | 33.7 | 64.7 | 76.3 |
| **Ours**    | 50.9 | 78.9 | 87.0 |
| **Zero-shot** | 41.6 | 69.8 | 80.5 |
| **Frozen**  | 41.6 | 69.8 | 80.5 |

(d) MSVD retrieval

### Table 5: Comparisons with state-of-the-art results on video question answering.

| Method      | MSRVTT | MSVD |
|-------------|--------|------|
| JSFusion [53] | 83.4 | -    |
| ActBERT [56] | 85.7 | -    |
| ClipBERT [29] | 88.2 | -    |
| VideoCLIP [50] | 92.1 | -    |
| MERLOT [54] | 90.9 | -    |
| VIOLET [16] | 91.9 | -    |
| **Ours**    | **92.4** | -    |

(a) MSRVTT Multiple Choice

| Method      | MSRVTT | MSVD |
|-------------|--------|------|
| Co-Mem [18] | 32.0 | 31.7 |
| HMEMA [14] | 33.0 | 33.7 |
| SSML [2] | 35.0 | 35.1 |
| HCRN [27] | 35.6 | 36.1 |
| DualVGR [47] | 35.5 | 39.0 |
| ClipBERT [29] | 37.4 | -    |
| **Ours**    | **38.3** | **39.5** |

(b) MSRVTT QA and MSVD QA

### 4.6 Ablation Study

**Data Scale.** In this section, we explore the impact of different pre-training data scales. We compare models that are pre-trained with different scales of data, i.e., 20%, 50%, 60%, 80% and 100%, on MSRVTT retrieval. As shown in Fig. 4, as the data scale decreases to 50%, R@1 of different scales are very close. Only if the scale drops to 20%, R@1 decreases to 34.6, which is still better than other state-of-the-art methods’ results, e.g., Frozen [4], SupportSet [41] and TACo [51]. The model without RWA has a similar trend to the one with RWA across different scales. It demonstrates that region features are the main reason why our method can be trained with very limited data.

Furthermore, to verify whether our method can generalize to larger datasets or not, we evaluate our method pre-trained with more data on MSRVTT retrieval. Specifically, we add a subset of Conceptual 12M [6] as pre-training data. The subset contains 7M samples that are disjoint from CC3M, denoted as CC7M. As results shown in Table 6, CC7M can further improve the results of our method.
verifies that our method can extend to massive data to achieve better performance. An interesting observation is that adding data from original datasets (i.e., WebVid and CC3M) cannot improve the performance, while adding extra data from a different domain (i.e., CC7M) can. This observation inspires us that the diversity of data plays an important role in VLP.

**Number of Regions.** To determine how many regions can represent a video frame, we vary the number of regions and test our method’s performance on MSRVTT retrieval. As the results shown in Fig. 5, R@1 is improved as the number of regions increases from 10 to 30. When the number continues to increase, the gain on R@1 is not obvious. The results verify our claim that a frame is actually worth around 30 objects.

**Sampling Strategy.** To quantitatively evaluate the impact of sparsely sampling strategy, we compare four models that sample 1, 2, 4, 8 frames per video during pre-training respectively. When conducting downstream tasks, 8 frames are sampled.

The results on MSRVTT retrieval are shown in Fig. 6. In general, the number of frames does not show obvious impact on downstream tasks. As the number of frames varies, the performance for zero-shot and finetuning settings is close. This means that we can sparsely sampling frames during pre-training to save pre-training time.

| Refine Attention | Select Token | Track Object | Temporal Model | Text→Video |
|------------------|--------------|--------------|----------------|------------|
| -                | -            | -            | -              | 34.5 58.3 70.2 |
| ✓                | -            | -            | -              | 36.0 61.0 71.8 |
| -                | ✓            | -            | -              | 32.8 55.5 69.2 |
| ✓                | -            | ✓            | -              | 33.6 59.2 69.7 |
| ✓                | -            | -            | ✓              | 35.9 60.8 71.8 |

Table 7: Ablation study of different model explorations. The first row denotes a baseline and the colored row denotes our method.

**Word of Interest.** We refine attention weights in Eq (7) to choose the most relevant words with respect to a video. To determine the effect of this operation, we compare it with another operation,
i.e., Select Token, which manually selects nouns and verbs as the input caption. Because nouns and verbs usually contain more semantics than other types of words [51]. Results on MSRVTT retrieval are shown in Table 7, our method performs the best, and Select Token even performs worse than baseline. This prove that a proper strategy to find the most relevant words is necessary, while naive manual selection is ineffective.

**Region Selection.** To compare different region selection methods: i) Sorted Selection and ii) Track Object, we give their performance on MSRVTT retrieval. As shown in Table 7, Sorted Selection performs better than Track Object. This means that discriminative region features are more important for VLP than covering more object categories.

**Temporal Modelling.** Frozen utilizes TimeSformer [5] as the backbone, which proposes a time-attention layer to model temporal information. In our work, we rely on temporal position embeddings to model temporal information. To evaluate the impact of time-attention layer, we conduct experiments on MSRVTT retrieval. As shown in Table 7, time-attention layer cannot bring improvement to our method. Thus, in our framework, temporal position embeddings are enough to model temporal clues.

### 4.7 Visualization

We also provide qualitative results on Fig. 7 to show the alignment between video and language. In Fig. 7a, we visualize the attended regions with respect to the most relevant word in the sentence. We observe that not only nouns are connected with corresponding regions, even verbs, such as “see” and “smile”, are also aligned with relevant regions. Fig. 7b shows the attention heatmap between video frames and corresponding sentence. Regions that are relevant to the sentence description show higher attention weights than others. Visualization verifies that our proposed global-local alignment successfully connects video and language from coarse-grained to fine-grained level.

### 5 Conclusion

In this work, we introduce the most efficient video-language pre-training method to date, which revitalizes region features with a bidirectional region-word alignment regularization. Region features that focus on salient areas remove spatial visual redundancy, which enables our VLP method to be extremely time-saving. The region-word alignment constraint builds up a fine-grained connection between video and language. Experimental results on downstream tasks and visualization prove the efficiency and effectiveness of our method.
References

[1] Ali, A., Schwartz, I., Hazan, T., Wolf, L.: Video and text matching with conditioned embeddings. In: CVPR. pp. 1565–1574 (2022)
[2] Amrani, E., Ben-Ari, R., Rotman, D., Bronstein, A.: Noise estimation using density estimation for self-supervised multimodal learning pp. 6644–6652 (2021)
[3] Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: CVPR. pp. 6077–6086 (2018)
[4] Bain, M., Nagrani, A., Varol, G., Zisserman, A.: Frozen in time: A joint video and image encoder for end-to-end retrieval. In: ICCV. pp. 1728–1738 (2021)
[5] Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? arXiv preprint arXiv:2102.05095 (2021)
[6] Changpinyo, S., Sharma, P., Ding, N., Soricut, R.: Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In: CVPR (2021)
[7] Chen, S., Zhao, Y., Jin, Q., Wu, Q.: Fine-grained video-text retrieval with hierarchical graph reasoning. In: CVPR. pp. 10638–10647 (2020)
[8] Chen, Y.C., Li, L., Yu, L., El Kholy, A., Ahmed, F., Gan, Z., Cheng, Y., Liu, J.: Uniter: Universal image-text representation learning. In: ECCV. pp. 104–120 (2020)
[9] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
[10] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. ICLR (2021)
[11] Dzabraev, M., Kalashnikov, M., Komkov, S., Petiushko, A.: Mdmmt: Multidomain multimodal transformer for video retrieval. In: CVPR. pp. 3354–3363 (2021)
[12] Egly, R., Driver, J., Rafal, R.D.: Shifting visual attention between objects and locations: evidence from normal and parietal lesion subjects. p. 161 (1994)
[13] Faghri, F., Fleet, D.J., Kiros, J.R., Fidler, S.: Vse++: Improving visual-semantic embeddings with hard negatives. arXiv preprint arXiv:1707.05612 (2017)
[14] Fan, C., Zhang, X., Zhang, S., Wang, W., Zhang, C., Huang, H.: Heterogeneous memory enhanced multimodal attention model for video question answering. In: CVPR. pp. 1999–2007 (2019)
[15] Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: ICCV. pp. 6202–6211 (2019)
[16] Fu, T.J., Li, L., Zhe Gan, K.L., Wang, W.Y., Wang, L., Liu, Z.: Violet: End-to-end video-language transformers with masked visual-token modeling. In: arXiv:2111.1268 (2021)
[17] Gabeur, V., Sun, C., Alahari, K., Schmid, C.: Multi-modal transformer for video retrieval. In: ECCV. pp. 214–229 (2020)
[18] Gao, J., Ge, R., Chen, K., Nevatia, R.: Motion-appearance co-memory networks for video question answering. In: CVPR. pp. 6576–6585 (2018)
[19] Hara, K., Kataoka, H., Satoh, Y.: Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In: CVPR. pp. 6546–6555 (2018)
[20] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
[21] Hershey, S., Chaudhuri, S., Ellis, D.P., Gemmeke, J.F., Jansen, A., Moore, R.C., Plakal, M., Platt, D., Saouris, R.A., Seybold, B., et al.: Cnn architectures for large-scale audio classification. In: ICSAPP. pp. 131–135 (2017)
[22] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR. pp. 4700–4708 (2017)
[23] Jang, Y., Song, Y., Yu, Y., Kim, Y., Kim, G.: Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In: CVPR. pp. 2758–2766 (2017)
[24] Jiang, J., Chen, Z., Lin, H., Zhao, X., Gao, Y.: Divide and conquer: Question-guided spatio-temporal contextual attention for video question answering. In: AAAI. pp. 11101–11108 (2020)

[25] Kiros, R., Salakhutdinov, R., Zemel, R.S.: Unifying visual-semantic embeddings with multi-modal neural language models. arXiv preprint arXiv:1411.2539 (2014)

[26] Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. IJCV pp. 32–73 (2017)

[27] Le, T.M., Le, V., Venkatesh, S., Tran, T.: Hierarchical conditional relation networks for video question answering. In: CVPR. pp. 9972–9981 (2020)

[28] Lee, K.H., Chen, X., Hua, G., Hu, H., He, X.: Stacked cross attention for image-text matching. In: ECCV. pp. 201–216 (2018)

[29] Lei, J., Li, L., Zhou, L., Gan, Z., Berg, T.L., Bansal, M., Liu, J.: Less is more: Clipbert for video-and-language learning via sparse sampling. In: CVPR. pp. 7331–7341 (2021)

[30] Li, L., Chen, Y.C., Cheng, Y., Gan, Z., Yu, L., Liu, J.: Hero: Hierarchical encoder for video+language omni-representation pre-training. In: EMNLP (2020)

[31] Li, S., Xiao, T., Li, H., Yang, W., Wang, X.: Identity-aware textual-visual matching with latent co-attention. In: CVPR. pp. 1890–1899 (2017)

[32] Li, X., Song, J., Gao, L., Liu, X., Huang, W., He, X., Gan, C.: Beyond rms: Positional self-attention with co-attention for video question answering. In: AAAI. pp. 8658–8665 (2019)

[33] Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., et al.: Oscar: Object-semantics aligned pre-training for vision-language tasks. In: ECCV. pp. 121–137 (2020)

[34] Liu, C., Mao, Z., Liu, A.A., Zhang, T., Wang, B., Zhang, Y.: Focus your attention: A bidirectional focal attention network for image-text matching. In: ACMMM. pp. 3–11 (2019)

[35] Liu, Y., Albanie, S., Nagrani, A., Zisserman, A.: Use what you have: Video retrieval using representations from collaborative experts. BMVC (2019)

[36] Luo, H., Ji, L., Shi, B., Huang, H., Duan, N., Li, T., Li, J., Bharti, T., Zhou, M.: Univl: A unified video and language pre-training model for multimodal understanding and generation. arXiv preprint arXiv:2002.06353 (2020)

[37] Miech, A., Alayrac, J.B., Smaira, L., Laptev, I., Sivic, J., Zisserman, A.: End-to-end learning of visual representations from uncurated instructional videos. In: CVPR. pp. 9879–9889 (2020)

[38] Miech, A., Laptev, I., Sivic, J.: Learning a text-video embedding from incomplete and heterogeneous data. arXiv preprint arXiv:1804.02516 (2018)

[39] Miech, A., Zhukov, D., Alayrac, J.B., Tapaswi, M., Laptev, I., Sivic, J.: Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In: ICCV. pp. 2630–2640 (2019)

[40] Mithun, N.C., Li, J., Metze, F., Roy-Chowdhury, A.K.: Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In: ICMR. pp. 19–27 (2018)

[41] Patrick, M., Huang, P.Y., Asano, Y., Metze, F., Hauptmann, A.G., Henriques, J.F., Vedaldi, A.: Support-set bottlenecks for video-text representation learning. In: ICLR (2021)

[42] Rouditchenko, A., Boggust, A., Harwath, D., Thomas, S., Kuehne, H., Chen, B., Panda, R., Feris, R., Kingsbury, B., Picheny, M., et al.: Cascaded multilingual audio-visual learning from videos. Interspeech pp. 3006–3010 (2021)

[43] Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)

[44] Scholl, B.J.: Objects and attention: The state of the art pp. 1–46 (2001)

[45] Sun, C., Myers, A., Vondrick, C., Murphy, K., Schmid, C.: Videobert: A joint model for video and language representation learning. In: ICCV. pp. 7464–7473 (2019)

[46] Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R., Saenko, K.: Translating videos to natural language using deep recurrent neural networks. arXiv preprint arXiv:1412.4729 (2014)
A Appendix

In this supplementary, we first provide more details of downstream tasks and implementation details. Then we give additional results on more downstream tasks to demonstrate the generalization ability of our method.

B Downstream Tasks.

Text-to-Video Retrieval. i) MSRVTT consists of 10K videos with 200K captions. We follow the previous works [4, 35] to train on 9K train and validation videos and test on the 1K test set. ii) DiDeMo consists of 10K videos with 40K captions. We concatenate all sentences of a video to a single query as previous works [4, 29], and we do not use the ground-truth localization annotations of this dataset. iii) LSMDC contains 128K videos. We follow the setting in [4] to test our method on the 1K test set. iv) MSVD contains 1,970 videos. We split MSVD into 1,200, 100 and 670 videos as the train, validation and test set.

Video Question Answering. i) MSRVTT Multiple Choice is a question answering task that videos are questions and captions are answer candidates. Each video has 5 captions, and only one is the positive one. ii) MSRVTT-QA is based on MSRVTT dataset. 243K open-ended questions and 1500 answers are annotated. iii) MSVD-QA is based on MSVD dataset. It contains a total number of 1,970 videos and 50,505 question-answer pairs.

C Implementation Details.

Text-to-Video Retrieval. We sum up the video-sentence and region-word similarities as the final similarity between video and text. The final similarity is used to rank all video-language pairs.

Multiple Choice. For MSRVTT Multiple Choice, we directly use the model finetuned for MSRVTT retrieval to evaluate the performance. Specifically, we regard videos as questions and regard captions as answers. Each video contains 5 captions, and only one matches the video. By using the model pre-trained for MSRVTT retrieval, we perform retrieval on each video and its corresponding 5 captions. The caption with the highest similarity is the predicted answer.
Figure 8: Overview of the proposed method applied to open-ended QA. We utilize a BUTD [3] head to fuse video and language features to perform open-ended QA tasks.

| Method      | TGIF-FrameQA |
|-------------|--------------|
| ST-VQA [23] | 49.3         |
| Co-Mem [18] | 51.5         |
| PSAC [32]   | 55.7         |
| HMEMA [14]  | 53.8         |
| HCRN [27]   | 55.9         |
| QueST [24]  | 59.7         |
| ClipBERT [29]| 60.3        |
| **Ours**    | **60.6**     |

Table 8: Comparisons with State-of-the-art results on TGIF-FrameQA.

**Open-Ended Question Answering.** For MSRVTT-QA and MSVD-QA that are open-ended question answering tasks, we adopt the framework in BUTD [3], which uses a multi-label classifier based on region features and text features of questions to perform question answering task. As shown in Fig. 8, features of [CLS] from language encoder and region features from video encoder are fused by the BUTD head and conduct classification to choose a proper answer.

**D Additional Experiments.**

To show the generalization of our method, we test it on additional downstream tasks: TGIF-FrameQA.

**D.1 Results on TGIF-FrameQA.**

TGIF-FrameQA is an open-ended QA tasks based on TGIF [23] dataset. TGIF [23] contains 165K QA pairs on 72K animated GIFs. FrameQA requires a model to highlight the fact that questions in this task can be answered according to a video. The answer comes from a dictionary of words of type object, number, color and location.
Results are shown in Table 8. Our method obtains accuracy of 60.6, which outperforms other methods. This experiment is a complementary for video QA tasks. The results also verify that our method works well on video QA tasks.