Revitalize Region Feature for Democratizing Video-Language Pre-training

Guanyu Cai[1,2,*], Yixiao Ge[3], Alex Jinpeng Wang[2], Rui Yan[2], Xudong Lin[4], Ying Shan[3], Lianghua He[1,†], Xiaohu Qie[3], Jianping Wu[5], and Mike Zheng Shou[2]

[1]Tongji University
[2]Show Lab, National University of Singapore
[3]ARC Lab, Tencent PCG
[4]Columbia University
[5]Tsinghua University

Video-Language, Pre-training

Kunyang Han

Video-language pre-training

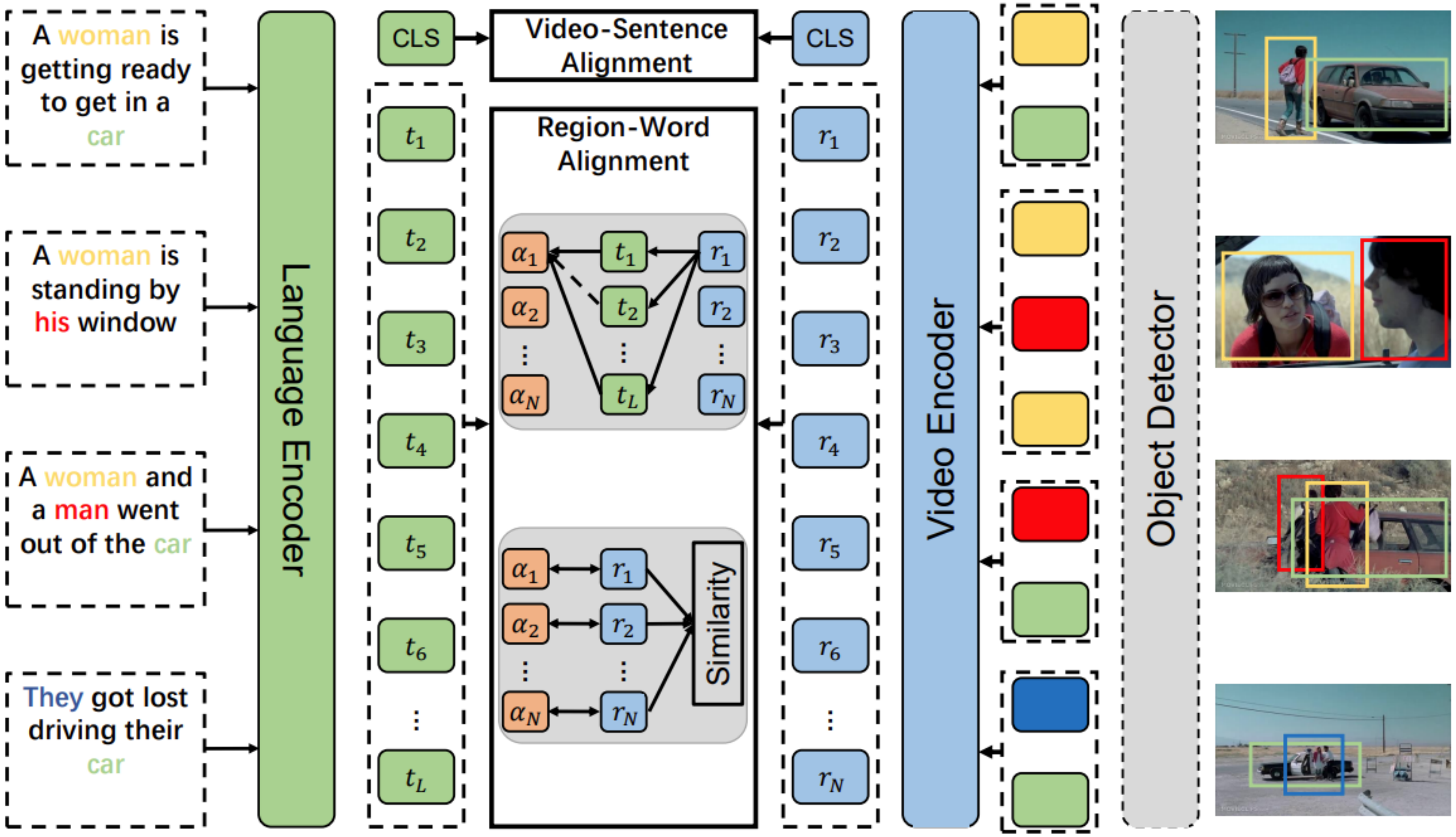
- Jointly learns video and language representations
- Down stream task
  - Text-to-video retrieval
  - Video question answering

Motivation

- Recent methods
  - Data-hungry: massive model parameters, uncurated raw inputs
  - Massive pre-training data and long pre-training time

- Remove visual redundancy
  - Temporal, sparsely sampled video is sufficient
  - Spatial, **a frame is actually worth around 30 objects**

Method – Architecture



Method - Architecture

- Input: paired video V and sentence T

- Video Encoder, $\{r_n\}_{n=0}^{N} = E_V(\{o_n + l_n + \mathbf{P}_m\}_{l=0}^{L})$
  - Pooled RoI, Regions detected by **Faster RCNN**. $\{o_n\}_{n=0}^{N}$
  - [CLS] token o_0 to represent the whole video.
  - Location vector with FC, $\{l_n = [x1, y1, x2, y2, w, h, w * h]\}_{n=0}^{N}$
  - Learned temporal position embeddings, $\mathbf{P}$

- Language Encoder, $\{t_l\}_{l=0}^{L} = E_L(\{w_l\}_{l=0}^{L})$
  - T is tokenized into word tokens $\{w_l\}_{l=0}^{L}$
  - [CLS] token w_0 to represent the whole sentence.

Method - Reduce Visual Redundancy

- Temporal, ClipBERT
  - Pre-training, **single** frame is sampled for each video.
  - Finetuning, dense sampling. (8 frames)
- Spatial
  - Extract **30 region** features per frame.
  - Sorted Selection. Top-k detection confidence.

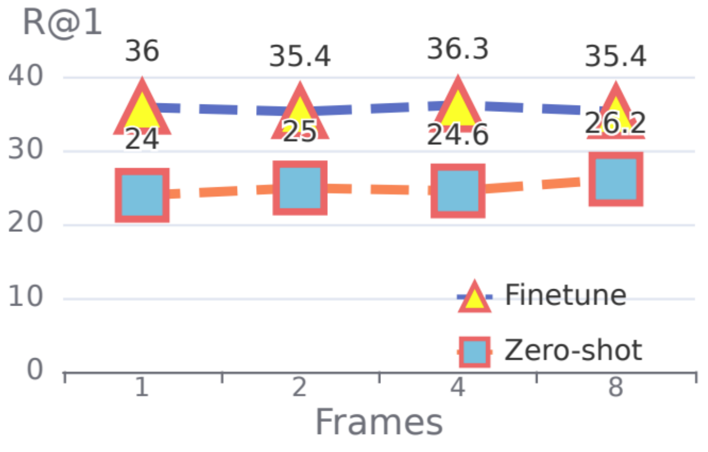


Figure 6: Different number of frames.

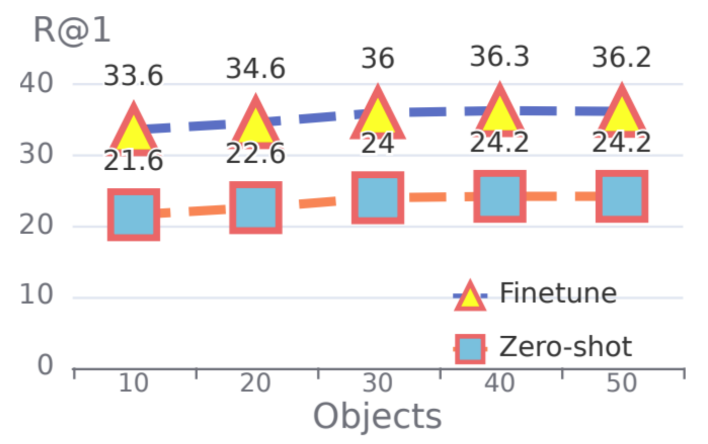


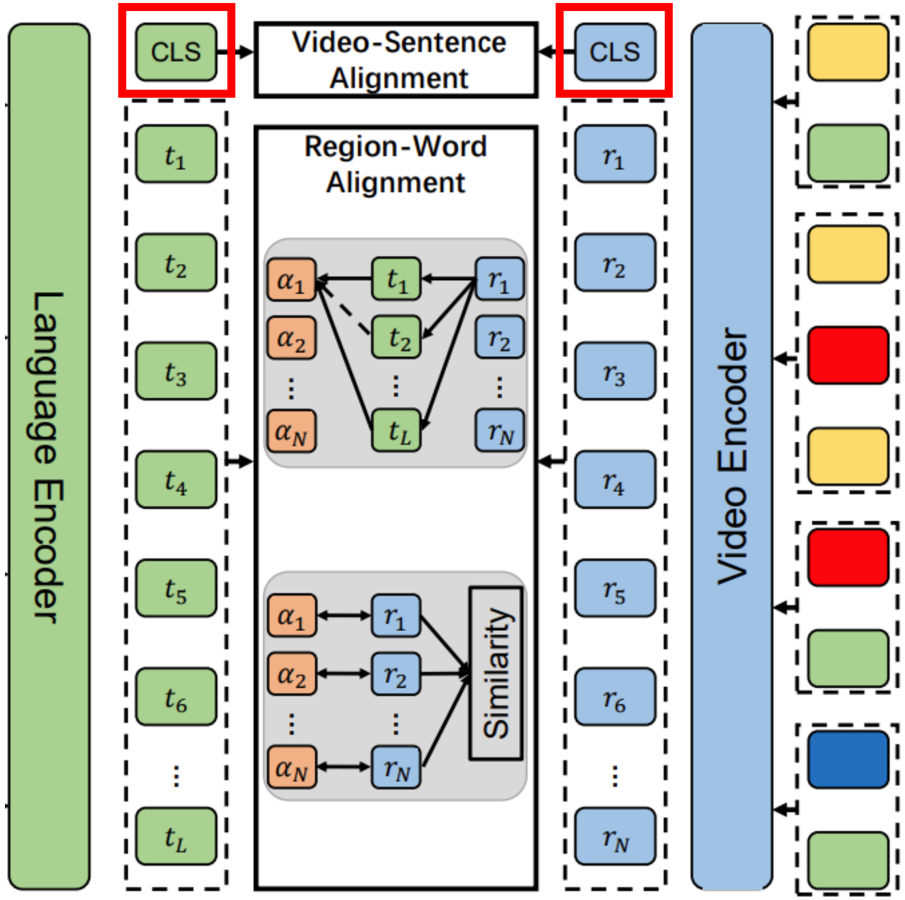
Figure 5: Different number of objects.

Method - Objective

- Video-sentence alignment
  - Contrastive learning with [CLS] in batch

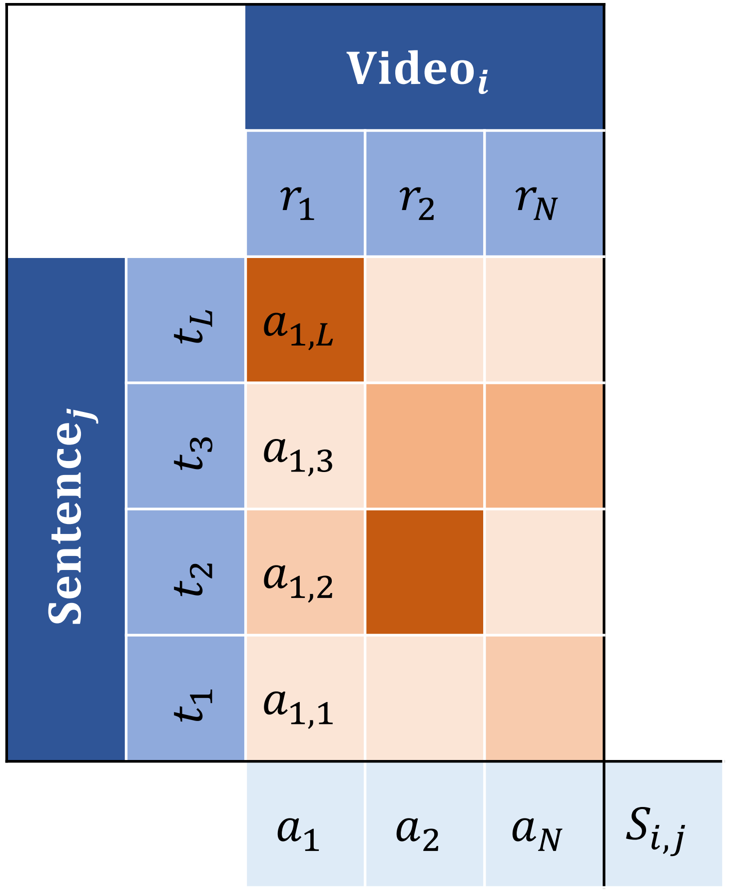
$$\mathcal{L}_{\text{v2l}}^{\text{global}} = -\frac{1}{B}\sum_{i}^{B} \log \frac{\exp(r_{\text{cls}}^{i\,T} t_{\text{cls}}^{i}/\sigma)}{\sum_{j}^{B} \exp(r_{\text{cls}}^{i\,T} t_{\text{cls}}^{j}/\sigma)}$$

$$\mathcal{L}_{\text{l2v}}^{\text{global}} = -\frac{1}{B}\sum_{i}^{B} \log \frac{\exp(t_{\text{cls}}^{i\,T} r_{\text{cls}}^{i}/\sigma)}{\sum_{j}^{B} \exp(t_{\text{cls}}^{i\,T} r_{\text{cls}}^{j}/\sigma)}$$



Method - Objective

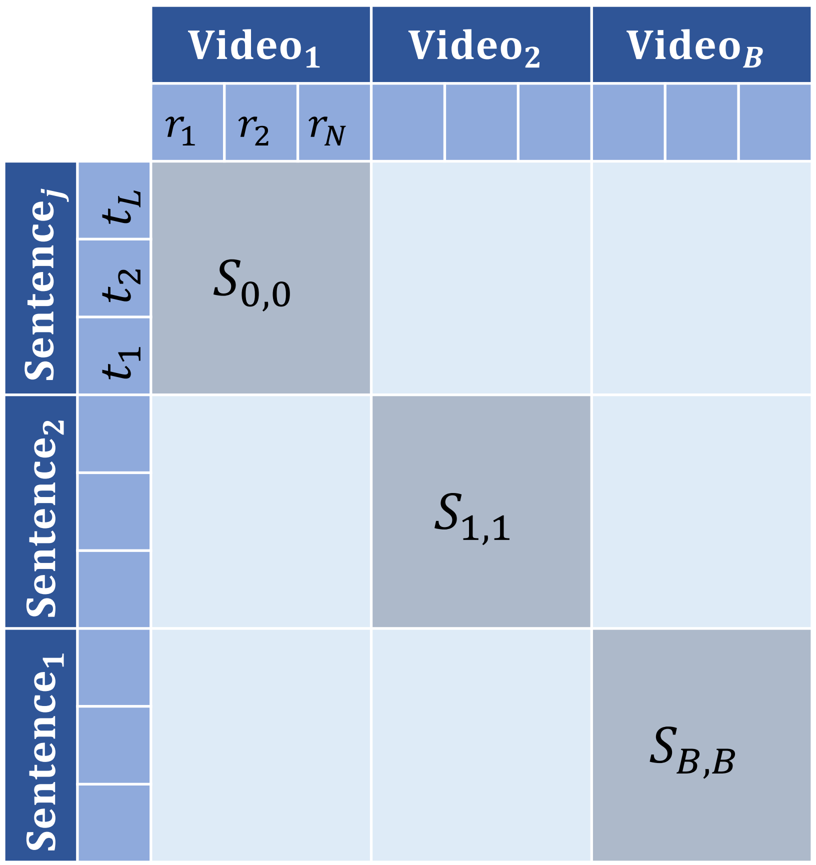
- Region-Word Alignment
  - n-th region, l-th word $$a_{n,l} = \frac{\exp(\langle r_n, t_l \rangle)}{\sum_{k=1}^{L} \exp(\langle r_n, t_k \rangle)}$$
  - n-th region, j-th sentence $$\alpha_n = \sum_{l=1}^{L} a_{n,l} t_l$$
  - i-th video and j-th sentence $$S_{i,j} = \frac{1}{N} \sum_{n=1}^{N} \langle r_n, \alpha_n \rangle$$



Method - Objective

- Region-Word Alignment
  - Final contrastive loss

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



Experiments

- Pre-training Datasets
  - WebVid2.5M, video-language pairs
  - Google Conceptual Captions, 3.3M image-language pairs
- Downstream Tasks
  - Text-to-Video Retrieval, Video Question Answering
- Implementation Details
  - V100 GPUs, batch size of 128 per GPU
  - 50 epochs

Complexity Analysis

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.

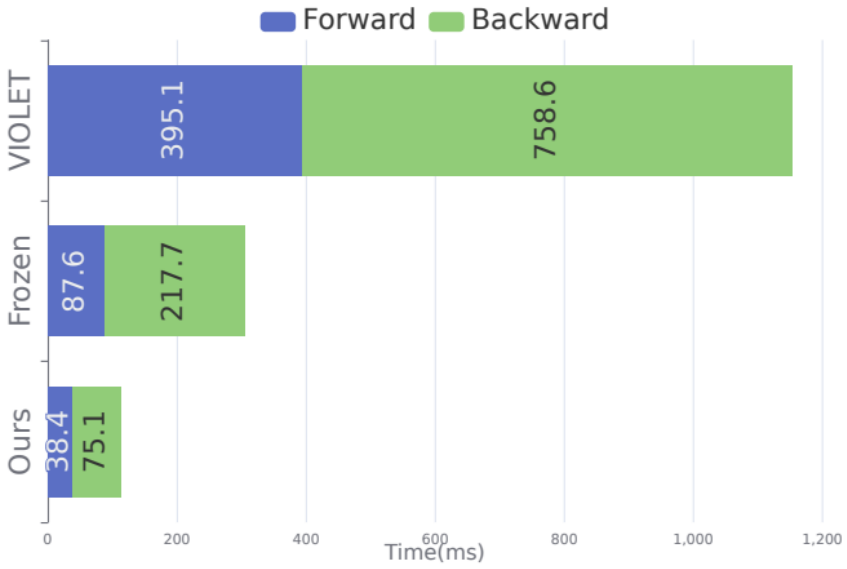


Figure 3: Comparing the running time of a training loop with existing video-language pre-training methods. Batchsize is set to 16 and mixed precision is disabled for all methods.

Downstream task

Method	Text→Video		
	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			
Ours	14.3	25.8	32

(a) LSMDC retrieval

Method	Text→Video		
	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
Zero-shot			
SupportSet [41]	12.7	27.5	36.2
Frozen [4]	18.7	39.5	51.6
Ours	**24.0**	**44.0**	**52.6**

(b) MSRVTT retrieval

Table 4: Comparisons with state-of-the-art results on video-language retrieval.

Downstream task

Method	MSRVTT
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

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