RaP: Redundancy-aware Video-language Pre-training for Text-Video Retrieval

Xing Wu[1,2,3],**Chaochen Gao**[1,2]*, **Zijia Lin**[3],**Zhongyuan Wang**[3],**Jizhong Han**[1],**Songlin Hu**[1,2†]
[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
[3]Kuaishou Technology
{gaochaochen,zangliangjun,hanjizhong,husonglin}@iie.ac.cn
{wuxing,wangzhongyuan}@kuaishou.com, linzijia07@tsinghua.org.cn

Abstract

Video language pre-training methods have mainly adopted sparse sampling techniques to alleviate the temporal redundancy of videos. Though effective, sparse sampling still suffers inter-modal redundancy: visual redundancy and textual redundancy. Compared with highly generalized text, sparsely sampled frames usually contain text-independent portions, called visual redundancy. Sparse sampling is also likely to miss important frames corresponding to some text portions, resulting in textual redundancy. Inter-modal redundancy leads to a mismatch of video and text information, hindering the model from better learning the shared semantics across modalities. To alleviate it, we propose Redundancy-aware Video-language Pre-training. We design a redundancy measurement of video patches and text tokens by calculating the cross-modal minimum dis-similarity. Then, we penalize the high-redundant video patches and text tokens through a proposed redundancy-aware contrastive learning. We evaluate our method on four benchmark datasets, MSRVTT, MSVD, DiDeMo, and LSMDC, achieving a significant improvement over the previous state-of-the-art results. Our code are available at https://github.com/caskcsg/VLP/tree/main/RaP.

1 Introduction

Text-video retrieval computes the semantic similarity between a text query and candidate videos, ranking more similar videos higher. Video-language pre-training can jointly learn the representation of video and text, allowing cross-modal similarity computation to be more effective and efficient, so it has been widely explored in text-video retrieval (Bain et al., 2021; Li et al., 2022a; Lei et al., 2021; Gorti et al., 2022). Videos are composed of dozens or hundreds of consecutive frames, usually containing much redundant information, already known as temporal redundancy. (Lei et al., 2021) proposes to sparsely sample frames from videos to alleviate temporal redundancy without incurring any drop in effect, followed by many works (Bain et al., 2021; Li et al., 2022a; Gorti et al., 2022).

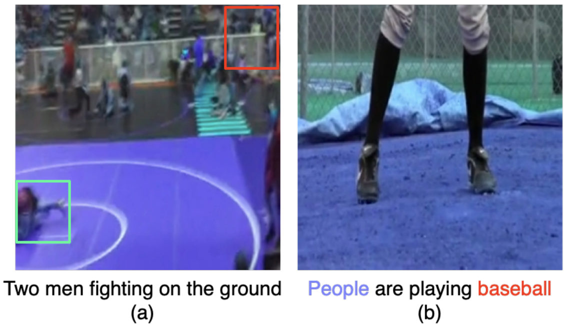

Figure 1: Examples of inter-modal redundancy. (a) Visual redundancy: the pixels in the red box are redundant with respect to the text description. (b) Textual redundancy: the token "baseball" in red font does not correspond to any portion in the video frame.

In addition to intra-modal redundancy, i.e., temporal redundancy, there is inter-modal redundancy between video and text. Some previous works (Zhu and Yang, 2020; Chen et al., 2020; Wang et al., 2022; Li et al., 2022a) focus on modeling fine-grained alignment, which can alleviate inter-modal redundancy to some extent. But they have not categorized and analyzed inter-modal redundancy in details. We summarize inter-modal redundancy into two categories: visual redundancy and textual redundancy, as shown in the example in Figure 1. **Visual redundancy** refers to the redundant information beyond textual semantics that exist in sparsely sampled frames. In contrast to highly generalized text, multiple video frames tend to contain portions that are semantically irrelevant to the text. **Textual redundancy** refers to the redundant portions in the text that are irrelevant to sparsely sampled frames. Sparsely sampling from the video will probably miss important frames associated with some text portions.

*The first two authors contribute equally.
†Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 3036–3047
December 7-11, 2022 ©2022 Association for Computational Linguistics

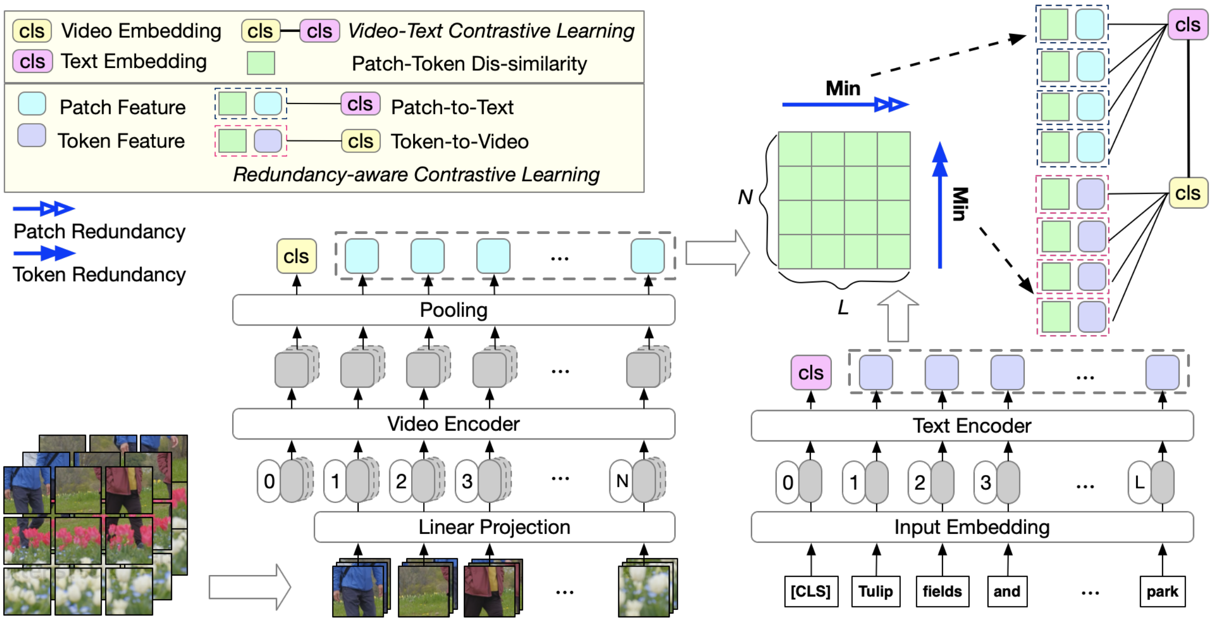

Figure 2: Redundancy-aware video-language pre-training method. Sparsely sampled frames are mapped into video embedding and patch features through multiple layers. Similarly, the text is also mapped into text embedding and token features. The dis-similarity matrix between the patch and token features is used to calculate the redundancy. We take the minimum value by row/column as the redundancy of each patch/token, respectively. Patch/token redundancy is then used for weighted patch-to-text/token-to-video contrastive learning to reduce the impact of high-redundancy patches/tokens.

Inter-modal redundancy will lead to the mismatch of video and text semantics, preventing the model from better learning shared semantics across modalities. Visually redundant pixels are encoded into the video embedding, and pre-training aligns the text embedding with the redundant video embedding, pushing the text embedding away from the correct text semantics. Similarly, pre-training aligns the video embedding with the redundant text embedding, pushing the video embedding away from the correct video semantics. Methods to alleviate redundancy through fine-grained alignment (Zhu and Yang, 2020; Chen et al., 2020; Wang et al., 2022) mainly rely on offline object detectors to extract objects or tags from sampled frames. These methods are based on the assumption that the objects extracted from the sampled frames are related to the text description, or that the tags extracted in one frame relate to other frames. However, there is uncertainty in the correlation between multiple frames, especially with sparse sampling. In addition, object detectors have the drawbacks of inaccurate detection, a limited number of categories, and unable to perform end-to-end optimization in video language pre-training.

To better alleviate the problem of inter-modal redundancy, we first propose a redundancy measurement, as shown in Figure 2. In video-language pre-training, each video frame is split into patches, and a text is tokenized into tokens. Take the Figure 1-(a) as an example. The green patch is low-redundant because it relates to “two”, “men” and “fighting” tokens. In contrast, the red patch is high-redundant because it corresponds to no token. Therefore, **the redundancy of a patch depends on how well it corresponds to the tokens in the text**. In other words, a patch is high-redundant if it has low semantic similarity to all tokens. So we use the minimum dis-similarity between a patch and all tokens as its visual redundancy. Symmetrically, **the redundancy of a token depends on how well it corresponds to the patches in the video**, and we use the minimum dis-similarity between a token and all patches as its textual redundancy.

To reduce the impact of high-redundant patches on learning text embedding or tokens on learning video embedding, we then propose redundancy-aware contrastive learning. We take patch-text pairs as additional positives in video-to-text constrastive learning and assign smaller weights to pairs with high-redundant patches. Similarly, we take token-video pairs as additional positives in text-to-video constrastive learning and assign smaller weights to those with high-redundant tokens. Specially, the weight equals $(1 - redundancy)$ in calculation.

Combining the above two points, we propose **R**edundancy-**a**ware Video-language **P**re-training (RaP) method, which is end-to-end trainable without relying on object detection, as shown in Figure 2. We evaluate RaP on four text-video retrieval datasets, MSRVTT, MSVD, DiDeMo, and LSMDC, achieving a significant improvement over the previous state-of-the-art results. Sufficient ablation studies also confirm the effectiveness of RaP:

Our contributions can be summarized as follows:

1. We summarize the inter-modal redundancy in video-language pre-training and propose a measurement for the redundancy.

2. We propose redundancy-aware contrastive learning to alleviate the two inter-modal redundancy and facilitate high-quality modelling of the shared semantics.

3. Experimental results show that our method significantly improves the state-of-the-art results on multiple text-video retrieval datasets.

2 Related Work

Video-language pre-training aims to learn joint representations between video and language. Videos consist of consecutive frames and often contain visually similar redundant information. Redundant information will bring two problems to video-language pre-training. One is the extra computational overhead, and the other is that the semantics of video and text cannot be well aligned.

To alleviate the problem above, prior approaches(Li et al., 2020; Luo et al., 2020; Miech et al., 2020, 2019; Sun et al., 2019; Zhu and Yang, 2020) use offline tools to extract video features, but cannot achieve end-to-end pre-training. ClipBERT(Lei et al., 2021) efficiently trains the video encoder end-to-end using only a few sparsely sampled frames. Later, some well-performing methods (Bain et al., 2021; Li et al., 2022a; Wang et al., 2022; Fu et al., 2021) adopt the sparse sampling strategy to alleviate temporal redundancy to reduce computational overhead. Since we focus on mitigating the misleading caused by redundant information, rather than temporal redundancy, we follow (Lei et al., 2021) to use sparsely sampled frames as input to the video encoder.

To better align video and text, some recent works introduce fine-grained alignment in video-language pre-training (Zhu and Yang, 2020; Chen et al., 2020; Wang et al., 2022; Li et al., 2022a). Those works identify regions with objects in video via offline trained object detectors or prompters. Then they align regions containing the objects with the text description, which alleviates inter-modal redundancy implicitly. Unlike them, we explicitly propose an efficient redundancy measure that quantifies the impact of different redundancy. The most related work is OA-Trans (Wang et al., 2022), which extracts objects and tags from an anchor frame with an offline detector, and uses the object-related regions or tags as additional input to reduce redundancy. Unlike OA-Trans (Wang et al., 2022), our learnable redundancy measurement matrix can be optimized end-to-end in pre-training. We do not rely on an offline detector, so we do not suffer the drawbacks of offline detectors.

3 Backgroud

This section introduces some background knowledge of video-language pre-training, including video-text input, encoders, and contrastive learning for training.

3.1 Text-Video Input

Video-language pre-training methods use text-video pairs as raw input, where the text is a description of the video. A video $\mathcal{V}$ is sparsely sampled to get K frames, obtaining a sequence of frames $\{\mathcal{F}_k\}_{k=0}^{K}$ Before being fed into the video encoder, each frame $\mathcal{F}_k$ will be divided into N sized frame patches $\{\mathcal{P}_n^k\}_{n=0}^{N}$. Frame patches are then mapped into input embeddings $\{\mathcal{P}e_n^k\}_{n=0}^{N}$ via projection, where $\mathcal{P}e_0^k$ is an additional [CLS] embedding to learn the global semantics of frame $\mathcal{F}_k$. Similarly, before being fed into the text encoder, a text $\mathcal{T}$ will be tokenized into L consecutive tokens and projected into token embeddings $\{\mathcal{T}e_l\}_{l=0}^{L}$, where $\mathcal{T}e_0$ is an additional [CLS] embedding to learn the global semantics of text.

3.2 Text-Video Encoders

Video Encoder We use visual transformer (ViT) (Dosovitskiy et al., 2020) as the video encoder to process each frame $\mathcal{F}_k$ separately. ViT takes frame patch embeddings $\{\mathcal{P}e_n^k\}_{n=0}^{N}$ as input, and output frame patch features $\{\mathcal{P}f_n^k\}_{n=0}^{N}$ corresponding to the $N+1$ positions . Then, we perform mean pooling operation on the features of the same position across K frames. We further transform the pooled feature of each position into a shared normalized low-dimensional (e.g. 256-dim) space, obtaining the video patch feature: $\mathbf{P}_n = \frac{1}{K}\sum_{k=0}^{K}\mathcal{P}f_n^k$. Unless otherwise specified, we will refer to the $\mathbf{P}_n$ as **patch feature** for convenience hereafter. Particularly, $\mathbf{P}_{cls}$ denotes the global **video embedding** in the [CLS] position.

Text Encoder We use BERT (Devlin et al., 2018) as a text encoder to process text, which takes embeddings $\{\mathcal{T}e_l\}_{l=0}^{L}$ as input. The output embedding of each corresponding position of BERT is transformed into the above low-dimensional space as the token feature embedding $\{\mathbf{T}_l\}_{l=0}^{L}$. Unless otherwise specified, we will refer to the $\mathbf{T}_l$ as **token feature** for convenience hereafter. Particularly, $\mathbf{T}_{cls}$ denotes the global **text embedding** in the [CLS] position.

3.3 Video-Text Contrastive Learning

Following CLIP(Radford et al., 2021), we align video and text features into a comparable shared embedding space via contrastive learning. Given the normalized video embedding $\mathbf{P}_{cls}$ and normalized text embedding $\mathbf{T}_{cls}$, the similarity function between video V and text T is:

$$\mathbf{s}(\mathbf{P}_{cls}, \mathbf{T}_{cls}) = \mathbf{P}_{cls} \cdot \mathbf{T}_{cls} \tag{1}$$

We aim to assign higher similarity scores to matched video-text pairs. Therefore, in contrastive learning, we take matched video-text pairs as positives and all other pairs formed a batch as negatives. Given a batch with B matched pairs $\{\mathbf{P}^i_{cls}, \mathbf{T}^i_{cls}\}_{i=1}^{B}$, the video-text contrastive loss of each pair $\{\mathbf{P}^i_{cls}, \mathbf{T}^i_{cls}\}$ consists of two symmetric terms, one for video-to-text contrastive learning:

$$\mathcal{L}_{\text{v2t}} = -\log \frac{\exp(\mathbf{s}\left(\mathbf{P}^i_{cls}, \mathbf{T}^i_{cls}\right)/\tau)}{\sum_{j=1}^{B}\exp(\mathbf{s}\left(\mathbf{P}^i_{cls}, \mathbf{T}^j_{cls}\right)/\tau)} \tag{2}$$

and the other for text-to-video contrastive learning:

$$\mathcal{L}_{\text{t2v}} = -\log \frac{\exp(\mathbf{s}\left(\mathbf{T}^i_{cls}, \mathbf{P}^i_{cls}\right)/\tau)}{\sum_{j=1}^{B}\exp(\mathbf{s}\left(\mathbf{T}^i_{cls}, \mathbf{P}^j_{cls}\right)/\tau)} \tag{3}$$

4 Redundancy-aware Video-language Pre-training

In this section, we introduce our proposed Redundancy-aware video-language pre-training in details. An overview of our approach can refer to Figure 2. First, we introduce how to measure cross-modal redundancy. Then, we introduce how to reduce the impact of redundancy on video-language pre-training.

4.1 Cross-modal Minimum Dis-similarity as Redundancy

We use the similarity function in equation (1) to calculate the cross-modal dis-similarity between a patch feature $\mathbf{P}_n$ and a token feature $\mathbf{T}_l$:

$$\mathbf{d}(\mathbf{P}_n, \mathbf{T}_l) = 1 - \mathbf{s}(\mathbf{P}_n, \mathbf{T}_l) \tag{4}$$

As shown in Figure 2, we calculate the dis-similarity between all *non*-[CLS] patch features $\{\mathbf{P}_n\}_{n=1}^{N}$ and all *non*-[CLS] token features $\{\mathbf{T}_l\}_{l=1}^{L}$, resulting in a dis-similarity matrix $\mathbb{M}$ with dimension $N \times L$. Each row of $\mathbb{M}$ denotes the dis-similarities between a patch feature $\mathbf{P}_n$ and all *non*-[CLS] token features $\{\mathbf{T}_l\}_{l=1}^{L}$. We take the minimum value of $\mathbb{M}$ by the row as the visual redundancy of each patch:

$$\mathbf{vr}_n = \mathbf{min}(\{\mathbb{M}_{nl}\}_{l=1}^{L}) \tag{5}$$

Symmetrically, each column of $\mathbb{M}$ denotes the dis-similarities between a token feature $\mathbf{T}_l$ and all *non*-[CLS] patch features $\{\mathbf{P}_n\}_{n=1}^{N}$. We take the minimum value of $\mathbb{M}$ by the column as the textual redundancy of each token:

$$\mathbf{tr}_l = \mathbf{min}(\{\mathbb{M}_{nl}\}_{n=1}^{N}) \tag{6}$$

4.2 Redundancy-aware Video-Text Contrastive Learning

We use the redundancy of patches and tokens to improve the contrastive learning process.

Redundancy-aware video-to-text contrastive learning In the original video-to-text contrastive learning, there is only one positive text sample for a given video. To reduce the impact of textual redundancy on video embedding learning, we treat all *non*-[CLS] tokens in the positive text as positives too, but we assign higher weights to low-redundancy token features in the loss calculation:

$$\mathcal{L}^{\text{R}}_{\text{v2t}} = -\log \frac{\sum_{l=1}^{L}\mathbf{w}_l \cdot \exp(\mathbf{s}\left(\mathbf{P}^i_{cls}, \mathbf{T}^i_l\right)/\tau)}{\sum_{j=1}^{B}\sum_{l=1}^{L}\exp(\mathbf{s}\left(\mathbf{P}^i_{cls}, \mathbf{T}^j_l\right)/\tau)} \tag{7}$$

, where $\mathbf{w}_l = 1 - \mathbf{tr}_l$. The weighted loss constrains video embeddings to pay more attention to low-redundancy token features while ignoring the high-redundancy ones.

Redundancy-aware text-to-video contrastive learning Symmetrically, in the original text-to-video contrastive learning, there is only one positive video sample for a given text. To reduce the impact of visual redundancy on text embedding learning, we treat all *non*-[CLS] video patches in the positive video as positives too, but we assign higher weights to low-redundancy patch features in the loss calculation:

$$\mathcal{L}^{\mathrm{R}}_{\mathrm{t2v}} = -\log \frac{\sum_{n=1}^{N} \mathbf{w}_n \cdot \exp(\mathbf{s}\left(\mathbf{T}^i_{cls}, \mathbf{P}^i_n\right)/\tau)}{\sum_{j=1}^{B}\sum_{n=1}^{N} \exp(\mathbf{s}\left(\mathbf{T}^i_{cls}, \mathbf{P}^j_n\right)/\tau)} \tag{8}$$

, where the $\mathbf{w}_n = 1 - \mathbf{vr}_n$. The weighted loss constrains text embeddings to pay more attention to low-redundancy patch features while ignoring the high-redundancy ones.

Redundancy-aware contrastive learning Overall, redundancy-aware contrastive learning (RaCL) in both directions constrains the embeddings of one modality to focus more on the low-redundant local features of the other modality. Therefore, RaCL allows different modalities to guide mutually to learn the correct shared semantics. The RaCL loss is defined as the sum of losses in both directions:

$$\mathcal{L}_{RaCL} = \mathcal{L}^{\mathrm{R}}_{\mathrm{v2t}} + \mathcal{L}^{\mathrm{R}}_{\mathrm{t2v}} \tag{9}$$

Video-text pre-training usually trains some auxiliary tasks to help convergence, such as LM tasks, The losses of these tasks are collectively referred to as $\mathcal{L}_{\mathrm{others}}$. So the total loss is calculated as:

$$\mathcal{L} = \mathcal{L}_{\mathrm{others}} + \lambda * \mathcal{L}_{RaCL} \tag{10}$$

, where λ is a balance hyperprarameter. We simply set A to 1.0.

5 Experiments

We conduct experiments on the four most commonly used text-video retrieval benchmark datasets, which will introduced in the 5.3 section. Following existing literature (Li et al., 2022a; Bain et al., 2021), we report Recall@1 (R1), Recall@5 (R5), Recall@10 (R10) and Median Rank (MdR).

5.1 Backbone Network

We use the (Li et al., 2022b) network as the backbone network for our video-language pre-training, with a ViT video encoder and a BERT text encoder. Since this is not the core of this paper, we leave the auxiliary tasks to the appendix A. Considering that our method is an optimization of video-text contrastive learning, our method can also be effective on other backbone networks using contrastive learning.

5.2 Pre-training data

Following the recent works (Li et al., 2022a; Bain et al., 2021; Fu et al., 2021), we jointly pre-train RaP on image-text and video-text datasets, as briefly described below.

WebVid2.5M(WebVid2M)(Bain et al., 2021) contains 2.5M image and text pairs collected from the web. The text data in WebVid describes the global video semantics.

Google Conceptual Captions (CC3M)(Sharma et al., 2018) consists of 3.3M image-text pairs from the web.

During pre-training, we make static videos by duplicate images from CC3M. Thus our pre-training data contains 5.5M video-text pairs, fewer than the widely used HT100M dataset(Miech et al., 2019). Note that, for a fair comparison, we do not compare with the works (Luo et al., 2021; Gorti et al., 2022; Cheng et al., 2021) initialized from CLIP(Radford et al., 2021), which has already been pre-trained on over 400M image-text pairs.

5.3 Text-Video Retrieval Datasets

MSRVTT(Xu et al., 2016) consists of 10K videos, each paired with about 20 human-labeled captions. We train with 7k/9k videos and report results for 1k test split.

DiDeMo(Anne Hendricks et al., 2017) contains 10K Flickr videos, annotated with 40K sentences. Following (Lei et al., 2021; Luo et al., 2020; Liu et al., 2019; Li et al., 2022a), we evaluate paragraph-to-video retrieval, where all sentence descriptions of videos are concatenated into a single query. For a fair comparison with previous methods, we do not use the ground-truth proposals for temporal localization.

MSVD(Chen and Dolan, 2011) contains 1,970 videos from YouTube and 80k English descriptions.Training, validation, and test splits consist of 1,200, 100, and 670 videos, respectively.

Method	R1↑	R5↑	R10↑	MdR↓
Zero-shot				
ActBERT(Zhu and Yang, 2020)‡	8.6	23.4	33.1	36.0
MIL-NCE(Miech et al., 2020)‡	9.9	24.0	32.4	29.5
SupportSet(Patrick et al., 2020)‡	8.7	23.0	31.1	36.0
VideoClip(Xu et al., 2021)‡	10.4	22.2	30.0	-
FiT(Bain et al., 2021)♣	18.7	39.5	51.6	10.0
VIOLET(Fu et al., 2021)♣	25.9	**49.5**	**59.7**	-
HD-VILA(Xue et al., 2022)◇	14.4	31.6	41.6	17.5
OA-Trans(Wang et al., 2022)♣	23.4	47.5	55.6	8.0
ALPRO(Li et al., 2022a)♣	24.1	44.7	55.4	8.0
RaP(ours)♣	**28.9**	47.5	56.8	**7.0**
Fine-tuning on 7k training videos				
JSFusion(Yu et al., 2018)	10.2	31.2	43.2	13.0
HT100M(Miech et al., 2019)‡	14.9	40.2	52.8	9.0
ActBERT(Zhu and Yang, 2020)‡	16.3	42.8	56.9	10.0
HERO(Li et al., 2020)‡	16.8	43.4	57.7	-
AVLNet(Le et al., 2020)‡	27.1	55.6	66.6	4.0
VideoClip(Xu et al., 2021)‡	30.9	55.4	66.8	4.0
NoiseEst(Amrani et al., 2021)‡	17.4	41.6	53.6	8.0
ClipBERT(Lei et al., 2021)	22.0	46.8	59.9	6.0
COTS(Lu et al., 2022)△	32.1	60.8	70.2	3.0
ALPRO(Li et al., 2022a)♣	33.9	60.7	73.2	3.0
RaP(ours)♣	**38.5**	**64.0**	**74.4**	3.0
Fine-tuning on 9k training videos				
SupportSet(Patrick et al., 2020)‡	30.1	58.5	69.3	3.0
FiT(Bain et al., 2021)♣	31.0	59.5	70.5	3.0
VIOLET(Fu et al., 2021)♣	34.5	63.0	73.4	-
COTS(Lu et al., 2022)△	36.8	63.8	73.2	2.0
HD-VILA(Xue et al., 2022)◇	35.0	65.2	**77.2**	3.0
OA-Trans(Wang et al., 2022)♣	35.8	63.4	76.5	3.0
RaP(ours)♣	**40.9**	**67.2**	76.9	**2.0**

Table 1: Comparisons with the state-of-the-art text-to-video retrieval methods with zero-shot and fine-tuning setups on MSRVTT. ♣: Methods using WebVid2M and CC3M datasets(5.5M). ‡: Methods using HT100M(Miech et al., 2019) dataset. ◇:Methods using HD-VILA-100M(Xue et al., 2022) dataset. △:Methods using 15.3M image-text pairs dataset.

LSMDC(Rohrbach et al., 2015) is a clip dataset of 118,081 videos, each with a caption description. The length of the video varies from 2 seconds to 30 seconds. In LSMDC, the training split consists of 101,079 videos, the validation split consists of 7,408 videos. We report results on the test split which contains 1,000 videos.

5.4 Implementation Details

During pre-training, we conduct experiments on 64 NVIDIA V100 GPUs using PyTorch framework(Paszke et al., 2017). We initialize our video encoder with ViT-B/16(Dosovitskiy et al., 2020) with 12 layers. The text encoder is initialized by BERT$_{base}$(Devlin et al., 2018). We randomly sample 4 frames from each video and resize each frame to 256×256. Then we split each resized frame into patches. AdamW(Loshchilov and Hutter, 2017) is adopted as the optimizer with a weight of 0.05. We train the model for 30 epochs with a batch size of 1920 (30 per GPU). The learning rate is initialized as 1e-6 and warmed to 3e-4 after 3,000 training iterations. We select the final checkpoint to fine-tune text-video retrieval datasets.

Method	R1↑	R5↑	R10↑	MdR↓
Zero-shot				
VideoClip(Xu et al., 2021)	16.6	46.9	-	-
FiT(Bain et al., 2021)♣	21.1	46.0	56.2	7.0
VIOLET(Fu et al., 2021)♣	23.5	49.8	59.8	-
OA-Trans(Wang et al., 2022)♣	23.5	50.4	59.8	6.0
ALPRO(Li et al., 2022a)♣	23.8	47.3	57.9	6.0
RaP(ours)♣	**29.5**	**55.7**	**65.6**	**4.0**
Fine-tuning				
ClipBERT(Lei et al., 2021)	20.4	48.0	60.8	6.0
TT-CE(Croitoru et al., 2021)	21.6	48.6	62.9	6.0
FiT(Bain et al., 2021)♣	31.0	59.8	72.4	3.0
VIOLET(Fu et al., 2021)♣	32.6	62.8	74.7	-
ATP(Buch et al., 2022)	26.1	50.5	-	-
HD-VILA(Xue et al., 2022)◇	26.0	54.8	69.0	4.0
OA-Trans(Wang et al., 2022)♣	34.8	64.4	75.1	3.0
ALPRO(Li et al., 2022a)♣	35.9	67.5	78.8	3.0
RaP(ours)♣	**42.9**	**71.2**	**80.2**	**2.0**

Table 2: Comparisons with the state-of-the-art text-to-video retrieval methods with zero-shot and fine-tuning setups on DiDeMo. ♣: Methods using WebVid2M and CC3M datasets(5.5M). ◇:Methods using HD-VILA-100M(Xue et al., 2022) dataset.

During the fine-tuning stage, we perform our experiment on 8 NVIDIA V100 GPUs. We sparsely sample 4 frames and resize them to the same video frame size(256*256) as the pre-training stage. The learning rate is initialized as 1e-5. For each benchmark dataset, we select a checkpoint according to the results of the validation split and use the checkpoint on the test split. For MSRVTT9k without a validation split, we train the model for 10 epochs and choose the final checkpoint. For inference, following(Li et al., 2022a), we uniformly sample 8 frames for each video to ensure reproducibility.

5.5 Experimental Results

We show results on the four text-video retrieval datasets. Since not every previous work has evaluated zero-shot and fine-tuning performance on all datasets, the baseline methods may not be the same for different datasets. In general, our method achieves significant improvements over the compared methods on all datasets.

MSRVTT Results As shown in Table 1, the zero-shot performance of RaP is 3.0% higher than the

Method	R1↑	R5↑	R10↑	MdR↓
Zero-shot				
SupportSet(Patrick et al., 2020)‡	21.4	46.2	57.7	6.0
RaP(ours)♣	**35.9**	**64.3**	**73.7**	**3.0**
Fine-tuning				
VSE(Kiros et al., 2014)	12.3	30.1	42.3	-
VSE++(Faghri et al., 2017)	15.4	39.6	53.0	9.0
Multi.Cues(Mithun et al., 2018)	20.3	47.8	61.1	6.0
CE(Liu et al., 2019)	19.8	49.0	63.8	6.0
SupportSet(Patrick et al., 2020)‡	28.4	60.0	72.9	4.0
FiT(Bain et al., 2021)♣	33.7	64.7	76.3	3.0
OA-Trans(Wang et al., 2022)♣	39.1	68.4	80.3	2.0
RaP(ours)♣	**45.4**	**74.8**	**83.6**	**2.0**

Table 3: Comparisons with text-to-video retrieval state-of-the-art methods with zero-shot and fine-tuning setups on MSVD. We treat each sentence as the textual query. ♣: Methods using WebVid2M and CC3M datasets(5.5M).

Method	R1↑	R5↑	R10↑	MdR↓
Zero-shot				
RaP(ours)♣	12.8	26.6	33.4	37.0
Fine-tuning				
JSFusion(Yu et al., 2018)	9.1	21.2	34.1	36.0
MoEE(Miech et al., 2018)	9.3	25.1	33.4	27.0
CE(Liu et al., 2019)	11.2	26.9	34.8	25.3
MMT(Gabeur et al., 2020)	12.9	29.2	38.8	19.3
AVLNet(Le et al., 2020)‡	17.0	38.0	48.6	-
Dig(Wang et al., 2021)	15.8	34.1	43.6	-
MDMMT(Dzabraev et al., 2021)	18.8	38.5	47.9	-
FiT(Bain et al., 2021)♣	15.0	30.8	39.8	20.0
VIOLET(Fu et al., 2021)♣	16.1	36.6	41.2	-
VTMCE(Ali et al., 2022)	14.9	33.2	-	-
HD-VILA(Xue et al., 2022)◇	17.2	32.9	43.0	16.0
OA-Trans(Wang et al., 2022)♣	18.2	34.3	43.7	18.5
RaP(ours)♣	**19.7**	**39.0**	**47.2**	**13.0**

Table 4: Comparisons with the state-of-the-art text-to-video retrieval methods with zero-shot and fine-tuning setups on LSMDC.♣: Methods using WebVid2M and CC3M datasets(5.5M). ◇:Methods using HD-VILA-100M(Xue et al., 2022) dataset.

previous best-performed method VIOLET in R1 scores. RaP also achieves the second-highest scores in R5 and R10. When fine-tuning on 7k training videos, RaP outperforms all previous fine-tuned methods and is 4.6%, 3.3% higher than ALPRO in R1 and R5. When fine-tuning on 9k training videos, RaP obtains more than 4.1% improvement over COTS in R1 scores.

DiDeMo Results As shown in Table 2, in the zero-shot setting, RaP achieves 5.7% improvement in R1 over the previous best-performed method ALPRO. From the fine-tuning results, RaP outperforms all previous methods. In particular, RaP exceeds ALPRO by +7% on R1. RaP also reduces the MdR metric of DiDeMo to 2.0, which is smaller than the previous models.

Task	R1↑	R5↑	R10↑	MdR↓
Zero-shot				
w/o RaCL	24.5	45.8	54.4	8.0
w/o RaCL$_{t2v}$	26.6	46.0	55.3	7.5
w/o RaCL$_{v2t}$	26.3	46.5	55.1	7.0
with RaCL(RaP)	**28.9**	**47.5**	**56.8**	**7.0**

Table 5: Ablation study of the newly proposed RaCL. We report the results on MSRVTT.

#frms	R1↑	R5↑	R10↑	MdR↓
Zero-shot				
1	21.3	40.2	49.3	11.0
2	25.5	45.8	55.3	7.0
4	**28.9**	**47.5**	56.8	**7.0**
8	26.6	45.2	**57.5**	7.0

Table 6: Ablation study of the number of frames.We report zero-shot results on MSRVTT.

MSVD Results Tabel 3 compares RaP with existing methods on MSVD. The zero-shot performance of RaP surpasses SupportSet and is even slightly higher than FiT's fine-tuning results on R1. After fine-tuning, RaP achieves more than 6.3%, 6.4%, and 3.3% improvement over other fine-tuned models in R1, R5, and R10 scores.

LSMDC Results Due to the ambiguity of text descriptions, LSMDC is a more challenging dataset, and the results of previous methods are relatively low. Table 4 shows that RaP outperforms all previous methods in fine-tuning setup, proving RaP's generalization ability in complex scenarios.

From the performance on these different datasets, RaP significantly outperforms the previous methods, which illustrates the importance of reducing the impact of inter-modal redundancy in video-language pre-training. Meanwhile, the improvement also justifies our redundancy measurement and validates the effectiveness of redundancy-aware contrastive learning.

5.6 Ablations and Analysis

Effect of Redundancy-aware contrastive learning To further verify the effectiveness of redundancy-aware contrastive learning, we compare the experimental results of the model with and without RaCL on the MSRVTT dataset. As shown in the table 5, removing the RaCL model will decrease performance. Therefore, RaCL plays

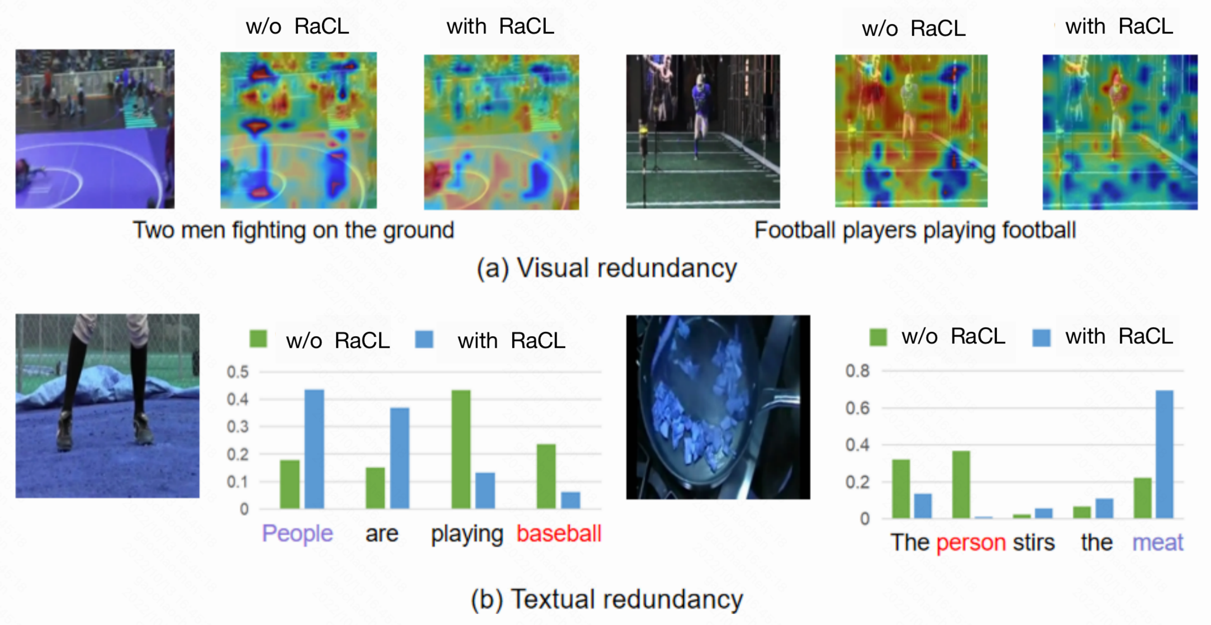

Figure 3: Visualization of the effect in identifying redundancy. We train the model without RaCL as our baseline. (a) Visual redundancy: the text [CLS] embedding is a query, and the video non-[CLS] patch embeddings are keys. The darker the blue, the higher the redundancy. the darker the red, the lower the redundancy. (b) Textual redundancy: the video [CLS] embedding is a query and the text non-[CLS] token embeddings are keys. The higher the value, the lower the redundancy.

a crucial role in the pre-training stages, which is one of the keys that RaP is ahead of others.

Effect of the number of frames We further explore how using different frame numbers in sparse sampling during pre-training affects the performance of RaP: We compare sampling 1, 2, 4, and 8 frames from each video, respectively. We report zero-shot results of the model on the MSRVTT dataset. As shown in Tabel 6, when sampling no more than 4 frames, the model's performance increases as the frame number increases. However, when the number of sampled frames increases to 8, the results begin to drop. We believe this is due to the excessive visual redundancy included when sampling more than 8 frames, making it more difficult for the model to learn. Perhaps larger training data can alleviate this problem, which is left to be explored in future work.

Effect of the coefficient λ In Equation 10, a coefficient λ is used to balance the RaCL loss in the total loss $\mathcal{L}$. As shown in the Table 7, we list the zero-shot results of the model on the MSRVTT dataset when we vary the λ values. We find that using $\lambda = 1.0$ performs the best on R1.

Qualitative Analysis We provide further visualization for qualitative analysis. Specifically, we visualize the weights map between [CLS] embedding from one modal and non-[CLS] embeddings from another, which is a bidirectional process. Fig 3-(a) shows the visualization of the weights allocated to each patch. Compared with the scattered concerns of the baseline model, RaP filters out redundant patches and pays more attention to patches that match the text. For example, RaP focuses on the two men fighting in the lower-left corner of the left frame, and also RaP pays attention to the player in the middle of the right frame. The weights of each token are visualized in Fig 3-(b). Compared with the baseline model, RaP increases the weight of text entities that appear in the frame and decreases the weight of missing entities in the frame. Through RaCL, the [CLS] embedding in RaP can filter redundant information and focus on relevant information between modalities, confirming Rap's significant improvement in the text-video retrieval task.

6 Conclusion

In this paper, we summarize the inter-modal redundancy in video language pre-training and propose a redundancy measurement. Then, we propose redundancy-aware contrastive learning to alleviate redundancy. Significant improvements on sev-

ratio	R1↑	R5↑	R10↑	MdR↓
Zero-shot				
0.5	25.8	46.4	55.4	7.0
1.0	**28.9**	47.5	56.8	7.0
2.0	26.7	**48.7**	**57.5**	**6.0**
4.0	26.8	46.7	56.4	8.0

Table 7: Ablation study of coefficient λ. We report zero-shot results on MSRVTT.

eral text-video retrieval datasets justify our redundancy measurement and validate the effectiveness of redundancy-aware contrastive learning.

7 Limitations

Taking the maximum similarity as the weight is a relatively weak constraint. When the redundant information exceeds a specific limit, it may lead to a decrease in the performance of the model. We guess that introducing an attention module to generate weights in future work may improve the model's performance under high redundancy.

References

A. Ali, I. Schwartz, T. Hazan, and L. Wolf. 2022. Video and text matching with conditioned embeddings. In 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pages 478–487, Los Alamitos, CA, USA. IEEE Computer Society.

Elad Amrani, Rami Ben-Ari, Daniel Rotman, and Alex Bronstein. 2021. Noise estimation using density estimation for self-supervised multimodal learning. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 6644–6652.

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

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

Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. 2022. Revisiting the" video" in video-language understanding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2917–2927.

David Chen and William B Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies, pages 190–200.

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

Xing Cheng, Hezheng Lin, Xiangyu Wu, Fan Yang, and Dong Shen. 2021. Improving video-text retrieval by multi-stream corpus alignment and dual softmax loss. arXiv preprint arXiv:2109.04290.

Ioana Croitoru, Simion-Vlad Bogolin, Marius Leordeanu, Hailin Jin, Andrew Zisserman, Samuel Albanie, and Yang Liu. 2021. Teachtext: Crossmodal generalized distillation for text-video retrieval. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 11583–11593.

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

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Maksim Dzabraev, Maksim Kalashnikov, Stepan Komkov, and Aleksandr Petiushko. 2021. Mdmmt: Multidomain multimodal transformer for video retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3354–3363.

Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2017. Vse++: Improving visual-semantic embeddings with hard negatives. arXiv preprint arXiv:1707.05612.

Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. 2021. Violet: End-to-end video-language transformers with masked visual-token modeling. arXiv preprint arXiv:2111.12681.

Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. 2020. Multi-modal transformer for video retrieval. In European Conference on Computer Vision, pages 214–229. Springer.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. arXiv preprint arXiv:2104.08821.

Satya Krishna Gorti, Noël Vouitsis, Junwei Ma, Keyvan Golestan, Maksims Volkovs, Animesh Garg, and Guangwei Yu. 2022. X-pool: Cross-modal language-video attention for text-video retrieval.

In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5006–5015.

Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. 2014. Unifying visual-semantic embeddings with multimodal neural language models. arXiv preprint arXiv:1411.2539.

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

Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. 2021. Less is more: Clipbert for video-and-language learning via sparse sampling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7331–7341.

Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven CH Hoi. 2022a. Align and prompt: Video-and-language pre-training with entity prompts. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4953–4963.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022b. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. arXiv preprint arXiv:2201.12086.

Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. 2020. Hero: Hierarchical encoder for video+ language omni-representation pre-training. arXiv preprint arXiv:2005.00200.

Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. 2019. Use what you have: Video retrieval using representations from collaborative experts. arXiv preprint arXiv:1907.13487.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101.

Haoyu Lu, Nanyi Fei, Yuqi Huo, Yizhao Gao, Zhiwu Lu, and Ji-Rong Wen. 2022. Cots: Collaborative two-stream vision-language pre-training model for cross-modal retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 15692–15701.

Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. 2020. Univl: A unified video and language pre-training model for multimodal understanding and generation. arXiv preprint arXiv:2002.06353.

Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2021. Clip4clip: An empirical study of clip for end to end video clip retrieval. arXiv preprint arXiv:2104.08860.

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

Antoine Miech, Ivan Laptev, and Josef Sivic. 2018. Learning a text-video embedding from incomplete and heterogeneous data. arXiv preprint arXiv:1804.02516.

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

Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K Roy-Chowdhury. 2018. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval, pages 19–27.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander Hauptmann, Joao Henriques, and Andrea Vedaldi. 2020. Support-set bottlenecks for video-text representation learning. arXiv preprint arXiv:2010.02824.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In International Conference on Machine Learning, pages 8748–8763. PMLR.

Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. 2015. A dataset for movie description. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3202–3212.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In ACL.

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

Jinpeng Wang, Yixiao Ge, Guanyu Cai, Rui Yan, Xudong Lin, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. 2022. Object-aware video-language pre-training for retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3313–3322.

Wenzhe Wang, Mengdan Zhang, Runnan Chen, Guanyu Cai, Penghao Zhou, Pai Peng, Xiaowei Guo, Jian Wu, and Xing Sun. 2021. Dig into multi-modal cues for video retrieval with hierarchical alignment. In IJCAI, pages 1113–1121.

Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. 2021. Videoclip: Contrastive pre-training for zero-shot video-text understanding. arXiv preprint arXiv:2109.14084.

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

Hongwei Xue, Tiankai Hang, Yanhong Zeng, Yuchong Sun, Bei Liu, Huan Yang, Jianlong Fu, and Baining Guo. 2022. Advancing high-resolution video-language representation with large-scale video transcriptions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5036–5045.

Jinyu Yang, Jiali Duan, Son Tran, Yi Xu, Sampath Chanda, Liqun Chen, Belinda Zeng, Trishul Chilimbi, and Junzhou Huang. 2022. Vision-language pre-training with triple contrastive learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 15671–15680.

Youngjae Yu, Jongseok Kim, and Gunhee Kim. 2018. A joint sequence fusion model for video question answering and retrieval. In Proceedings of the European Conference on Computer Vision (ECCV), pages 471–487.

Linchao Zhu and Yi Yang. 2020. Actbert: Learning global-local video-text representations. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 8746–8755.

A Appendix

A.1 Auxiliary Pre-training Tasks

Video-Text Matching Task Following (Li et al., 2022b), we inject visual information by inserting an additional cross-attention layer between the self-attention layer and the feed-forward network of each transformer block of the text encoder. The modified text encoder shares parameters with the text encoder except for the newly injected layers. As shown in Figure 4-(a),we construct the input by wrapping the text with a task-specific prompt, i.e. “[encoded] text”. The output embedding of the additional “[Encode]” is the fused multimodal representation of the video and text, which is used to determine whether the video matched the text.

Video-grounded Language Model Task Following (Li et al., 2022b), we replaces the bidirectional self-attention layers in the video-grounded text encoder with causal self-attention layers. The modified text encoder shares parameters with the text encoder except for the replaced layers. As shown in Figure 4-(b), we construct the input by wrapping the text with a task-specific prompt, i.e. “[Decode] text”. An end-of-sequence token “[EOS]” is used to signal the end of the decoding process.

Intra-Modal Contrastive Learning Tasks Following (Yang et al., 2022), we perform intra-modal contrastive learning on text and video separately to promote a uniform distribution of representations, as shown in Figure 4-(c). For the video modality, A video will generate two views after data augmentation. We consider these two views as a positive pair. For the text modality, we follow (Gao et al., 2021) and use standard dropout as a minimal data augmentation method.

These tasks are jointly trained with the redundancy-aware contrastive learning task to optimize the model's parameters together.

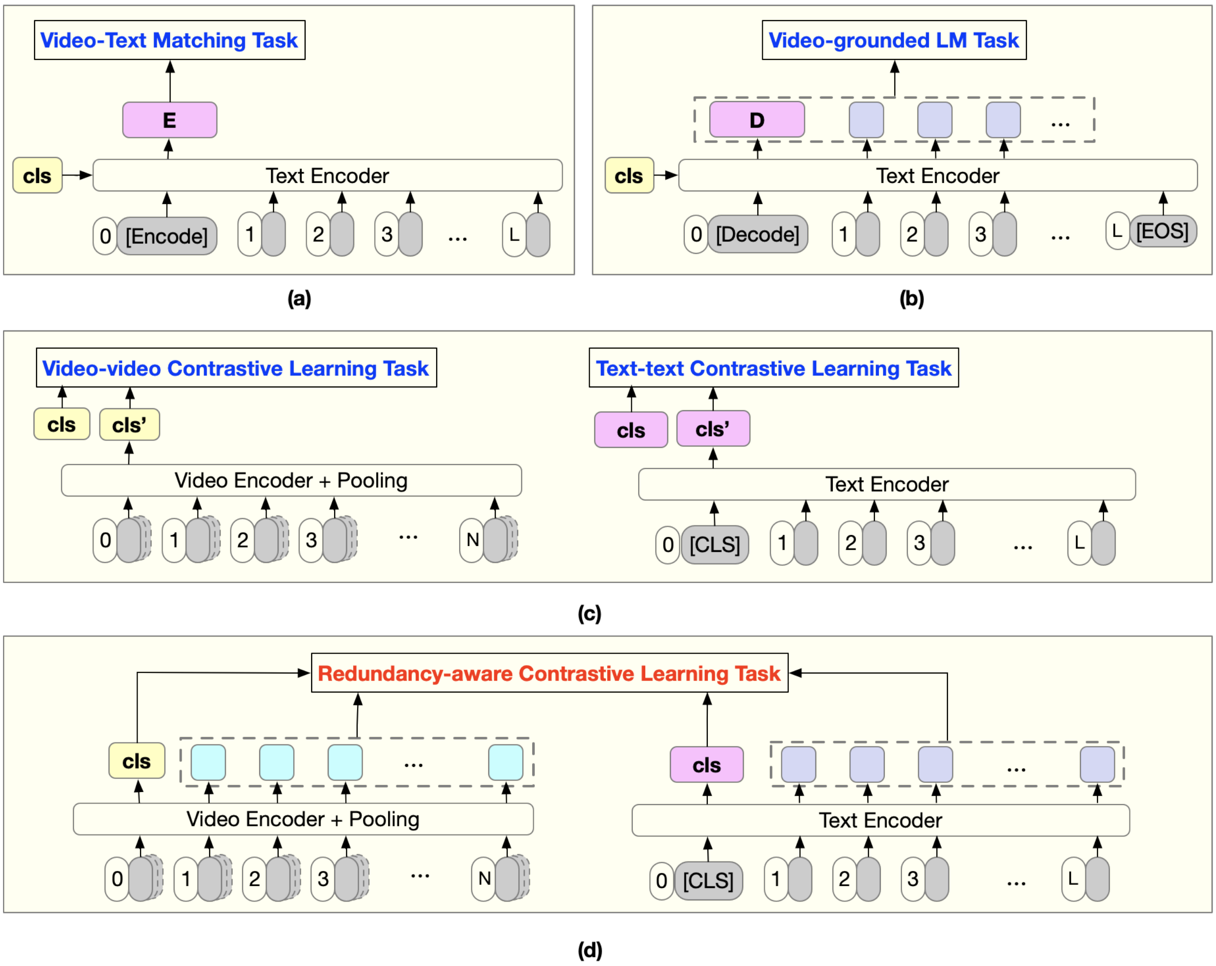

Figure 4: The Pre-training tasks. In addition to our designed redundancy-aware contrastive learning task, there are auxiliary tasks: a video-text matching task, a video-grounded language model task and two intra-modal contrastive learning tasks.