

Disentangled Representation Learning for Text-Video Retrieval

Qiang Wang, Yanhao Zhang, Yun Zheng, Pan Pan, Xian-Sheng Hua

DAMO Academy, Alibaba Group

{qishi.wq, yanhao.zyh, zhengyun.zy, panpan.pp, xiansheng.hxs}@alibaba-inc.com

Abstract. Cross-modality interaction is a critical component in Text-Video Retrieval (TVR), yet there has been little examination of how different influencing factors for computing interaction affect performance. This paper first studies the interaction paradigm in depth, where we find that its computation can be split into two terms, the interaction contents at different granularity and the matching function to distinguish pairs with the same semantics. We also observe that the single-vector representation and implicit intensive function substantially hinder the optimization. Based on these findings, we propose a disentangled framework to capture a sequential and hierarchical representation. Firstly, considering the natural sequential structure in both text and video inputs, a **Weighted Token-wise Interaction (WTI) module** is performed to decouple the content and adaptively exploit the pair-wise correlations. This interaction can form a better disentangled manifold for sequential inputs. Secondly, we introduce a **Channel DeCorrelation Regularization** (CDCR) to minimize the redundancy between the components of the compared vectors, which facilitate learning a hierarchical representation. We demonstrate the effectiveness of the disentangled representation on various benchmarks, *e.g.*, surpassing CLIP4Clip largely by +2.9%, +3.1%, +7.9%, +2.3%, +2.8% and +6.5% R@1 on the MSR-VTT, MSVD, VATEX, LSMDC, AcitivityNet, and DiDeMo, respectively.

Keywords: Video Retrieval, Cross-modality Interaction, Decorrelation

1 Introduction

Text-Video Retrieval (TVR) has significant worth with the explosive growth of video content on the internet. Alignment between different modalities requires particular consideration of both intra-modal representation and cross-modality interaction. The inherent asymmetry across different modalities raises an expression challenge for Text-Video Retrieval. The pioneering works [12,26] incorporate multi-modality features and aggregate information from different pre-trained experts to boost performance. While the open-ended text queries and diverse vision contents require tremendous labor.

With the great success of NLP pre-training [8,27,22,36,37], Vision-Language Pre-Training (VLPT) has received increasing attention [23,21,24,28,2]. Recently,

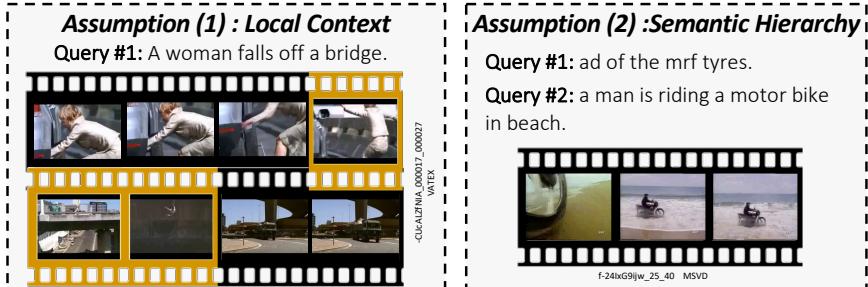


Fig. 1. Illustrations of common underlying assumptions for Text-Video Retrieval. (1) The description could be relevant to the local video context. (2) Multiple distinct sentences can hint at the same video. Our disentangled representation addresses both assumptions through token-wise interaction and channel decorrelation.

the Contrastive Language–Image Pre-training (CLIP [35]) leverages a massive amount of image-caption pairs to learn generic vision and language representations, showing impressive improvements in diverse vision-language (VL) tasks. Researchers [34,2,29,11,13] explore the power of this design principle to text-video retrieval with additional temporal fusion module. CLIP4Clip [29] utilizes a temporal transformer to aggregate sequential features into a *single* high-dimensional representation, and retrievals by dot-product search.

However, the text-video retrieval is uniquely characterized by its sequential frames. Thus we highlight the common underlying assumptions behind TVR in Figure 1: (1) the description related with the local segment still needs to be retrieved, and (2) human-generated sentences naturally have a hierarchical structure and can describe in different views. Based on these perspectives, we can re-examine the existing algorithmic framework, especially the **interaction** components.

Figure 2 shows typical interaction archetypes and the proposed variations. In determining the process flow for the interaction block, only a few properties are commonly considered. One is the granularity of input content, and the other is the interaction function.

The **single-vector representation** is widely used in the fields of biometrics [7,46] and text retrieval [15], and its retrieval process is extremely concise and efficient. While as shown in Figure 2 (a), the over-abstract representation will introduce a lack of fine-grained matching capabilities. Therefore, the Mixtures of Experts (MoE) approaches [26,12,25] take advantage of individual domains to integrate generalizable aggregated features. MMT [12] and CE [26] explicitly construct a multi-expert fusion mechanism to boost retrieval performance. In addition, HIT [25] performs hierarchical cross-modality contrastive matching at both feature-level and semantic-level. They can all be summed up as a hierarchical fusion architecture (Figure 2(b)).

In contrast to the above parameter-free dot-product function, the deeply-contextualized interaction has emerged that fine-tunes MLP models for estimating relevance [41,48] (Figure 2(c)). The recent cross transformer interaction

approaches [23,29] can inference the complex relationship between *arbitrary-length* text and video. However, the neural network interaction lacks the typical inductive bias for matching and usually suffers from optimization difficulty and performance degradation [29]. Furthermore, these heavy interactions will bring a prohibitively computational cost for real-world deployments.

We present a disentangled framework to address the above challenges, where a novel **token-wise interaction** and **channel decorrelation regularization** collaboratively decouples the sequential and hierarchical representation. Concretely, we propose a lightweight token-wise interaction that fully interacts with all sentence tokens and video frame tokens (Figure 2(e-f)). Compared with the single-vector interaction (Figure 2(a)(c)) and multi-level interaction (Figure 2(b)), our method can preserve more fine-grained clues. Compared to the cross transformer interaction (Figure 2(d)), the proposed interaction mechanism significantly ease the optimization difficulty and computational overhead. In addition to the interaction mechanism, we employ a Channel DeCorrelation Regularization (CDCR) to minimize the redundancy between the components of the compared vectors, which facilitate learning a hierarchical representation. The proposed modules are orthogonal to the existing pretraining techniques [23,21,24,28,2] and can be easily implemented by a few lines of code in modern libraries.

We validate the retrieval ability of our approach on multiple text-video retrieval datasets. Our experimental results outperform the state-of-the-art methods under widely used benchmarks, *e.g.*, surpassing CLIP4Clip [29] largely by +2.9%, +3.1%, +7.9%, +2.3%, +2.8% and +6.5% R@1 on the MSR-VTT [44], MSVD [43], VATEX [42], LSMDC [39], AcitivityNet [10], and DiDeMo [1], respectively. Notably, using ViT-B/16 [35] and QB-Norm [3] post processing, our best model can achieve 53.3% T2V R@1 and 56.2% V2T R@1, surpassing all existing single-model entries.

Our empirical analysis suggests that there is vast room for improvement in the design of interaction mechanisms. The findings used in this paper make some initial headway in this direction. We hope that this study will spur further investigation into the operational mechanisms used in modeling interaction.

2 Related Work

Feature Representation for Text-Video Retrieval. In recent studies [26,35,34,29], the text and video inputs are usually considered separately for efficient deployment through a Bi-Encoder architecture [38,18]. The text encoder absorbs progress in the NLP field, and is upgraded from the early Word2Vec [30] to the BERT-like models [8,27]. While for the video input, due to its rich semantic content, researchers utilize multi-modality features [12] and a variety of pre-trained experts [26] to boost performance. Recently, CLIP [35] proposes a concise contrastive learning method and trains on a large-scale dataset with 400 million pairs to obtain the generic cross-modal representation. Indeed, in the span of just a few months, several CLIP-based TVR methods [29,13,11] constantly refresh the state-of-the-art results on all benchmarks. Our approach also benefits

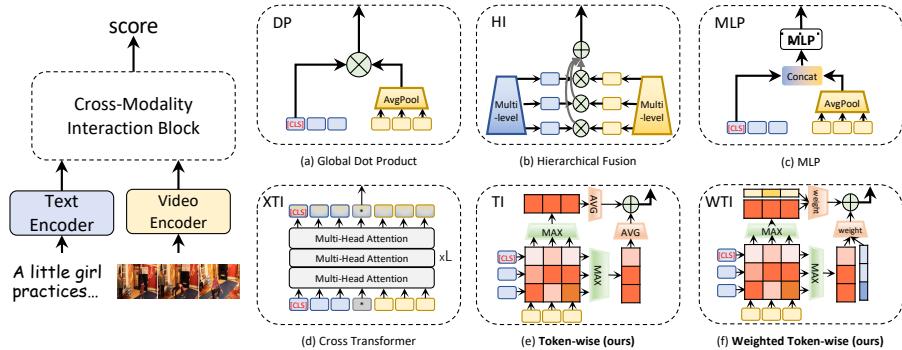


Fig. 2. Illustrations of the text-video retrieval architecture and six categories of interaction methods. Each subplot shows a typical interaction module.

from existing pre-training, while with mainly focusing on the design of the interaction module. We conduct extensive and fair ablation studies to evaluate the effectiveness of our approach.

Interaction Mechanism for Text-Video Retrieval. Over the past few years, most TVR studies have focused on improving performance by increasing the power of text and visual encoders [6,12,25,26]. Thus the simple dot-product interaction is widely adopted for computing global similarity [29]. The time-series video features are aggregated with average pooling, LSTM [14], or temporal transformer [29]. Recent works [6,12,26] on the Mixture of Experts (MoE [16]) show that integrating diverse domain experts can improve the overall model’s capability with a voting procedure. In [25], hierarchical feature alignment methods help models utilize information from different dimensions. The pioneering work, JSFusion [48], proposes to measure dense semantic similarity between all sequence data and learn a sophisticated attentions network. The Cross Transformer [23,28,29] establish the multi-modality correspondence by joint encoding texts and videos, which can capture both intra- and inter-modality context. We conduct the empirical study on the latest instantiation of interaction. Our work is also related to several approaches that analyze the interaction mechanism on Information Retrieval (IR) [18] and Text-Image Retrieval [19]. This work targets a deeper understanding of the interaction mechanism for Text-Video Retrieval in a new perspective.

Contrastive Learning for Text-Video Retrieval. A common underlying theme that unites retrieval methods is that they aim to learn closer representations for the labelled text-video pairs. The triplet loss based models[33,26] use a max-margin approach to separate positive from negative examples and perform intra-batch comparisons. With the popularity of self-supervised learning (SSL) [5,4,31], InfoNCE [31] have dominated recent retrieval tasks, which mainly maximizes the diagonal similarity and achieves superior performance for noisy data. The Barlow Twins method [49] have recently been proposed as a new solution for SSL. They use a covariance matrix to indicate inter-channel

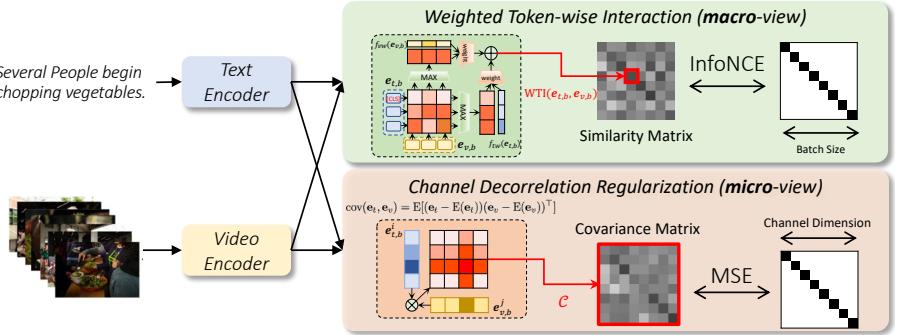


Fig. 3. Overview of the proposed disentangled representation learning design for Text-Video Retrieval. The pipeline of our method consists of two components: the Weighted Token-wise Interaction (WTI) block that exhaustively matching all sequential representation for the sentence and video; the Channel DeCorrelation Regularization (CDCR) minimizes the redundancy for the sequential representation.

redundancy. We employ a sequential channel decorrelation regularization to reduce inter-channel redundancy and competition, which fits the assumptions of hierarchical semantics in Text-Video Retrieval.

3 Methodology

The framework of the proposed Text-Video Retrieval (TVR) is shown in Figure 3. Our method first generates sequential feature representation for both text and video. In the cross-modality interaction stage, the network will reason the dense correlation for all pair-wise features and dynamically adjust weights based on the token contents. We additionally introduce a channel decorrelation regularization to minimize the redundancy of these features. Consequently, given a text query $\mathbf{t} = \{t^i\}_{i=1}^{N_t}$ with N_t tokens, and a set of video documents $\mathcal{V} = \{\mathbf{v}_n\}_{n=1}^N$, the text-video retrieval is formulated as a cross-modality similarity measurement, $\mathcal{S}(\mathbf{t}, \mathbf{v})$. We can extract text features \mathbf{e}_t and video features \mathbf{e}_v through text encoder $E_t(\mathbf{t}, \theta_t)$ and video encoder $E_v(\mathbf{v}, \theta_v)$, respectively. Different from the global dot-product operation with single vector, our embedding features retain the sequential structure of text and video frames with $\mathbf{e}_t \in \mathbb{R}^{N_t \times D}$, $\mathbf{e}_v \in \mathbb{R}^{N_v \times D}$, where N_v is the sampled video frames and D is the feature dimension. We further introduce a Weighted Token-wise Interaction (WTI) module to estimate the cross-modality correlation and discover potential activate tokens. In order to improve the domain generalization, we leverage the covariance matrix to indicate the redundancy of features and employ a simple Channel Decorrelation Regularization (CDCR) to improve the performance. The decorrelation loss implicitly alleviates an overlooked channel competition, achieving impressive performance in retrieval more complex sentences.

Table 1. Comparisons of different interaction mechanisms. N denotes number of video documents (N is large and depends on applications); D denotes representation dimension ($D = 512$, by default); N_t and N_v denote length of text token and frame token, N_{t+v} is the sum of N_t and N_v ($N_t = 32, N_v = 12$); L denotes number of network layer and S denotes feature levels ($L = 4, S = 3$, by default).

Interaction	Contents	Function	Computational Complexity	Memory
DP	single	parameter-free	$\mathcal{O}(ND)$	$\mathcal{O}(ND)$
HI	multi-level	light-parameter	$\mathcal{O}(NSD)$	$\mathcal{O}(NSD)$
MLP	single	black-box	$\mathcal{O}(N(D^2L + D))$	$\mathcal{O}(ND)$
XTI	token-wise	black-box	$\mathcal{O}(N(D^2(N_{t+v}) + N_{t+v}^2D)L)$	$\mathcal{O}(NN_vD)$
TI	token-wise	parameter-free	$\mathcal{O}(NN_tN_vD)$	$\mathcal{O}(NN_vD)$
WTI	token-wise	light-parameter	$\mathcal{O}(N(N_tN_vD + N_{t+v}))$	$\mathcal{O}(NN_v(D + 1))$

3.1 Feature Extractor

We utilize the efficient Bi-Encoder architecture for feature extraction. For each query sentence \mathbf{t} , we add the [CLS] and [SEP] token to indicate the start and the end of the sentence, and adopt the pretrained BERT-BASE [8] to encode the text representation $\mathbf{e}_t = E_t(\mathbf{t})$. For each video \mathbf{v} , we uniformly select N_v frames as keyframes and employ off-the-shelf transformer-based networks, *e.g.* ViT [9], to extract sequential features $\mathbf{e}_v = E_v(\mathbf{v})$.

Retrieval across modalities benefits from large-scale pre-training tasks [18,8,35]. In this paper, we mainly focus on the design of the interaction module rather than the pretrained network. Thus our feature extractors are initialized from the CLIP [35], and we finetune in an end-to-end manner.

3.2 Study of Interaction Mechanisms

To facilitate our study, we develop a generalized interaction formulation that is able to represent various module designs and show a comparison in Table 1. We then show how the existing interaction mechanisms can be represented within this formulation, and how ablations can be conducted using this formulation with respect to different interaction module elements.

Single Vector Dot-product Interaction: When measuring feature similarity, researchers use simple dot-product operation, which is an intuitive solution. Specifically, two global representations are compared in the ℓ_2 normalized embedding space:

$$\text{DP}(\mathbf{e}_t, \mathbf{e}_v) = \frac{(\mathbf{e}_t^{[\text{CLS}]})^\top \cdot \bar{\mathbf{e}}_v}{\|\mathbf{e}_t^{[\text{CLS}]}\|_2 \cdot \|\bar{\mathbf{e}}_v\|_2}, \quad (1)$$

where $\mathbf{e}_t^{[\text{CLS}]}$ is the text feature for [CLS] token and $\bar{\mathbf{e}}_v = \frac{1}{N_v} \sum_{i=1}^{N_v} \mathbf{e}_v^i$ is the average video representation. The dot-product method is widely used in biometrics [7,46], text-image retrieval [47], and text document retrieval [15]. This interaction method is extremely efficient and can be accelerated by the ANN library, *e.g.*, FAISS [17]. While, due to the single-vector representation, the sequential

structure is heavily coupled in encoder. Compressing the long sequence into a single vector will push the network to learn extremely complex abstractions and may easily miss out fine-grained clues. The dot-product function also encourages smooth latent spaces, which will discourage the hierarchical semantics learning. **Hierarchical Interaction:** Similar to the global feature representation, multi-layer [25], multi-modal [12] and multi-experts [26] use the gated fusion mechanism to ensemble the hierarchical representation:

$$\text{HI}(\mathbf{e}_t, \mathbf{e}_v) = \sum_{s=1}^S w(\mathbf{e}_{t,s}^{[\text{CLS}],s})^\top \bar{\mathbf{e}}_{v,s}, \quad (2)$$

where w_s can be a normalized weight or a binarized gated unit. We note that this hierarchical interaction actually contains two essential factors: one is the retrieval-oriented feature pool, and the other is the dynamic fusion block. The content of the comparison has been greatly enriched from expert models. While the learning of the experts requires labor intensive manual annotation.

MLP on Global Vector: In order to improve the nonlinear measurement ability of the interaction block, researchers [41,29] propose to use neural network to learn the metric, among which the representative work [41] builds a Multi-layer Perceptron (MLP) measurement. They directly concatenate the compared features as a whole, and optimize the similarity through MLP:

$$\text{MLP}(\mathbf{e}_t, \mathbf{e}_v) = f_\theta([\mathbf{e}_t^{[\text{CLS}]}, \bar{\mathbf{e}}_v]). \quad (3)$$

where $[,]$ denotes concatenation operation. Although the neural network brings a nonlinear metric space, it also operates in a black-box setting. The pure-parameterizative method needs to consume massive amount of labelled data in the training process. And when deployed on large-scale (billions) documents, even two fully connected layers will bring prohibitively computational overhead. **Cross Transformer Interaction:** The standard self-attention [40] is adopted for cross-modality matching, which can handle variable-length inputs:

$$\text{XTI}(\mathbf{e}_t, \mathbf{e}_v) = \text{MHA}_\theta([\mathbf{e}_t, \mathbf{e}_v]). \quad (4)$$

The Cross Transformer aims to capture and model the inter-modal relations between texts and videos by exchanging key-value pairs in the multi-headed attention (MHA) mechanism. In Section 4.2, we observe that the cross transformer interaction is difficult to optimize and usually occurs performance degradation on video retrieval datasets. Furthermore, compared with the dot-product operation, multi-head attention brings thousand times of the computational overhead, as shown in Table 1 and Table 2.

Token-wise Interaction: Recently, ColBERT [18] and FILIP [45] propose token-wise interaction for document and image retrieval. We introduce this token-wise interaction to TVR, which cleverly solves the local context matching problem (Assumption (1) in Figure 1):

$$\text{TI}(\mathbf{e}_t, \mathbf{e}_v) = \left(\sum_{i=1}^{N_t} \max_{j=1}^{N_v} (\tilde{\mathbf{e}}_t^i)^\top \tilde{\mathbf{e}}_v^j + \sum_{j=1}^{N_v} \max_{i=1}^{N_t} (\tilde{\mathbf{e}}_t^i)^\top \tilde{\mathbf{e}}_v^j \right) / 2, \quad (5)$$

Algorithm 1 PyTorch-style pseudocode for Weighted Token-wise Interaction.

```

# t: text input          v: video input
# f_t: text encoder network   f_v: video encoder network
# f_tw: text weight network   f_vw: video weight network
# B: batch size           D: dimensionality of the embeddings
# N_t: text token size     N_v: frame token size

def weighted_token_wise_interaction(t, v):
    # compute embeddings
    e_t = f_t(t) # BxN_txD
    e_v = f_v(v) # BxN_vxD

    # generate fusion weights
    text_weight = torch.softmax(f_tw(e_t), dim=-1)      # BxN_t
    video_weight = torch.softmax(f_vw(e_v), dim=-1)      # BxN_v

    # normalize representation
    e_t = e_t / e_t.norm(dim=-1, keepdim=True) # BxN_txD
    e_v = e_v / e_v.norm(dim=-1, keepdim=True) # BxN_vxD

    # token interaction
    logits = torch.einsum("atc,bvc->abtv", [e_t, e_v]) # BxBxN_txDxN_v
    t2v_logits = logits.max(dim=-1)[0] # BxBxN_txDxN_v -> BxBxN_t
    t2v_logits = torch.einsum("abt,at->ab", [t2v_logits, text_weight])
# BxBxN_t -> BxB
    v2t_logits = logits.max(dim=-2)[0] # BxBxN_v
    v2t_logits = torch.einsum("abv,bv->ab", [v2t_logits, video_weight])
# BxBxN_v -> BxB

    retrieval_logits = (t2v_logits + v2t_logits) / 2.0 # BxB

    return retrieval_logits

```

where $\tilde{\mathbf{e}} = \mathbf{e}^i / \|\mathbf{e}^i\|_2$ is the channel-wise normalization operation. The token-wise interaction is a parameter-free operation and allows efficient storing and indexing [18].

Weighted Token-wise Interaction: Intuitively, not all words and video frames contribute equally. We provide an adaptive approach to adjust the weight magnitude for each token:

$$\text{WTI}(\mathbf{e}_t, \mathbf{e}_v) = \left(\sum_{i=1}^{N_t} f_{tw,\theta}^i(\mathbf{e}_t) \max_{j=1}^{N_v} (\tilde{\mathbf{e}}_t^i)^\top \mathbf{e}_v^j + \sum_{j=1}^{N_v} f_{vw,\theta}^j(\mathbf{e}_v) \max_{i=1}^{N_t} (\tilde{\mathbf{e}}_t^i)^\top \tilde{\mathbf{e}}_v^j \right) / 2, \quad (6)$$

where $f_{tw,\theta}$ and $f_{vw,\theta}$ are composed of classic MLP and a SoftMax function. The adaptive block is lightweight and takes a **single**-modality input, allowing offline pre-computation for large-scale video documents. In the online process, the attention module introduces negligible computational overhead, as shown in Table 2. Our method inherits the general matching priors and is empirically verified to be more effective and efficient.

WTI can be easily implemented by a few lines of code in modern libraries. Algorithm 1 shows a simplified code based on PyTorch [32].

3.3 Channel Decorrelation Regularization

Given a batch of B video-text pairs, WTI generates an $B \times B$ similarity matrix. The Text-Video Retrieval is trained in a supervised way. We employ the

InfoNCE loss [31] to maximize the similarity between labelled video-text pairs and minimize the similarity for other pairs:

$$\begin{aligned}\mathcal{L}_{\text{InfoNCE}} &= \mathcal{L}_{v2t} + \mathcal{L}_{t2v} \\ \mathcal{L}_{v2t} &= -\frac{1}{B} \sum_i^B \log \frac{\exp(\text{WTI}(\mathbf{e}_{t,i}, \mathbf{e}_{v,i}) / \tau)}{\sum_j^B \exp(\text{WTI}(\mathbf{e}_{t,i}, \mathbf{e}_{v,j}) / \tau)} \\ \mathcal{L}_{t2v} &= -\frac{1}{B} \sum_i^B \log \frac{\exp(\text{WTI}(\mathbf{e}_{t,i}, \mathbf{e}_{v,i}) / \tau)}{\sum_j^B \exp(\text{WTI}(\mathbf{e}_{t,j}, \mathbf{e}_{v,i}) / \tau)},\end{aligned}\quad (7)$$

where τ is temperature hyper-parameter.

The contrastive loss provides a *macro* objective to optimize the global similarity. Referring expression comprehension, the multi-modality retrieval often requires semantic information from *micro*-views, *e.g.*, the channel-level. Inspired from self-supervised learning methods [49], we utilize the covariance matrix to measure the redundancy between features and employ a simple ℓ_2 -norm minimization to optimize the hierarchical representation:

$$\begin{aligned}\mathcal{L}_{\text{CDCR}} &= \sum_i (1 - \mathcal{C}^{ii})^2 + \alpha \sum_i \sum_{j \neq i} (\mathcal{C}^{ij})^2 \\ \mathcal{C}^{ij} &\triangleq \frac{\sum_b \mathbf{e}_{t,b}^{(i)} \mathbf{e}_{v,b}^{(j)}}{\sqrt{\sum_b (\mathbf{e}_{t,b}^{(i)})^2} \sqrt{\sum_b (\mathbf{e}_{v,b}^{(j)})^2}},\end{aligned}\quad (8)$$

where $\mathbf{e}_{t,b}^{(i)}$ is the i -th channel of b -th text feature $\mathbf{e}_{t,b}$. The coefficient α controls the magnitude of the redundancy term. The total training loss \mathcal{L}_{all} is defined as:

$$\mathcal{L}_{\text{all}} = \mathcal{L}_{\text{InfoNCE}} + \lambda \mathcal{L}_{\text{CDCR}}, \quad (9)$$

where λ is the weighting parameter. Surprisingly, empirical results show that the cross-modality channel decorrelation regularization brings significant improvements for all interaction mechanisms, as shown in Table 2.

4 Experiments

In this section, we firstly present the configuration details of our algorithm and then conduct thorough ablation experiments to explore the relative importance of each component in our method on MSR-VTT [44]. Finally we conduct comprehensive experiments on six benchmarks: MSR-VTT [44], MSVD [43], VATEX [42], LSMDC [39], ActivityNet [10] and DiDeMo [1].

4.1 Experimental Settings

Dataset: We conduct experiments on six benchmarks for video-text retrieval tasks including:

- **MSR-VTT** [44] contains 10,000 videos with 20 captions for each. We report results on 1k-A which adopts 9,000 videos with all corresponding captions for training and utilizes 1,000 video-text pairs as test.
- **MSVD** [43] includes 1,970 videos with 80,000 captions. We report results on split set, where train, validation and test are 1200, 100 and 670 videos.
- **VATEX** [42] contains 34,991 videos with multilingual annotations. The training split contains 25,991 videos. We report the results on the split set includes 1500 videos for validation and 1500 videos for test.
- **LSMDC** [39] contains 118081 videos and equal captions extracted from 202 movies with a split of 109673, 7408, and 1000 as the train, validation, and test set. Every video is selected from movies ranging from 2 to 30 seconds.
- **ActivityNet** [10] consists of 20,000 YouTube videos. We concatenate all descriptions of a video to a single query and evaluate on the ‘val1’ split.
- **DiDeMo** [1] contains 10,000 videos annotated with 40,000 sentences. All sentence descriptions for a video are also concatenated into a single query for text-video retrieval.

Evaluation Metric: We follow the standard retrieval task [29] and adopt Recall at rank K (R@K), median rank (MdR) and mean rank (MnR) as metrics. Higher R@K and lower MdR or MnR indicates better performance.

Implementation Details: We utilize the standard Bi-Encoder from CLIP [35] as pre-trained feature extractor. Concretely, the vision encoder is comprised of a vanilla ViT-B/32 [9] and 4-layers of temporal transformer blocks. Each block employs 8 heads and 512 hidden channels. The temporal position embedding and network weight parameters are initialized from the CLIP’s text encoder. The fixed video length and caption lengths are 12 and 32 for MSR-VTT, MSVD, VATEX, LSMDC and 64 and 64 for ActivityNet and DiDeMo. Following [35], the special tokens, [CLS] and [SEP], and the text tokens are concatenated as inputs to a 12-layer linguistic transformer [8,35]. We leverage the MLP to capture the weight factor for tokens, and our ablation studies suggest a 2-layer structure. The adaptive module is initialized from scratch with random weights.

Training Schedule: For fair comparisons with our baseline, we follow training schedules from CLIP4Clip [29]. The network is optimized by Adam [20] with a batch size of 128 in 5 epochs. The initial learning rate for vision encoder and text encoder are set 1×10^{-7} , and the initial learning rate for the temporal transformer and the adaptive module are set to 1×10^{-4} . All learning rates follow the cosine learning rate schedule with a linear warmup. We apply a weight decay regularization to all parameters except for bias, layer normalization, token embedding, positional embedding and temperature. During training, we set the temperature $\tau = 100$, CDCR weight $\alpha = 0.06$ and overall weighting parameter $\lambda = 0.001$.

4.2 Ablation Study

We quantitatively evaluate the key components, Weighted Token-wise Interaction (WTI) and Channel DeCorrelation Regularization (CDCR), on MSR-VTT 1K [44]. Table 2 summarizes the ablation results.

Table 2. Ablation of Disentangled Representation on the 1K validation set of MSR-VTT [44]. Time shows inference speed for indexing 1 million video documents on a Tesla V100 GPU.

Interaction	InfoNCE [31]				+CDCL				Time (ms)
	R@1↑	R@5↑	R@10↑	MnR↓	R@1↑	R@5↑	R@10↑	MnR↓	
DP	42.8	72.1	81.4	16.3	44.2	72.7	82.0	14.5	415
HI	43.5	72.9	81.7	16.1	44.1	72.6	82.8	14.1	531
MLP	29.3	54.8	64.2	33.8	-	-	-	-	25,304
XTI	41.8	71.2	82.7	16.2	-	-	-	-	80,453
TI	44.8	73.7	82.9	13.5	45.5	72.0	82.5	13.3	536
WTI	46.3	73.7	83.2	13.0	47.4	74.6	83.8	12.8	565
DP _{+ViT-B/16}	45.9	73.8	82.3	13.8	46.6	73.3	82.8	13.4	415
TI_{+ViT-B/16}	47.3	76.7	84.8	13.7	49.1	75.7	85.1	12.7	536
WTI_{+ViT-B/16}	48.8	76.1	84.3	13.5	50.2	76.5	84.7	12.4	565

Table 3. Effect of dual-path and layers for weight model on MSR-VTT 1K [44].

Dual	t2v				v2t				t2v+v2t			
	R@1↑	R@5↑	R@10↑	MnR↓	R@1↑	R@5↑	R@10↑	MnR↓	R@1↑	R@5↑	R@10↑	MnR↓
TI	43.4	71.0	80.2	16.1	43.5	71.8	82.7	14.0	44.8	73.7	82.9	13.5
WTI	45.4	72.3	81.7	13.4	45.3	74.6	83.3	13.6	46.3	73.7	83.2	13.0
Layers	1FC				2FC				3FC			
WTI	46.3	73.9	82.9	13.7	46.3	73.7	83.2	13.0	45.7	72.9	81.4	14.0

Effect of Weighted Token-wise Interaction: Compared with the Single-Vector Dot-Product Interaction (DP), the Token-wise Interaction (TI) only adds a dense correlation, while the performance boosts +2.0% R@1. Our Weighted Token-wise Interaction (WTI) dramatically improves the performance by an R@1 of 3.5%, which shows that the proposed interaction structure indeed helps the model to adequately exploit pair-wise correlations. Compared with other light-parameter function, our method improves by +2.8% R@1 over Hierarchical Interaction (HI). An explanation is that the token representation is more valuable than the multi-level features for the single-scale ViT [9]. As observed in [29], the MLP interaction (MLP) and Cross Transformer Interaction (XTI), under black-box function, would lead to degradation problems. Further, by adopting ViT-B/16 [35], WTI greatly promotes the T2V R@1 to 48.8%.

Effect of Channel DeCorrelation Regularization: In Table 2, we evaluate the importance of Channel Decorrelation Regularization on different interaction structures except block-box function. Surprisingly, CDR increases performance significantly for all interactions, *i.e.* by +1.4%, +0.6%, +0.7% and +1.1% for DP, HI, TI and WTI respectively. The micro channel-level regularization makes it easier to leverage semantic hierarchy.

Effect of Dual-path Token-wise Retrieval: In Table 3, we test the importance of dual-path interaction of t2v and v2t, subset of Eq. 5. By adopting dual-path to compute the logits, our model provides stable gains by +1.4% and +1.0% for TI and WTI, respectively.

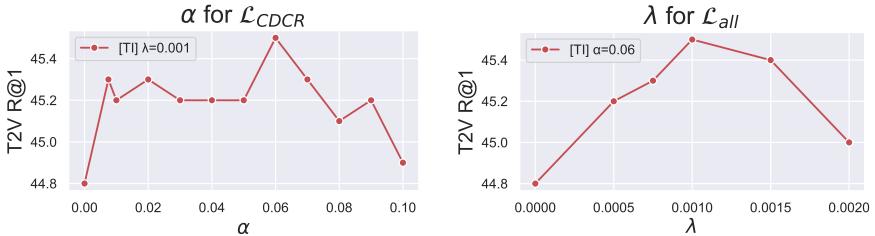


Fig. 4. Hyper-parameter for feature decorrelation. (a) α for \mathcal{L}_{CDCR} (b) λ for \mathcal{L}_{all} .

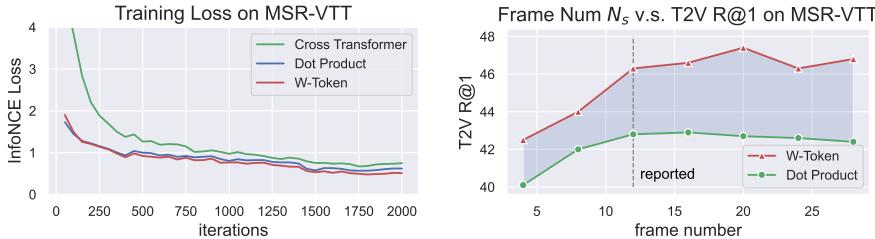


Fig. 5. The training loss comparisons (a) and the influence of frame number (b).

Effect of MLP layers: By default, we simply set the weighted function with MLP, which is composed of FC+ReLU blocks. In Table 3, we observe that R@1 is slightly better when applying 2FC layers and drop by 3FC, likely because a heavier structure suffers from over-fitting.

Hyper-parameter Selection for DeCorrelation: The parameter α and λ specifies the scale and importance of the Channel DeCorrelation Loss. We evaluate the scale range setting $\alpha \in [0, 0.1]$ and $\lambda \in [0, 0.002]$ as shown in Figure 4. We find that R@1 is improved from 44.8% to 45.5% when $\alpha = 0.06$ and saturated with $\alpha = 0.07$. We can get the best R@1 when the λ is around 0.001. As a result, we adopt $\alpha = 0.06$ and $\lambda = 0.001$ in our model to achieve the best performance.

Training Convergence Analysis: To further reveal the source of the performance improvement, we present the training loss in Figure 5. The training of pure-parameterizable interaction (Cross Transformer) is not trivial since it lacks the basic inductive biases for distance measurement. Inspired by this observation, we seek a light-weight way to leverage the heuristic property for better convergence, and therefore achieve a faster convergence rate.

Benefit for More Frames: Figure 5(b) shows performance for DP and WTI with different frames. Compared with the naïve baseline DP, our WTI can achieve more significant gains, +4.9% v.s. +2.8%, with more frames. For fair comparisons with CLIP4Clip [29], we mainly report results at 12 frames.

Inference Speed: Besides recall metric, we report the inference speed of all the interaction structure in Table 2. Our lightweight interaction introduces negligible computational overhead, bringing remarkable improvements. Our approach reduce the inference time by several orders of magnitude, compared with the heavy cross transformer.

Table 4. Retrieval results on the validation set of MSR-VTT 1K [44].

Method	Text→ Video				Video →Text			
	R@1↑	R@5↑	R@10↑	MdR↓	R@1↑	R@5↑	R@10↑	MdR↓
HERO [23]	16.8	43.4	57.7	-	-	-	-	-
UniVL [28]	21.2	49.6	63.1	6.0	-	-	-	-
ClipBERT [21]	22.0	46.8	59.9	6.0	-	-	-	-
MDMMT [12]	26.6	57.1	69.6	4.0	27.0	57.5	69.7	3.7
SUPPORT [33]	27.4	56.3	67.7	3.0	26.6	55.1	67.5	3.0
FROZEN [2]	31.0	59.5	70.5	3.0	-	-	-	-
CLIP4Clip [29]	44.5	71.4	81.6	2.0	42.7	70.9	80.6	2.0
Ours	47.4	74.6	83.8	2.0	45.3	73.9	83.3	2.0
+ViT-B/16	50.2	76.5	84.7	1.0	48.9	76.3	85.4	2.0
+QB-Norm [3]	53.3	80.3	87.6	1.0	56.2	79.9	87.4	1.0

4.3 Comparison with State-of-the-Arts

In this subsection, we compare the proposed model with recent state-of-the-art methods on the six benchmarks, MSR-VTT [44], MSVD [43], VATEX [42], LSMDC [39], ActivityNet [10] and DiDeMo [1].

For **MSR-VTT** in Table 4, our model significantly surpasses the ClipBERT [21] by absolute 25.5% R@1, reaching 47.4% R@1 on MSR-VTT, indicating the benefits and necessity of large scale image-text pre-training for video-text retrieval. We achieve 2.9% improvement compared to CLIP4Clip [29], which shows the benefit from token-wise interaction. Our WTI, employing ViT-B/16 and QB-Norm [3], yields a remarkable T2V R@1 53.3%.

Table 5 shows results for other benchmarks. For **MSVD**, our model also demonstrates competitive performance with top-performing CLIP based model [29]. For **VATEX**, our approach achieves +7.6% R@1 improvement for text-video retrieval and +3.8% improvement for video-text retrieval. For **LSMDC**, our approach achieves +2.3% R@1 improvements for text-video retrieval.

For **ActivityNet**, we outperform the state-of-the-art method by a large margin of +2.8% R@1 on text-video retrieval. For **DiDeMo**, we achieve remarkable performance 47.9% R@1 and a relative performance improvement of 16.5% in T2V compared to CLIP4Clip. It is worth noting that long sentences are used in ActivityNet and DiDeMo. The significant improvements on both datasets further proves the advantages of our disentangled representation.

Overall, the consistent improvements across different benchmarks strongly demonstrate the effectiveness of our algorithm. We hope that our studies will spur further investigation in modeling interaction.

5 Conclusion

In this work, we present an empirical study to give a better general understanding of interaction mechanisms for Text-Video Retrieval. Then we propose a novel

Table 5. Retrieval results on the validation set of MSVD [43], VATEX [42], LSMDC [39], ActivityNet [10] and DiDeMo [1].

Method	Text→ Video				Video →Text			
	R@1↑	R@5↑	R@10↑	MdR↓	R@1↑	R@5↑	R@10↑	MdR↓
	Retrieval performance on MSVD [43]							
CE [26]	19.8	49.0	63.8	6.0	-	-	-	-
SUPPORT [33]	28.4	60.0	72.9	4.0	-	-	-	-
FROZEN[2]	33.7	64.7	76.3	3.0	-	-	-	-
CLIP [35]	37.0	64.1	73.8	3.0	54.9	82.9	89.6	1.0
CLIP4Clip[29]	45.2	75.5	84.3	2.0	48.4	70.3	77.2	2.0
Ours	48.3	79.1	87.3	2.0	62.3	86.3	92.2	1.0
Ours_{+ViT-B/16}	50.0	81.5	89.5	2.0	68.7	92.5	95.6	1.0
Retrieval performance on VATEX [42]								
CLIP [35]	39.7	72.3	82.2	2.0	52.7	88.8	94.9	1.0
SUPPORT [33]	44.9	82.1	89.7	1.0	58.4	84.4	91.0	1.0
CLIP4Clip [29]	55.9	89.2	95.0	1.0	73.2	97.1	99.1	1.0
Ours	63.5	91.7	96.5	1.0	77.0	98.0	99.4	1.0
Ours_{+ViT-B/16}	65.7	92.6	96.7	1.0	80.1	98.5	99.5	1.0
Retrieval performance on LSMDC [39]								
CE [26]	11.2	26.9	34.8	25.3	-	-	-	-
MMT [12]	12.9	29.9	40.1	19.3	-	-	-	-
CLIP [35]	11.3	22.7	29.2	46.5	-	-	-	-
MDMMT [12]	18.8	38.5	47.9	12.3	-	-	-	-
CLIP4Clip [29]	22.6	41.0	49.1	11.0	-	-	-	-
Ours	24.9	45.7	55.3	7.0	24.9	44.1	53.8	9.0
Ours_{+ViT-B/16}	26.5	47.6	56.8	7.0	27.0	45.7	55.4	8.0
Retrieval performance on ActivityNet [10]								
CE [26]	17.7	46.6	-	6.0	-	-	-	-
MMT [12]	28.9	61.1	-	4.0	-	-	-	-
SUPPORT[33]	28.7	60.8	-	2.0	-	-	-	-
CLIP4Clip [29]	41.4	73.7	85.3	2.0	-	-	-	-
Ours	44.2	74.5	86.1	2.0	42.2	74.0	86.2	2.0
Ours_{+ViT-B/16}	46.2	77.3	88.2	2.0	45.7	76.5	87.8	2.0
Retrieval performance on DiDeMo [1]								
CE [26]	15.6	40.9	-	8.2	27.2	51.7	62.6	5.0
ClipBERT [21]	21.1	47.3	61.1	6.3	-	-	-	-
FROZEN [2]	31.0	59.8	72.4	3.0	-	-	-	-
CLIP4Clip [29]	41.4	68.2	79.1	2.0	42.8	69.8	79.0	2.0
Ours	47.9	73.8	82.7	2.0	45.4	72.6	82.1	2.0
Ours_{+ViT-B/16}	49.0	76.5	84.5	2.0	49.9	75.4	83.3	2.0

disentangled representation method, which are collaboratively implemented by a Weighted Token-wise Interaction (WTI) to solve sequential matching problem from macro-view and a Channel DeCorrelation Regularization (CDCR) that reduces feature redundancy from a micro-view. We demonstrate the effectiveness of the proposed approach for modeling better sequential and hierarchical clues on six datasets. We wish our cross-modality interaction will inspire more theoretical researches towards more powerful interaction design.

Appendix

This document brings additional details of Weighted Token-wise Interaction. Additional qualitative and quantitative results are also given for completeness. Figure 6 shows the internal operating mechanism of WTI, and also powerfully explains the performance improvement of our algorithm.

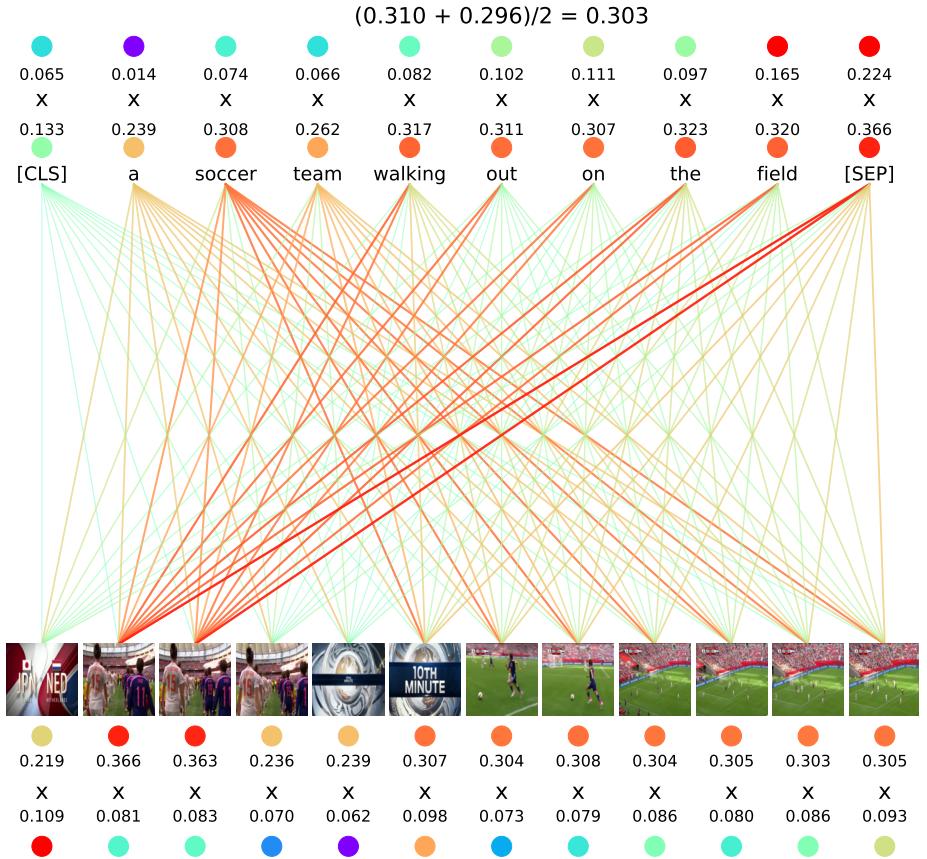


Fig. 6. Visualization of the internal operating mechanism of WTI on MSR-VTT [44].

6 Discussion

How Weighted Token-wise Interaction (WTI) improve performance?
In this section, we discuss the reasons for considering **token-wise** interaction and the **adaptive weight** module.

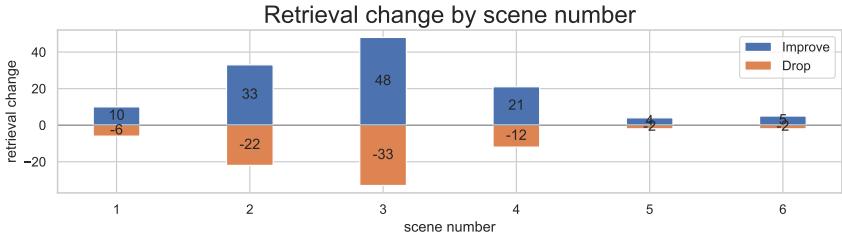


Fig. 7. The improvement with different scenes number on the MSR-VTT 1K val [44].



Fig. 8. The word cloud of top-1 weighted ordinary tokens on the MSR-VTT 1K val [44].

First, the text and video input are naturally sequential, and keeping the sequential structure can reduce the information loss and preserve more fine-grained cues. Using the sequential representation for interaction is essentially an ensemble learning method. The **Max** operation in our token-wise interaction can select matching video clips, as shown in Fig. 6, which can help us resolve **local context assumptions** in the main paper. Otherwise, if using average pooling [29], the overall similarity will be pulled down by irrelevant segments. Quantitatively, we analyze the relationship between recall improvement and the number of scenes in a video (Fig. 7), and find that our algorithm achieves most improvements at 3 scenes. Therefore, token-wise interaction is intuitively a sensible practice.

Second, learning of adaptive weights is necessary, especially when CLIP [35] is used as a pre-trained network. CLIP performs large-scale training (on 256 GPUs for 2 weeks) with large-scale dataset (400 millions), and we cannot completely change the distribution of position embedding and features by a small-scale fine-tuning dataset. The default global token, [SEP], shows a dominant role, achieving top1 important token for 98.7% videos and an average weight of

0.2838, far exceeding the average weight of other tokens (0.0674) on MSR-VTT 1K validation set. Simply averaging all sequential features is unfair to [SEP], resulting in a suboptimal solution. On the other hand, the importance of words in the text are also quite different, and our experimental analysis shows that nouns/verbs account for the main weight, as shown in Fig. 8.

Therefore, a good practice is to consider the token-wise interaction and adaptive weight module and devising a solution to effectively fuse these two factors (as WTI).

7 More Implementation Details

In this section, we present more implementation details that were omitted in the main paper for brevity.

7.1 Pseudocode with Mask Inputs

Please note we omit the **mask** inputs at the main paper for brevity in Section 3.2. Here we present complete pseudo-codes for our Weighted Token-wise Interaction (WTI) and Channel Decorrelation Regularization (CDCR) in Algorithm 2- 4. We believe the pseudocode would aid an independent researcher to better replicate the proposed interaction.

7.2 Network Architecture for Weight Branch

Figure 6 shows details of the network architectures for learning weights. We use the single-modal input for each branch to avoid coupling between modalities.

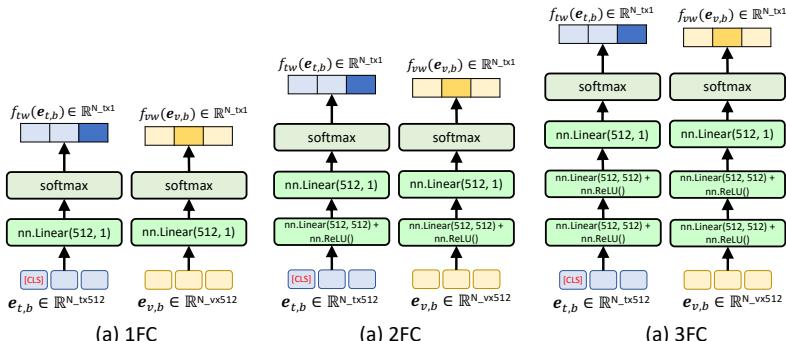


Fig. 9. Network architectures for weight branch.

Algorithm 2 PyTorch-style pseudocode for Weighted Token-wise Interaction.

```

# t: text input           v: video input
# mask_t: text mask      mask_v: video mask
# f_t: text encoder network f_v: video encoder network
# f_tw: text weight network f_vw: video weight network
# B: batch size          D: dimensionality of the embeddings
# N_t: text token size   N_v: frame token size

def weighted_token_wise_interaction(t, mask_t, v, mask_v):
    # compute embeddings
    e_t = f_t(t) # BxN_txD
    e_v = f_v(v) # BxN_vxD

    # generate fusion weights
    text_weight = f_tw(e_t).squeeze() # BxN_t
    # fill masked text with -inf
    text_weight.masked_fill_(1 - mask_t, float("-inf")) # BxN_v
    text_weight = torch.softmax(text_weight, dim=-1) # BxN_t

    video_weight = f_tv(e_v).squeeze() # BxN_v
    # fill masked video with -inf
    vision_weight.masked_fill_(1 - mask_v, float("-inf")) # BxN_v
    video_weight = torch.softmax(vision_weight, dim=-1) # BxN_v

    # normalize representation
    e_t = e_t / e_t.norm(dim=-1, keepdim=True) # BxN_txD
    e_v = e_v / e_v.norm(dim=-1, keepdim=True) # BxN_vxD

    # token interaction
    logits = torch.einsum("atc,bvc->abtv", [e_t, e_v]) # BxBxN_txDxN_v

    # mask for logits
    logits = torch.einsum('abtv,at->abtv', [logits, mask_t])
    logits = torch.einsum('abtv,bv->abtv', [logits, mask_v])

    t2v_logits, t2v_max_idx = logits.max(dim=-1) # BxBxN_txDxN_v -> BxBxN_t
    t2v_logits = torch.einsum("abt,at->ab", [t2v_logits, text_weight])
# BxBxN_t -> BxB
    v2t_logits, v2t_max_idx = logits.max(dim=-2) # BxBxN_txDxN_v -> BxBxN_v
    v2t_logits = torch.einsum("abv,bv->ab", [v2t_logits, video_weight])
# BxBxN_v -> BxB

    retrieval_logits = (t2v_logits + v2t_logits) / 2.0 # BxB

    return retrieval_logits

```

7.3 Complexity Analysis

Single Vector Dot-product Interaction: For each video, we only need to store a D -dimensional feature, and the overall feature storage complexity is $\mathcal{O}(ND)$. During the query process, a similarity calculation consists of D multiplication operations. The overall computational complexity is $\mathcal{O}(ND)$.

Hierarchical Interaction: For the hierarchical interaction, we consider the simplest case, directly using the constant weighted average to fuse the multi-layer features. Therefore, both computation and storage are increased by a factor of S compared to DP.

MLP on Global Vector: We adopt the structure of stacking $L \times$ FC+ReLU blocks, where the input channel of the first layer is $2D$, the hidden channels size is D , and the final output is a single similarity. The computational complexity is $\mathcal{O}(N(2DD + DD(L-2) + D)) = \mathcal{O}(N(LD^2 + D))$. We can observe that with the default configuration ($D = 512, L = 4$), the computational complexity of MLP

Algorithm 3 PyTorch-style pseudocode for Channel DeCorrelation Regularization with single-vector representation.

```
# e_t: text feature           e_v: video feature
# B: batch size              D: dimensionality of the embeddings
# alpha: the magnitude of the redundancy term

def channel_decorrelation_regularization_single(e_t, e_v):
    # batch norm
    t_norm = (e_t - e_t.mean(0)) / e_t.std(0)  # NxD
    v_norm = (e_v - e_v.mean(0)) / e_v.std(0)  # NxD

    B, D = t_norm.shape
    cov = torch.einsum('ac,ad->cd', t_norm, v_norm) / B  # DxD
    # loss
    on_diag = torch.diagonal(cov).add_(-1).pow_(2).sum()
    off_diag = cov.flatten()[1:].view(D - 1, D + 1)[:, :-1].pow_(2).sum()
    cdcr = on_diag + off_diag * alpha
    return cdcr
```

Table 6. Comparisons of different interaction mechanisms. N denotes number of video documents (N is large and depends on applications); D denotes representation dimension ($D = 512$, by default); N_t and N_v denote length of text token and frame token, N_{t+v} is the sum of N_t and N_v ($N_t = 32, N_v = 12$); L denotes number of network layer and S denotes feature levels ($L = 4, S = 3$, by default).

Interaction	Contents	Function	Computational Complexity	Memory
DP	single	parameter-free	$\mathcal{O}(ND)$	$\mathcal{O}(ND)$
HI	multi-level	light-parameter	$\mathcal{O}(NSD)$	$\mathcal{O}(NSD)$
MLP	single	black-box	$\mathcal{O}(N(D^2L + D))$	$\mathcal{O}(ND)$
XTI	token-wise	black-box	$\mathcal{O}(N(D^2N_{t+v} + N_{t+v}^2D)L)$	$\mathcal{O}(NN_vD)$
TI	token-wise	parameter-free	$\mathcal{O}(NN_tN_vD)$	$\mathcal{O}(NN_vD)$
WTI	token-wise	light-parameter	$\mathcal{O}(N(N_tN_vD + N_{t+v}))$	$\mathcal{O}(NN_v(D + 1))$

is about **2,049**× of the DP, which is why neural network-based interactions are difficult to deploy. They are usually designed to be used as a fine-grained ranking module. The storage complexity of MLP is same as DP.

Cross Transformer Interaction: It is usually assumed that a Multi-Head Attention block (MHA) consists of linear layer and **QKV** structure. The length of the sequence input is N_{t+v} , so the computational complexity of the linear layer is $\mathcal{O}(D^2N_{t+v})$. An dot-product version of the QKV operation requires $\mathcal{O}(N_{t+v}^2D)$ complexity. Therefore, the overall computational complexity for $L \times$ MHA is $\mathcal{O}(N(D^2N_{t+v} + N_{t+v}^2D)L)$. With the default configuration ($D = 512, L = 4, N_t = 32, N_v = 12, N_{t+v} = 44$), the computational complexity of XTI is about **24,464**× of the DP. At the same time, due to the use of sequential representation for video, the storage complexity rises to $\mathcal{O}(NN_vD)$.

Token-wise Interaction: Token-wise interactions need to calculate an $N_t \times N_v$ size of similarity matrix for text and video tokens. The computational complexity is $\mathcal{O}(N(N_tN_vD))$ and the storage complexity is $\mathcal{O}(NN_vD)$.

Weighted Token-wise Interaction: For weighted token-wise interaction, the complexity for the online search process is a little more complicated. For the

Algorithm 4 PyTorch-style pseudocode for Channel DeCorrelation Regularization with sequential representation.

```

# e_t: text feature           e_v: video feature
# mask_t: text mask          mask_v: video mask
# B: batch size               D: dimensionality of the embeddings
# N_t: text token size        N_v: frame token size

def channel_decorrelation_regularization_sequential(e_t, mask_t, e_v, mask_v):
    # select max indexes for each text-video pair
    i4t = t2v_max_idx[torch.arange(B), torch.arange(B)] # BxBxN_t -> BxN_t
    i4v = v2t_max_idx[torch.arange(B), torch.arange(B)] # BxBxN_v -> BxN_v

    e_4_t = e_v[torch.arange(B).repeat_interleave(N_t), i4t.flatten()]
    # (BxN_t)xD
    e_4_v = e_t[torch.arange(B).repeat_interleave(N_v), i4v.flatten()]
    # (BxN_v)xD
    e_t = e_t.reshape(-1, D) # (BxN_t)xD
    e_v = e_v.reshape(-1, D) # (BxN_v)xD
    mask_t = mask_t.flatten().type(torch.bool)
    mask_v = mask_v.flatten().type(torch.bool)

    e_t = e_t[mask_t]
    e_v = e_v[mask_v]
    e_4_t = e_4_t[mask_t]
    e_4_v = e_4_v[mask_v]

    # cov for t2v
    t_norm = (e_t - e_t.mean(0)) / e_t.std(0) # XxD
    v_norm = (e_4_t - e_4_t.mean(0)) / e_4_t.std(0) # XxD
    X = t_norm.shape
    cov1 = torch.einsum('ac,ad->cd', t_norm, v_norm) / B # DxD

    # cov for v2t
    v_norm = (e_t - e_t.mean(0)) / e_t.std(0) # XxD
    t_norm = (e_4_v - e_4_v.mean(0)) / e_4_v.std(0) # XxD
    X = t_norm.shape[0]
    cov2 = torch.einsum('ac,ad->cd', t_norm, v_norm) / B # DxD

    cov = (cov1+cov2)/2

    # loss
    on_diag = torch.diagonal(cov).add_(-1).pow_(2).sum()
    off_diag = cov.flatten()[1:].view(D - 1, D + 1)[:, :-1].pow_(2).sum()
    cdcr = on_diag + off_diag * alpha
    return cdcr

```

query sentence, we need to calculate an additional MLP to generate the text weights, which brings $\mathcal{O}(D^2LN_s)$. For video documents, we can **pre-compute** sequential video features and the corresponding video weights. The total storage overhead is $\mathcal{O}(N(N_vD + N_v))$. Note that **during the online process, there is no need to dynamically calculate the weights for each video**. Compared with TI, only two additional weighting operations are added in the online process. Therefore, the overall computational complexity is $\mathcal{O}(N(N_tN_vD + N_{t+v}))$. Compared to XTI, our computational complexity is reduced **63.69** times.

8 Visualizations

More visualization results of WTI are shown in Fig. 10 and 11.

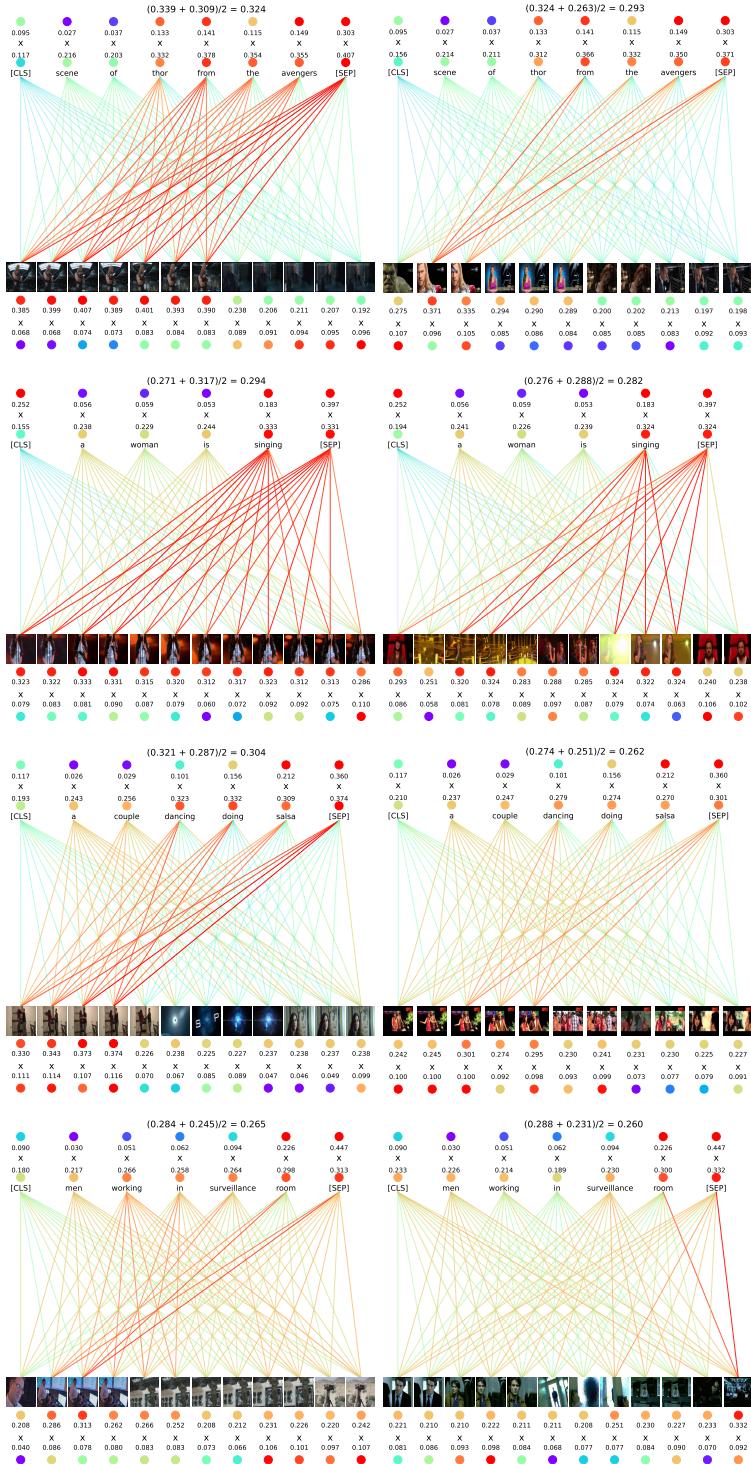


Fig. 10. Qualitative results on images from MSR-VTT [44].

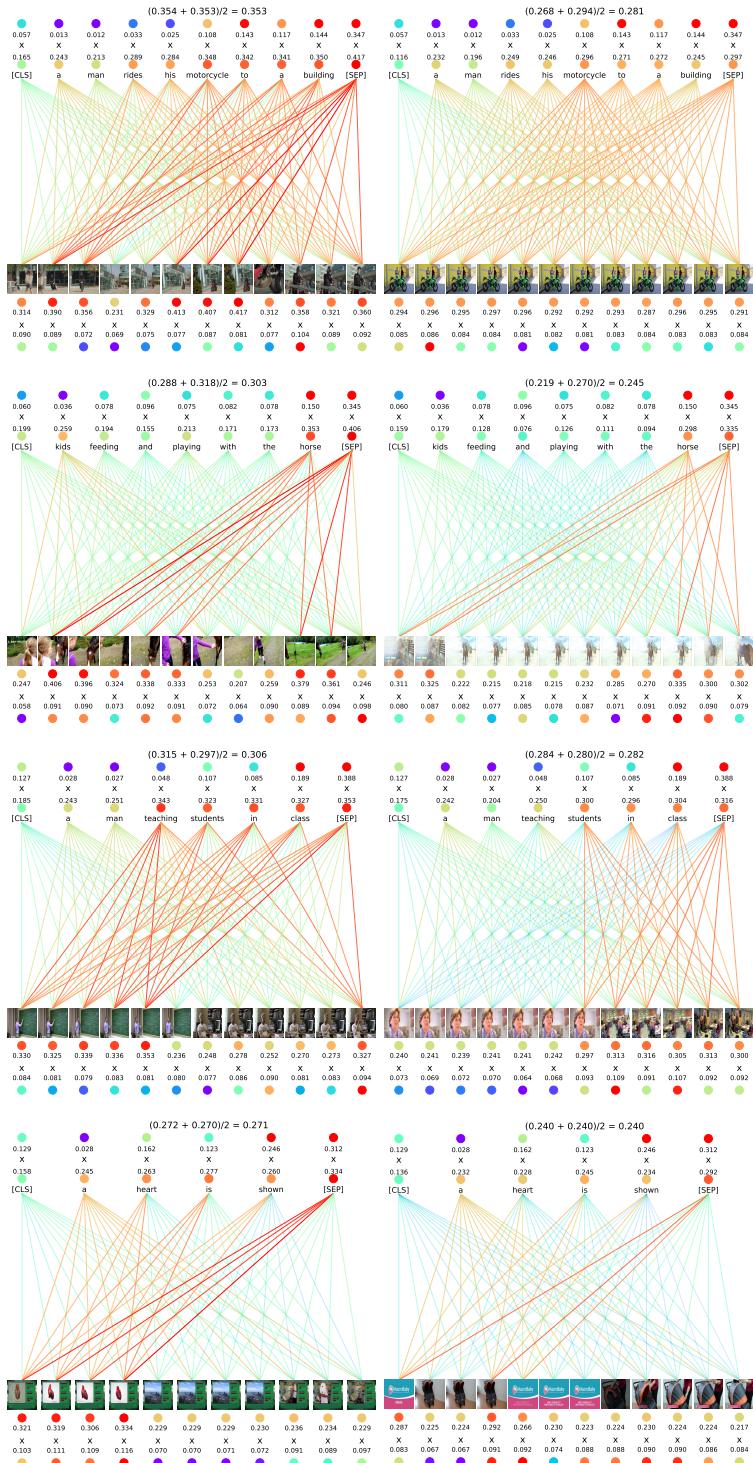


Fig. 11. Qualitative results on images from MSR-VTT [44].

References

1. Anne Hendricks, L., Wang, O., Shechtman, E., Sivic, J., Darrell, T., Russell, B.: Localizing moments in video with natural language. In: Proceedings of the IEEE international conference on computer vision. pp. 5803–5812 (2017)
2. Bain, M., Nagrani, A., Varol, G., Zisserman, A.: 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. pp. 1728–1738 (2021)
3. Bogolin, S.V., Croitoru, I., Jin, H., Liu, Y., Albanie, S.: Cross modal retrieval with querybank normalisation. arXiv preprint arXiv:2112.12777 (2021)
4. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
5. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15750–15758 (2021)
6. Croitoru, I., Bogolin, S.V., Leordeanu, M., Jin, H., Zisserman, A., Albanie, S., Liu, Y.: Teachtext: Crossmodal generalized distillation for text-video retrieval. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11583–11593 (2021)
7. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4690–4699 (2019)
8. 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)
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
10. Fabian Caba Heilbron, Victor Escorcia, B.G., Niebles, J.C.: Activitynet: A large-scale video benchmark for human activity understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 961–970 (2015)
11. Fang, H., Xiong, P., Xu, L., Chen, Y.: Clip2video: Mastering video-text retrieval via image clip. arXiv preprint arXiv:2106.11097 (2021)
12. Gabeur, V., Sun, C., Alahari, K., Schmid, C.: Multi-modal transformer for video retrieval. In: European Conference on Computer Vision. pp. 214–229. Springer (2020)
13. Gao, Z., Liu, J., Chen, S., Chang, D., Zhang, H., Yuan, J.: Clip2tv: An empirical study on transformer-based methods for video-text retrieval. arXiv preprint arXiv:2111.05610 (2021)
14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
15. Huang, P.S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. pp. 2333–2338 (2013)
16. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural computation **3**(1), 79–87 (1991)

17. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* **7**(3), 535–547 (2019)
18. Khattab, O., Zaharia, M.: Colbert: Efficient and effective passage search via contextualized late interaction over bert. In: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. pp. 39–48 (2020)
19. Kim, W., Son, B., Kim, I.: Vilt: Vision-and-language transformer without convolution or region supervision. In: International Conference on Machine Learning. pp. 5583–5594. PMLR (2021)
20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
21. 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: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7331–7341 (2021)
22. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461 (2019)
23. 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)
24. 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: European Conference on Computer Vision. pp. 121–137. Springer (2020)
25. Liu, S., Fan, H., Qian, S., Chen, Y., Ding, W., Wang, Z.: Hit: Hierarchical transformer with momentum contrast for video-text retrieval. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11915–11925 (2021)
26. Liu, Y., Albanie, S., Nagrani, A., Zisserman, A.: Use what you have: Video retrieval using representations from collaborative experts. arXiv preprint arXiv:1907.13487 (2019)
27. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
28. 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)
29. Luo, H., Ji, L., Zhong, M., Chen, Y., Lei, W., Duan, N., Li, T.: Clip4clip: An empirical study of clip for end to end video clip retrieval. arXiv preprint arXiv:2104.08860 (2021)
30. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
31. Van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv e-prints pp. arXiv–1807 (2018)
32. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
33. Patrick, M., Huang, P.Y., Asano, Y., Metze, F., Hauptmann, A., Henriques, J., Vedaldi, A.: Support-set bottlenecks for video-text representation learning. arXiv preprint arXiv:2010.02824 (2020)
34. Portillo-Quintero, J.A., Ortiz-Bayliss, J.C., Terashima-Marín, H.: A straightforward framework for video retrieval using clip. In: Mexican Conference on Pattern Recognition. pp. 3–12. Springer (2021)

35. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. arXiv preprint arXiv:2103.00020 (2021)
36. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
37. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683 (2019)
38. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019)
39. Rohrbach, A., Rohrbach, M., Tandon, N., Schiele, B.: A dataset for movie description. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3202–3212 (2015)
40. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
41. Wang, G., Luo, C., Xiong, Z., Zeng, W.: Spm-tracker: Series-parallel matching for real-time visual object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3643–3652 (2019)
42. Wang, X., Wu, J., Chen, J., Li, L., Wang, Y.F., Wang, W.Y.: Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4581–4591 (2019)
43. Wu, Z., Yao, T., Fu, Y., Jiang, Y.G.: Deep learning for video classification and captioning. In: Frontiers of multimedia research, pp. 3–29 (2017)
44. Xu, J., Mei, T., Yao, T., Rui, Y.: Msr-vtt: A large video description dataset for bridging video and language. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5288–5296 (2016)
45. Yao, L., Huang, R., Hou, L., Lu, G., Niu, M., Xu, H., Liang, X., Li, Z., Jiang, X., Xu, C.: Filip: Fine-grained interactive language-image pre-training. arXiv preprint arXiv:2111.07783 (2021)
46. Ye, M., Shen, J., Lin, G., Xiang, T., Shao, L., Hoi, S.C.: Deep learning for person re-identification: A survey and outlook. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
47. Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Transactions of the Association for Computational Linguistics **2**, 67–78 (2014)
48. Yu, Y., Kim, J., Kim, G.: A joint sequence fusion model for video question answering and retrieval. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 471–487 (2018)
49. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: Self-supervised learning via redundancy reduction. In: International Conference on Machine Learning. pp. 12310–12320. PMLR (2021)