

Hybrid-Tower: Fine-grained Pseudo-query Interaction and Generation for Text-to-Video Retrieval

Bangxiang Lan¹ * Ruobing Xie² Ruixiang Zhao¹
 Xingwu Sun² Zhanhui Kang² Gang Yang^{1†} Xirong Li¹

¹Renmin University of China ²Large Language Model Department, Tencent

<https://lbx73737373.github.io/PIG-ProjectPage/>

Abstract

The Text-to-Video Retrieval (T2VR) task aims to retrieve unlabeled videos by textual queries with the same semantic meanings. Recent CLIP-based approaches have explored two frameworks: Two-Tower versus Single-Tower framework, yet the former suffers from low effectiveness, while the latter suffers from low efficiency. In this study, we explore a new Hybrid-Tower framework that can hybridize the advantages of the Two-Tower and Single-Tower framework, achieving high effectiveness and efficiency simultaneously. We propose a novel hybrid method, Fine-grained Pseudo-query Interaction and Generation for T2VR, i.e., PIG, which includes a new pseudo-query generator designed to generate a pseudo-query for each video. This enables the video feature and the textual features of pseudo-query to interact in a fine-grained manner, similar to the Single-Tower approaches to hold high effectiveness, even before the real textual query is received. Simultaneously, our method introduces no additional storage or computational overhead compared to the Two-Tower framework during the inference stage, thus maintaining high efficiency. Extensive experiments on five commonly used text-video retrieval benchmarks demonstrate that our method achieves a significant improvement over the baseline, with an increase of 1.6% ~ 3.9% in R@1. Furthermore, our method matches the efficiency of Two-Tower models while achieving near state-of-the-art performance, highlighting the advantages of the Hybrid-Tower framework.

1. Introduction

Recently, Text-to-Video Retrieval (T2VR) has seen significant improvements due to the advancement of the Contrastive Language-Image Pre-Training (CLIP) [33] model. Researchers have been putting effort into fully exploiting

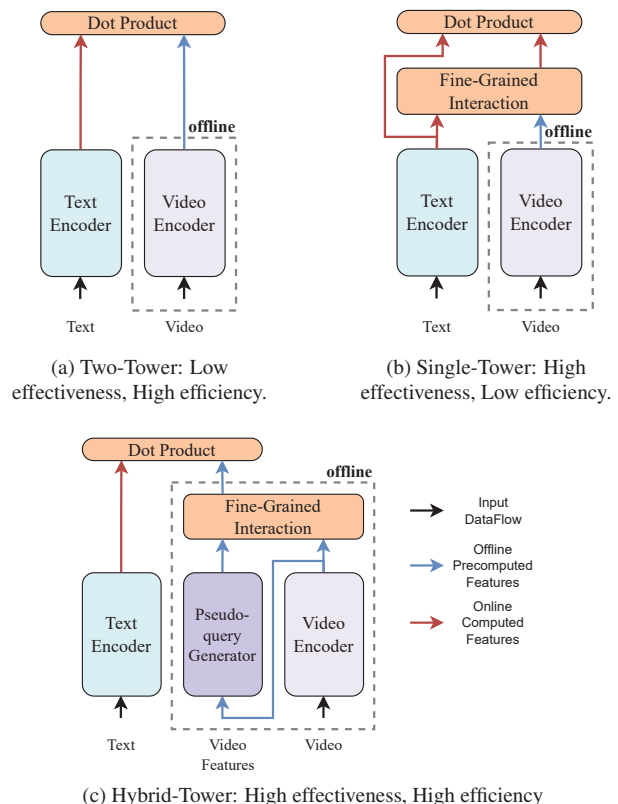


Figure 1. Illustration of two conventional text-to-video retrieval model frameworks: (a) Two-Tower, (b) Single-Tower and (c) our proposed Hybrid-Tower. The modules enclosed in dashed boxes indicate that they can be precomputed offline, so the bottleneck during online inference lies in other modules.

*Work performed as an intern at Tencent. (bangxiang@ruc.edu.cn)

†Corresponding author: Gang Yang (yanggang@ruc.edu.cn)

two classic frameworks: Two-Tower and Single-Tower, as shown in Fig. 1a and Fig. 1b, respectively. Both frameworks consist of a video encoder and a text encoder. After extracting text and video features, their similarity is computed using a simple dot product. Single-Tower methods introduce an additional fine-grained interaction module, such as similarity fusion in X-CLIP [29] or feature fusion in XPool [11]. This fine-grained interaction enables Single-Tower models to derive more accurate video representations, leading to improved retrieval performance compared to Two-Tower models on average.

However, in real-world retrieval scenarios, the interaction module in Single-Tower models become the bottleneck for retrieval speed. This is because, in Single-Tower models, video feature extraction is coupled with text queries; that is, for a given text query, every video in the dataset must undergo feature fusion in an additional interaction module. In contrast, Two-Tower models rely solely on a simple dot product, allowing the final video representations to be computed offline and stored in advance, making retrieval significantly faster than in Single-Tower models. The computational offline components in each framework are illustrated in Fig. 1. In summary, while Single-Tower models achieve higher retrieval accuracy through fine-grained interaction, they suffer from lower efficiency.

Given the limitations of existing frameworks, we ask: *Is there a new framework that can simultaneously achieve high retrieval speed and high retrieval accuracy?* In this paper, we propose a new Hybrid-Tower framework, illustrated in Fig. 1c, which contains a novel pseudo-query generator. This generator enables acquisition of pseudo-queries related to different videos before the arrival of real text queries, allowing the video features to perform a fine-grained interaction with pseudo-text queries, similar to Single-Tower models. We propose a method under Hybrid-Tower framework: Fine-grained Pseudo-query Interaction and Generation for T2VR, namely **PIG**. It tackles this problem through two core components: (i) *pseudo-query generator* that leverages multi-grained visual features and an informative token selection mechanism to generate discriminative pseudo-queries. (ii) *pseudo-interaction fusioner* that performs fine-grained pseudo-query interaction to enhance the final video representation. Experimental results on five standard datasets show that PIG could achieve near-SOTA performance while maintaining significantly higher efficiency. Furthermore, the new proposed framework could encourage further investigation in the Text-to-Video retrieval community. In summary, our contributions are as follows:

- We introduce a novel framework for T2VR: Hybrid-Tower, which combines and obtains the effectiveness of Single-Tower models and the efficiency of Two-Tower models.
- We propose the PIG method under the Hybrid-Tower

framework, which generates pseudo-queries to enable fine-grained feature fusion with videos to realize effective interaction. The core of PIG is a causal attention-powered pseudo-query generator with an informative token selection (ITS) module and a fine-grained pseudo-interaction fusioner.

- Extensive experiments conducted on five video retrieval datasets, *i.e.*, MSRVTT-1ka [46], MSRVTT-3k [43], MSVD [3], VATEX [39], and DiDeMo [13], demonstrate the effectiveness and efficiency of our method. Our proposed method achieves near-SOTA performance of Single-Tower and the SOTA efficiency of Two-Tower models, simultaneously.

2. Related work

2.1. Effective T2VR

The majority of existing literature focuses on obtaining more effective video and text representations. Based on whether text and video encoders are jointly trained with the similarity calculation module, current approaches can be categorized into two groups: feature re-learning and CLIP-based end-to-end methods within a Single-Tower architecture, as illustrated in Fig. 1a.

Feature re-learning methods typically use pre-trained 2D-CNNs [8, 22], 3D-CNNs [25, 30], or combinations thereof [10, 14] to generate initial video features. Text encoding usually employs non-trainable bag-of-words models [22] or pre-trained encoders like Word2Vec [7], BERT [22], and GPT [5]. Both video and text features are then projected into a shared latent space to evaluate their relevance based on their distances. Although improvements such as novel feature fusion or enhancement modules have been proposed [5, 10, 14], the performance of feature re-learning methods remains largely dependent on initial feature quality.

With advances in image-language models, CLIP-based methods have emerged as more effective for Text-to-Video Retrieval (T2VR), outperforming traditional feature re-learning methods. A critical component of these methods is the fine-grained text-video interaction within the Single-Tower framework, enhancing video representations. Interaction modules have progressively evolved from simple video-sentence alignment [28] to more detailed frame-sentence [9, 11, 38], frame-word [16, 29], and patch-word alignments [12, 21, 40]. However, increasing complexity in these interaction modules significantly reduces efficiency. As shown in Tab. 1, recent Single-Tower models such as UCoFia incur tens-of-thousands-fold efficiency costs for marginal performance improvements, presenting a major bottleneck for T2VR advancement.

2.2. Efficient T2VR

Depending on how the term “efficiency” is interpreted, existing studies can be categorized into three groups: (i) training efficiency, aiming to train T2VR models in a parameter-efficient fine-tuning (PEFT) manner [15, 18, 45]; (ii) feature extraction efficiency, focusing on reducing the size of the video encoder [32, 47]; and (iii) serving efficiency, which aims to boost speed in a real retrieval scenario, where video features are pre-cached. Our study falls into the third category.

To enhance serving speed, efficient CLIP-based methods typically retain only the simple dot product operation, fitting into the Two-Tower framework as illustrated in Fig. 1b. PromptSwitch [6] is among the first efficient CLIP-based T2VR methods, introducing a prompt cube into CLIP to iteratively model pair-to-pair temporal frame interactions. EERCF [36] adopts a two-stage retrieval strategy, first utilizing coarse-grained video representations to recall the top-k candidates rapidly, which are then reranked using finer-grained representations. TeachCLIP employs external fine-grained teachers, leveraging knowledge distillation, injecting the frame-sentence dependencies from the teacher into an Attentional frame-Feature Aggregation (AFA) module.

In contrast to these methods, PIG adopts a hybrid perspective. Specifically, it belongs to our novel Hybrid-Tower framework (see Fig. 1c), which retains the Two-Tower structure during serving while enhancing effectiveness through pseudo-query generation and pseudo-interactions with pre-cached video representations.

3. Method

We propose a novel PIG framework that involves pre-identifying potential text queries associated with target videos to achieve an optimal balance between computational efficiency and retrieval effectiveness.

3.1. Preliminaries

Given a multi-modality dataset consisting of N_v video clips and N_t text, *i.e.*, $\mathcal{D} = \{v_i, t_j\}$, where each video clip has one or more corresponding text queries. The goal of text-to-video retrieval (T2VR) is to rank the videos in the video gallery based on their relevance to a given text query. The method for T2VR typically employs a text encoder $\phi_t(\cdot)$ and a video encoder $\phi_v(\cdot)$ to abstract the video and the text into a shared cross-modal feature space:

$$\mathbf{v} = \phi_v(v); \mathbf{t} = \phi_t(t), \quad (1)$$

where $\mathbf{v}, \mathbf{t} \in \mathbb{R}^d$ and d denotes the embedding dimension of the feature space. The cosine similarity $s(\mathbf{t}, \mathbf{v}) = \frac{\mathbf{t} \cdot \mathbf{v}}{\|\mathbf{t}\| \|\mathbf{v}\|}$ is adopted to measure the distance between a video clip and a text query. In training, given a batch of B text-video pairs,

a widely-adopted symmetric contrastive loss *i.e.*, InfoNCE [29, 31, 33, 35] is used to optimize the model:

$$\mathcal{L}_{t2v} = -\frac{1}{B} \sum_i \log \frac{\exp(s(\mathbf{t}_i, \mathbf{v}_i) \cdot \tau)}{\sum_{j=1}^B \exp(s(\mathbf{t}_i, \mathbf{v}_j) \cdot \tau)}, \quad (2)$$

$$\mathcal{L}_{v2t} = -\frac{1}{B} \sum_i \log \frac{\exp(s(\mathbf{v}_i, \mathbf{t}_i) \cdot \tau)}{\sum_{j=1}^B \exp(s(\mathbf{v}_i, \mathbf{t}_j) \cdot \tau)}, \quad (3)$$

$$\mathcal{L}_{cons} = \frac{1}{2}(\mathcal{L}_{t2v} + \mathcal{L}_{v2t}), \quad (4)$$

where τ is a learnable temperature scaling factor.

One of the recent trends in CLIP-based T2VR is to strengthen temporal modeling ability for video encoder of CLIP [18, 21, 24, 26, 28, 44]. Among these models, CLIP-ViP [44] stands out as a powerful approach that introduces video proxy tokens and ViP-guided attention mechanism. Thus, we choose CLIP-ViP as the backbone encoder of our approach.

3.2. Overall Framework

In this work, we propose a novel Hybrid-Tower framework for T2VR and introduce a new model, PIG, which features fine-grained pseudo-query interaction and generation, as illustrated in Fig. 2. PIG can generate pseudo-query features from visual inputs to perform a fine-grained pseudo interaction of query-video in a Single-Tower manner ahead of the arrival of real textual queries. After the pseudo interaction, the learned video representations are retrieved by queries in an efficient Two-Tower manner. Since we can “pre-fuse” the pseudo query information into video representations, the effectiveness of Single-Tower methods and the efficiency of Two-Tower methods can be simultaneously assured. There are four essential components in PIG: a text encoder $\phi_t(\cdot)$, a video encoder $\phi_v(\cdot)$, a pseudo-query generator $\phi_g(\cdot)$ and a pseudo interaction fusioner $\phi_f(\cdot)$ as follows.

3.3. Text and Video Encoders

Given a video clip v comprising m frames $\{f_1, \dots, f_m\}$, the video encoder $\phi_v(\cdot)$ feeds the whole frame sequence into the visual encoder of CLIP [33], *i.e.*, a Vision Transformer (ViT) to produce visual features. The CLIP-ViP is applied as the backbone of our method with just a minor modification: we add the frame-level [cls] token to the ViT of CLIP-ViP, enabling it to provide extra frame-level visual outputs. With the help of video proxies in CLIP-ViP [44], we can obtain multi-grained visual features by selecting the video-level [cls], frame-level [cls] and patch-level tokens from the output of the last ViT layer, denoted as $\mathbf{x}'_v$, $\mathbf{x}'_f$, and $\mathbf{x}'_p$, respectively. These features are then projected into the cross-modal space through a visual linear projection layer in CLIP. These multi-grained features

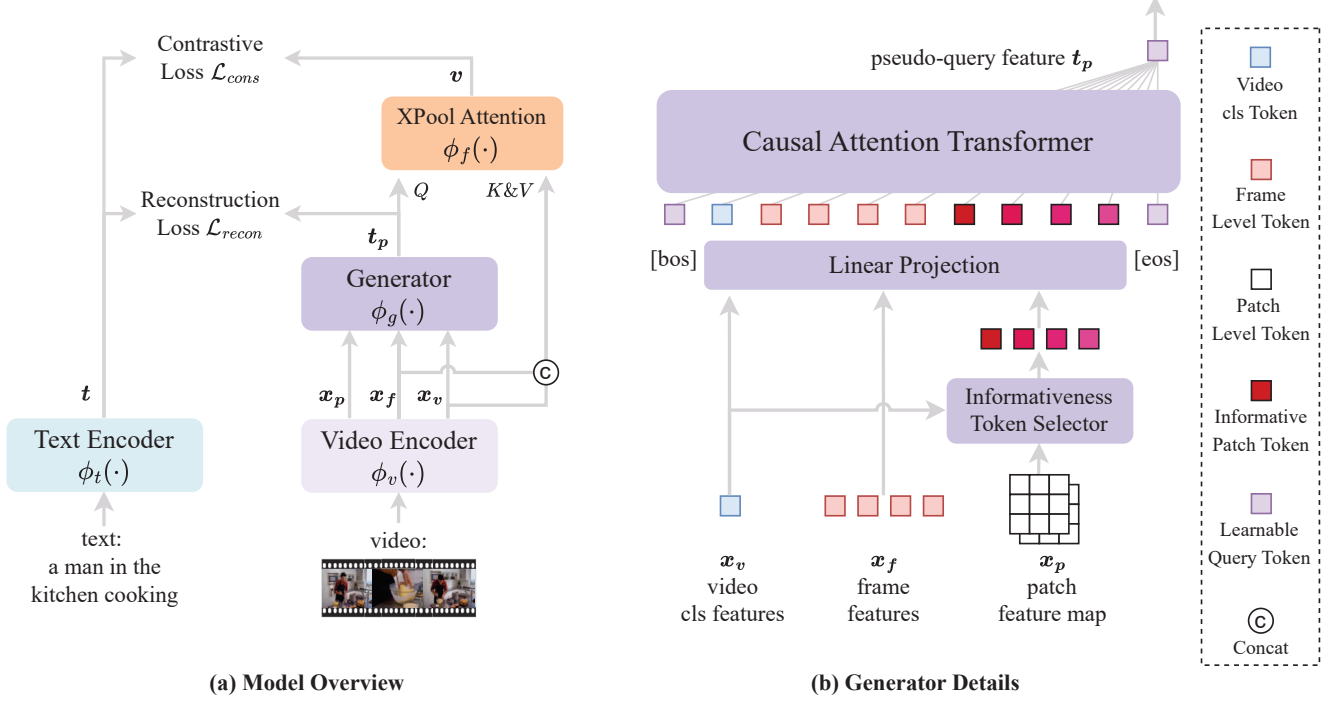


Figure 2. An overview of our proposed PIG for text-to-video retrieval: (a) the overall structure of our model. (b) A detailed illustration of our proposed pseudo-query generator.

are extracted for the next pseudo-query generation. More formally, the procedure can be expressed as follows:

$$\begin{cases} \{f_1, \dots, f_m\} \leftarrow \text{video-to-frames}(v), \\ \mathbf{x}'_v, \mathbf{x}'_f, \mathbf{x}'_p \leftarrow \text{ViT}(\{f_1, \dots, f_m\}), \\ \mathbf{x}_v, \mathbf{x}_f, \mathbf{x}_p \leftarrow \text{Linear}(\mathbf{x}'_v, \mathbf{x}'_f, \mathbf{x}'_p), \end{cases} \quad (5)$$

where $\mathbf{x}_v \in \mathbb{R}^{4 \times d}$, $\mathbf{x}_f \in \mathbb{R}^{m \times d}$, and $\mathbf{x}_p \in \mathbb{R}^{m \times n \times d}$, with m representing the number of frames, n the sequence length of patch tokens and 4 the number of video-level tokens.

The text encoder is the standard CLIP text encoder, which extracts text features $\mathbf{t} = \phi_t(t)$, where $t \in \mathbb{R}^d$.

3.4. Pseudo-Query Generator

After video encoding, we design a pseudo-query generator that takes multi-grained visual features as input to generate (or reconstruct) fine-grained and discriminative pseudo text query $\mathbf{t}_p$, as $\mathbf{t}_p = \phi_g([\mathbf{x}_v, \mathbf{x}_f, \mathbf{x}_p])$, where $[\cdot, \cdot]$ denotes concatenation, $\mathbf{t}_p \in \mathbb{R}^d$ and $[\mathbf{x}_v, \mathbf{x}_f, \mathbf{x}_p] \in \mathbb{R}^{(4+m+m \times n) \times d}$.

Pseudo-query generator is crucial for generating discriminative pseudo queries of text that can enhance video representations via multi-modal feature fusion. We break this down into two parts: selecting the input features of the generator, and designing the architecture of the generator.

Informativeness Token Selection. Generating or reconstructing a discriminative pseudo-query from video information is highly challenging, particularly in our cross-modal generation scenario. To maximize the use of multi-grained video features [9, 40], we feed video-level, frame-level, and patch-level features into our generator. Due to the high redundancy and noise of patch-level features, they should be concentrated. To this end, we propose a simple yet effective Informativeness Token Selector (ITS) module, as shown in Fig. 3. Due to the attention sparsity in ViT, where only a few tokens provide meaningful information (with background tokens even hindering performance)[20], it is natural to utilize this mechanism to highlight the important patch tokens. Specifically, given the first video-level [cls] token $\mathbf{x}_v \in \mathbb{R}^{1 \times d}$ and all the patch-level tokens $\mathbf{x}_p$, we calculate the attention scores of the video encoder’s last attention layer:

$$\begin{cases} Q_v^h \leftarrow \mathbf{x}_v W_Q^h, & K_p^h \leftarrow \mathbf{x}_p W_K^h, \\ S_h \leftarrow \text{softmax}\left(\frac{Q_v^h (K_p^h)^\top}{\sqrt{d/n}}\right), \\ S \leftarrow \max_h S_h, \end{cases} \quad (6)$$

where W_Q^h and W_K^h are the linear projections in the h -th head. The informativeness matrix $S \in [0, 1]^{m \times n}$ represents the importance level of each video patch token, obtained by applying a max operation across the multi-heads.

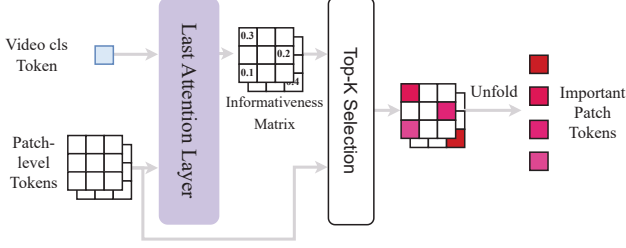


Figure 3. **Informativeness Token Selector.** We calculate the attention distribution of video cls token over all patch-level tokens to obtain an informativeness matrix. Referring to the informativeness matrix, we can select the most important top-k patch tokens.

Referring to the index of top-k values in the informativeness matrix, we select the top-k most informative patch tokens. After unfolding, we obtain the important patch-level tokens $\mathbf{x}_{ip} \in \mathbb{R}^{k \times d}$. Finally, by concatenating the original video-level features $\mathbf{x}_v$, frame-level features $\mathbf{x}_f$ and important patch-level features $\mathbf{x}_{ip}$ in a coarse-to-fine manner, the multi-grained input to our pseudo-query generator is $[\mathbf{x}_v, \mathbf{x}_f, \mathbf{x}_{ip}] \in \mathbb{R}^{(4+m+k) \times d}$.

Causal attention powered query generator. To better reconstruct text features, we implement our generator using a causal attention transformer, initialized with the CLIP text encoder. After a linear projection, we feed $[[\text{bos}], \mathbf{x}_v, \mathbf{x}_f, \mathbf{x}_{ip}, [\text{eos}]]$ into the transformer. The $[\text{bos}]$ and $[\text{eos}]$ tokens help facilitate the convergence during training. Finally, the output of the last transformer layer at the $[\text{eos}]$ position represents the pseudo query feature. That is $\mathbf{t}_p = \phi_g([\mathbf{x}_v, \mathbf{x}_f, \mathbf{x}_{ip}])$.

3.5. Pseudo-Interaction Fusiner

Then a cross-modal fusiner module is introduced to enhance the video features. We implement our fusiner with the XPool Attention module [11]. It can be seen as a one-layer cross attention without the residual connection, functioning as using the text feature as guidance to perform an attention pooling over a sequence of visual features. This attention pooling mechanism can learn to highlight the relevant frames to a given textual query while suppressing frames not described. To perform a fine-grained feature fusion, we utilize pseudo-query $\mathbf{t}_p$ as Q, concatenated video, frame level features $[\mathbf{x}_v, \mathbf{x}_f] \in \mathbb{R}^{(4+m) \times d}$ as K&V. Finally, the fusiner outputs the final video representation $\mathbf{v} \in \mathbb{R}^d$ enhanced by “pre-query video interaction modeling” as follows:

$$\begin{cases} \mathbf{v}' & \leftarrow \text{LN}(\text{Attention}(Q : \mathbf{t}_p, K\&V : [\mathbf{x}_v, \mathbf{x}_f])), \\ \mathbf{v} & \leftarrow \text{LN}(\text{FC}(\mathbf{v}') + \mathbf{v}'), \end{cases} \quad (7)$$

where LN is a Layer Normalization layer [1] and FC is a fully connected network. The learned final video representation $\mathbf{v}$ is then used for video retrieval in an efficient

Two-Tower manner (in Sec. 3.6). Throughout our hybrid-tower framework, the finer-grained modeling of multimodal feature extraction, generation and interaction are well concerned.

3.6. Training and Inference

Training. During training, we use the real text features $\mathbf{t}$ as supervision to constrain the reconstructed pseudo-query features $\mathbf{t}_p$ from videos, thereby optimizing the generator. The reconstruction loss used here is the cosine distance loss [20, 34],

$$\mathcal{L}_{recon} = 1 - \text{sim}_{cos}(\mathbf{t}_p, \mathbf{t}), \quad (8)$$

where $\text{sim}_{cos}()$ denotes the cosine similarity between two features. In summary, the total objective is defined by combining $\mathcal{L}_{cons}$ in Eq. 4 and $\mathcal{L}_{recon}$ in Eq. 8 as:

$$\mathcal{L} = \mathcal{L}_{cons} + \alpha \mathcal{L}_{recon}, \quad (9)$$

where α is the scaling weight for the reconstruction loss. Our training process consists of two stages: (1) pretraining the generator using only reconstruction objectives while keeping the encoders and fusion module frozen; (2) fine-tuning the entire network with both contrastive and reconstruction objectives. It is worth noting that neither training stage introduces extra data.

Inference. In inference, all video features can be precalculated and stored offline, allowing the generator to generate corresponding pseudo-queries for each video. This enables offline fine-grained feature fusion, ensuring both Two-Tower level efficiency and Single-Tower level effectiveness.

4. Experiment

4.1. Experimental Setup

Datasets. We conduct experiments on four public datasets for evaluation, including: (1) **MSRVTT** [43], which consists of 10K YouTube video clips, each paired with 20 human-labeled captions. Following [37], we use two splits: MSRVTT-1k [46], which contains 9K videos for training and 1K video-text pairs for testing; and MSRVTT-3k [43], the original split, which consists of nearly 7K videos for training and 3K video-text pairs for testing. The MSRVTT-1k split is used as the primary dataset for our ablation study. (2) **MSVD** [3] contains 1,970 videos, with 80K captions, averaging 40 captions per video. We follow the official split of 1,200 and 670 as the train and test set, respectively. (3) **VATEX** [39] is comprised of around 35K videos, each with multiple annotations. Due to some unavailable videos, we follow the split in [37], which includes 23,896 videos for training, 1,375 for validation, and 1,398 for testing. (4) **DiDeMo** [13] consists of nearly 10k video clips and 40k captions in total. We adopt the official data split which

Model	FLOPs ↓	MSRVTT-1k			MSRVTT-3k			MSVD			VATEX			Mean
		R@1	R@5	SumR	R@1	R@5	SumR	R@1	R@5	SumR	R@1	R@5	SumR	
Feature re-learning with CLIP feature:														
SEA [23]	10.0K	37.2	67.1	182.6	19.9	44.3	120.7	34.5	68.8	183.8	52.4	90.2	238.5	36.0
W2VV++ [22]	2.0K	39.4	68.1	185.6	23.0	49.0	132.7	37.8	71.0	190.4	55.8	91.2	243.0	39.0
MMT [10]	3.5K	39.5	68.3	186.1	24.9	50.5	137.4	40.6	72.0	194.3	54.4	89.2	238.6	39.9
LAFF [14]	4.0K	45.8	71.5	199.3	29.1	54.9	149.8	45.4	70.6	200.6	59.1	91.7	247.1	44.9
Single-Tower CLIP-based end-to-end (visual backbone: ViT-B/32):														
TS2-Net [26]	6.1K	46.7	72.6	200.5	29.9	56.4	153.6	44.6	75.8	204.9	61.1	91.5	248.6	45.6
X-CLIP [29]	220.9K	45.3	73.7	200.8	31.2	57.4	156.7	47.2	77.0	210.1	62.2	90.9	248.5	46.5
XPool [11]	275.0K	46.0	72.8	201.5	—	—	—	—	—	—	—	—	—	—
DRL [38]	220.4K	46.2	74.0	203.2	—	—	—	—	—	—	—	—	—	—
ProST [21]	2017.5K	48.2	74.6	206.2	—	—	—	—	—	—	62.6	91.3	249.3	—
UCoFiA [40]	9836.7K	49.1	72.1	202.7	—	—	—	47.1	77.1	208.5	62.7	90.3	248.1	—
Two-Tower CLIP-based end-to-end (visual backbone: ViT-B/32):														
CLIP4Clip [28]	0.5K	42.8	71.6	195.5	29.4	54.9	150.1	45.6	76.1	206.6	61.6	91.1	248.5	44.9
CenterCLIP [47]	1.5K	44.2	71.6	197.9	—	—	—	47.3	76.8	209.7	—	—	—	—
Cap4Video* [41]	1.0K	45.6	71.7	81.2	—	—	—	—	—	—	—	—	—	—
PromptSwitch [6]	0.5K	43.6	71.5	195.7	—	—	—	46.3	75.8	206.6	—	—	—	—
STAN [24]	0.5K	46.9	72.8	202.5	—	—	—	—	—	—	—	—	—	—
TeachCLIP [37]	0.5K	46.8	74.3	203.7	30.9	57.1	156.0	47.4	77.3	210.2	63.6	91.9	251.6	47.2
CLIP-ViP* [44]	0.5K	46.0	72.7	200.6	30.2	56.0	155.3	45.1	74.8	206.5	62.1	90.2	249.3	45.9
PIG(Ours)	0.5K	48.6	72.8	203.0	31.8	57.3	157.1	47.9	75.9	207.2	64.0	91.5	252.1	48.0

Table 1. **T2VR Performance of different methods on multiple datasets.** Note that we replicate existing methods using their author-provided source code where applicable, so the numbers might (slightly) differ from those in their original papers. None of the reported methods is applied by post-processing, as it is not practical in real retrieval scenarios. Cap4Video* refers to the Two-Tower version (global matching version) of Cap4Video. CLIP-ViP* refers to the version without pre-training on external large-scale datasets. Mean in the last column refers to the mean of R@1 across all datasets. FLOPs indicates the computational cost of video-text matching per pair during inference.

has 8,394 video clips for training, 1,065 for validation, and 1,004 for testing.

Evaluation criteria. Following previous works, we utilize the most common rank-based metrics, *i.e.*, Recall at top-k (R@k, k=1, 5, 10), SumR (R1+R5+R10) and mean rank (MnR) to evaluate models.

Implementation Details. Experiments are conducted on 8 NVIDIA 3090 GPUs. The default setting is as follows, unless otherwise stated. We use the CLIP model initialized by OpenAI-released version. The learning rate is set to $9e-5$ during the pseudo-query generation pertaining stage and $1e-6$ during full fine-tuning. The maximum length of frame / word tokens is set to 12/50, respectively, with a batch size of 128. For DiDeMo’s longer clips, we follow [44] and use 32/64 tokens. The α in Eq. 9 is set to 2 and the top-k in informativeness token selection is set to top-16 for optimal performance.

4.2. Performance Comparison

Baselines. Both feature re-learning based methods and CLIP-based end-to-end methods are compared. For the purpose of fair comparison and reproducibility of research, we only include feature re-learning methods with CLIP feature and those open-sourced methods as follows:

- *Feature re-learning:* SEA [23], W2VV++ [22], MMT [10] and LAFF [14].
- *Single-Tower CLIP-based end-to-end:* TS2-Net [26], X-CLIP [29], XPool [11], DRL [38], ProST [21] and UCoFiA [40].
- *Two-Tower CLIP-based end-to-end:* CLIP4clip [28], CenterCLIP [47], Cap4Video [41], PromptSwitch [6], STAN [24], TeachCLIP [37] and our backbone baseline CLIP-ViP [44].

Among the end-to-end methods, only some of them have results across all datasets, see Tab. 1.

#	Setup	R@1↑	R@5↑	R@10↑	MnR↓
0	Full-setup	48.6	72.8	81.6	15.4
<i>Loss Functions:</i>					
1	$w/o \mathcal{L}_{recon}$	45.8	71.8	81.5	16.7
<i>Input of Generator:</i>					
2	$\phi_g(\mathbf{x}_v)$ as $\mathbf{t}_p$	46.3	72.0	81.4	16.1
3	$\phi_g([\mathbf{x}_v, \mathbf{x}_f])$ as $\mathbf{t}_p$	47.5	71.8	81.5	15.9
4	$\phi_g([\mathbf{x}_f, \mathbf{x}_{ip}])$ as $\mathbf{t}_p$	48.1	71.8	81.4	15.6
<i>Architecture of Generator:</i>					
5	Causal Attn. $\rightarrow$ MLP	44.1	70.5	80.2	17.1
6	Causal Attn. $\rightarrow$ Q-former [2, 19]	47.6	72.5	81.0	15.5
<i>Architecture of Fusioner:</i>					
7	XPool $\rightarrow$ Cross Attn.	42.2	70.0	80.6	16.2
8	XPool $\rightarrow$ Co Attn. [27]	44.5	71.6	80.3	17.0
<i>Form of Pseudo-Query:</i>					
9	Captions (ShareGPT4Video [4])	43.8	69.2	78.5	18.5
10	Captions (m-PLUG2 [42])	47.4	71.7	82.0	15.3

Table 2. **Ablation Study of PIG.** Backbone: CLIP-ViT-B/32. Dataset: MSRVT-1k.

Efficiency comparison. Following [37], we calculate the FLOPs¹ required for per video-text matching during serving, *i.e.*, video features can be pre-extracted offline and stored in advance. As Tab. 1 shows, our method achieves the best efficiency of 0.5K FLOPs per matching by keeping the retrieval process in a Two-Tower manner. We also report additional results from both online and offline perspectives in Tab. 3. The goal of our work is to improve online-stage performance, and given the strong gains achieved, we consider the offline overhead acceptable.

Effectiveness comparison. Tab. 1 presents the performance of various methods on multiple video retrieval benchmarks. As observed, feature re-learning models (top section) are notably inferior to CLIP-based end-to-end methods. As for the end-to-end methods, our method PIG stands out with a mean R@1 of 48.0, clearly outperforming its Two-Tower counterparts. In particular, the backbone model of PIG, CLIP-ViP, is enhanced, yielding R@1 improvements of 2.6 on MSRVT-1k, 1.6 on MSRVT-3k, 2.8 on MSVD, and 1.9 on VATEX, thereby surpassing the previous state-of-the-art TeachCLIP. Moreover, although Single-Tower models such as ProST and UCoFiA achieve slight gains in R@1 (0.5) and R@5 (1.8), they incur 4,035-fold and 19,673-fold higher inference costs, respectively, compared to PIG. Thus, the exceptionally low computational cost of PIG (0.5K FLOPs) ensures remarkable efficiency, making it highly suitable for real-time retrieval applications. These observations, along with its robust performance across diverse datasets, demonstrate that PIG balances high retrieval effectiveness with outstanding computational efficiency.

¹<https://github.com/sovrasov/flops-counter.pytorch>

Model	Offline Stage		Online Stage		R@1 (↑)
	Video feature extraction (↓) (FLOPs)	Model (↓) parameters (M)	Per video-text matching (↓) (FLOPs)	Video feature storage (↓) (KB)	
ProST	53.89G	180.4	2017.5K	294	48.2
TeachCLIP	53.65G	164.2	0.5K	2	46.8
<i>F-Pig(Ours)</i>	54.90G	191.0	0.5K	2	48.6

Table 3. **Online/offline overhead.** Dataset: MSRVT-1k.

#	Setup	MSRVT-1k			DiDeMo		
		R@1↑	R@5↑	R@10↑	R@1↑	R@5↑	R@10↑
<i>Baselines:</i>							
0	CLIP4Clip (B/32)	42.8	71.6	81.1	42.0	69.0	78.2
1	CLIP-ViP (B/32)	46.0	72.7	81.8	39.5	70.1	76.9
2	CLIP-ViP (B/16)	49.4	73.6	84.0	41.8	71.0	80.3
<i>Backbone of PIG:</i>							
3	PIG w CLIP4Clip (B/32)	45.1	72.1	81.5	44.1	71.9	81.3
4	PIG w CLIP-ViP (B/32)	48.6	72.8	81.6	41.2	67.9	80.5
5	PIG w CLIP-ViP (B/16)	51.2	75.1	84.5	43.3	70.8	81.9

Table 4. **Performance of PIG with different backbones and sizes.** Here, B/32 and B/16 refer to CLIP-ViT-B/32 and CLIP-ViT-B/16 backbones, respectively.

We also evaluate our model with different backbone scales and architectures. As shown in Tab. 4, whether we vary size (Setup-1/2 vs Setup-4/5) or architecture (Setup-0/1 vs Setup-3/4), our model consistently outperforms the baselines. We find that performance on DiDeMo is poor when CLIP-ViP is used as the backbone. This counterintuitive result stems from DiDeMo’s long videos (32 frames), which force CLIP-ViP’s proxy attention to handle 32×49 tokens simultaneously, which is very challenging. After switching the backbone to CLIP4Clip, the absolute R@1 on DiDeMo increases from 41.2 to 44.1.

4.3. Ablation Studies

The essence of our PIG is the pseudo-query generator and the pseudo-interaction fusioner. Hence, PIG needs to be evaluated along multiple dimensions, including choice of generator’s input, architecture of generator, the architecture of fusioner, the form of pseudo-query.

Choice of Input Visual Tokens. The input visual features for the generator include video-level ($\mathbf{x}_v$), frame-level ($\mathbf{x}_f$), and informative patch-level ($\mathbf{x}_{ip}$) features.

As shown in Tab. 2, PIG consistently surpasses the baseline regardless of feature granularity, achieving R@1 improvements from 0.3 to 2.6. Notably, informative patch-level features yield the greatest gain, while the best results are obtained when using all multi-grained inputs. Without top-k selection, feeding all patch-level features causes out-of-memory issues; for example, a 12-frame video with CLIP-ViT-B/32 yields $12 \times 49 = 588$ patch features. This highlights the necessity of reducing patch-level feature numbers.

Model	MSVD	VATEX	DiDeMo	Mean
CLIP4Clip	43.7	48.5	30.3	40.8
X-CLIP	44.8	48.9	33.5	42.4
TeachCLIP	44.0	49.1	31.3	41.5
F-Pig	44.9	49.8	30.9	42.0

Table 5. **Cross-dataset results.** Backbone: CLIP-ViT-B/32. Training data: MSRVT-1k.

Architecture of generator. As Tab. 2 shows, switching from the causal attention transformer initialized by CLIP’s text encoder to other architectures (Setup-5, Setup-6) degrades performance. Due to its sequence modeling ability, the causal attention transformer naturally handles our coarse-to-fine multi-grained visual inputs.

Architecture of fusioner. The effectiveness of different fusioner architectures is evaluated in Tab. 2. XPool Attention (Setup-0) outperforms Cross Attention (Setup-7) and Co Attention (Setup-8) across all metrics. The latter two lag because of their residual connection, which directly adds the weak pseudo-query feature to the final video representation, thereby weakening its quality. By contrast, our design captures fine-grained text–frame dependencies more effectively.

Form of pseudo-query. How to generate effective pseudo-queries? There are two straightforward solutions: (1) utilizing an external video captioner to generate raw auxiliary captions as pseudo-queries [41]; (2) training a pseudo-query generator aligned with the text features of real queries (our method). We employ two powerful video captioning models, ShareGPT4Video [4] and m-plug2 [42], to generate auxiliary captions on MSRVT-1ka. The fine-tuned CLIP text encoder is then used to extract auxiliary text features as pseudo-query features. As shown in Tab. 2, our solution (Setup-0) outperforms the two caption-based solutions (Setup-9, Setup-10), demonstrating that our trained pseudo-query generator produces better text features.

Reconstruction Loss. Removing text feature supervision significantly harms our model’s performance (Setup-1). Without the guidance of our reconstruction loss, the pseudo-queries degenerate into meaningless noise.

Cross dataset evaluation. To check whether the generated pseudo-queries are restricted to a specific dataset domain, we evaluate our method in out-of-domain retrieval settings [17, 37]. As shown in Tab. 5, we directly test our model across different datasets. PIG achieves the best R@1 performance on MSVD and VATEX, demonstrating the strong generalization ability of our approach.

4.4. Qualitative Analysis.

Embedding space visualization. Fig. 4 shows the t-SNE plot of the joint embedding space. The pseudo-text features generated solely from video blend seamlessly with the real text features, demonstrating that our method learns

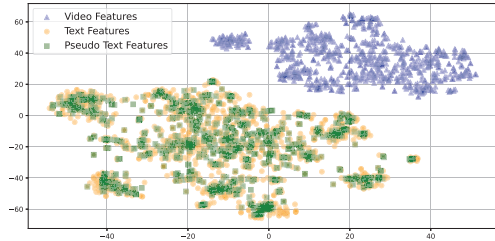


Figure 4. t-SNE visualization of the embedding space generated by our PIG on the MSRVT-1k test set.

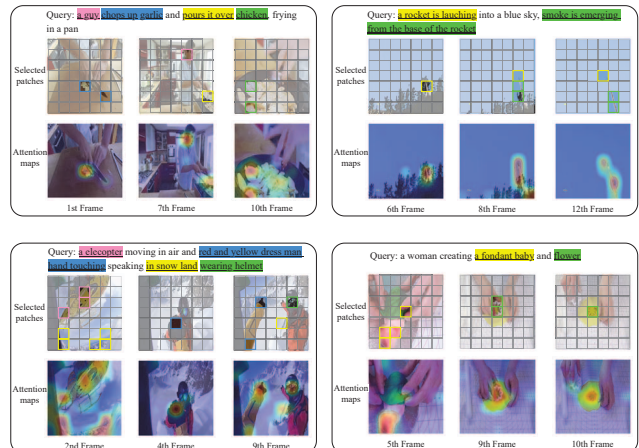


Figure 5. **Qualitative results of our ITS module.** Top row: patch tokens selected by our ITS module, which are then fed into the pseudo-query generator. Bottom row: frame attention maps. The corresponding patch tokens with the phrases in the query are highlighted in the same color.

meaningful pseudo-query embeddings aligned with the distribution of real text queries. Consequently, these pseudo-queries effectively guide the video features toward a finer-grained visual representation.

Qualitative Examples. Fig. 5 presents qualitative results on MSRVT-1k. Leveraging our ITS module, we retain the patches that contain the fine-grained information needed to generate discriminative pseudo-queries.

5. Conclusion

We propose a Hybrid-Tower framework for Text-to-Video Retrieval (T2VR) that balances the effectiveness of Single-Tower models with the efficiency of Two-Tower models. Our method, Fine-grained Pseudo-query Interaction and Generation (PIG), introduces a pseudo-query generator and fusioner to enable fine-grained feature fusion before the arrival of real queries. This achieves near-SOTA accuracy while maintaining efficiency. Extensive experiments validate its superiority, highlighting its potential to inform future research in T2VR.

Acknowledgments This work was supported by NSFC (62172420), Tencent Marketing Solution Rhino-Bird Focused Research Program and the Young Elite Scientists Sponsorship Program by CAST (2023QNRC001).

References

- [1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 5
- [2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 7
- [3] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, 2011. 2, 5
- [4] Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Zhenyu Tang, Li Yuan, et al. Sharegpt4video: Improving video understanding and generation with better captions. *NeurIPS*, 2025. 7, 8
- [5] Ioana Croitoru, Simion-Vlad Bogolin, Marius Lordeanu, Hailin Jin, Andrew Zisserman, Samuel Albanie, and Yang Liu. Teachtext: Crossmodal generalized distillation for text-video retrieval. In *CVPR*, 2021. 2
- [6] Chaorui Deng, Qi Chen, Pengda Qin, Da Chen, and Qi Wu. Prompt switch: Efficient clip adaptation for text-video retrieval. In *ICCV*, 2023. 1, 3, 6
- [7] Jianfeng Dong, Xirong Li, and Cees GM Snoek. Predicting visual features from text for image and video caption retrieval. *TMM*, 2018. 2
- [8] Jianfeng Dong, Xirong Li, Chaoxi Xu, Xun Yang, Gang Yang, Xun Wang, and Meng Wang. Dual encoding for video retrieval by text. *IEEE TPAMI*, 2021. 2
- [9] Bo Fang, Wenhao Wu, Chang Liu, Yu Zhou, Yuxin Song, Weiping Wang, Xiangbo Shu, Xiangyang Ji, and Jingdong Wang. Uatvr: Uncertainty-adaptive text-video retrieval. In *ICCV*, 2023. 2, 4
- [10] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *ECCV*, 2020. 2, 6
- [11] Satya Krishna Gorti, Noël Vouitsis, Junwei Ma, Keyvan Golestan, Maksims Volkovs, Animesh Garg, and Guangwei Yu. X-pool: Cross-modal language-video attention for text-video retrieval. In *CVPR*, 2022. 1, 2, 5, 6
- [12] Peiyan Guan, Renjing Pei, Bin Shao, Jianzhuang Liu, Weimian Li, Jiayi Gu, Hang Xu, Songcen Xu, Youliang Yan, and Edmund Y Lam. Pidro: Parallel isomeric attention with dynamic routing for text-video retrieval. In *CVPR*, 2023. 2
- [13] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with temporal language. In *EMNLP*, 2018. 2, 5
- [14] Fan Hu, Aozhu Chen, Ziyue Wang, Fangming Zhou, Jianfeng Dong, and Xirong Li. Lightweight attentional feature fusion: A new baseline for text-to-video retrieval. In *ECCV*, 2022. 2, 6
- [15] Siteng Huang, Biao Gong, Yulin Pan, Jianwen Jiang, Yiliang Lv, Yuyuan Li, and Donglin Wang. Vop: Text-video co-operative prompt tuning for cross-modal retrieval. In *CVPR*, 2023. 3
- [16] Peng Jin, Jinfa Huang, Pengfei Xiong, Shangxuan Tian, Chang Liu, Xiangyang Ji, Li Yuan, and Jie Chen. Video-text as game players: Hierarchical banzhaf interaction for cross-modal representation learning. In *CVPR*, 2023. 1, 2
- [17] Peng Jin, Hao Li, Zesen Cheng, Kehan Li, Xiangyang Ji, Chang Liu, Li Yuan, and Jie Chen. Diffusionret: Generative text-video retrieval with diffusion model. In *ICCV*, 2023. 1, 8
- [18] Xiaojie Jin, Bowen Zhang, Weibo Gong, Kai Xu, Xueqing Deng, Peng Wang, Zhao Zhang, Xiaohui Shen, and Jiashi Feng. Mv-adaptor: Multimodal video transfer learning for video text retrieval. In *CVPR*, 2024. 3
- [19] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. 7
- [20] Kunchang Li, Yali Wang, Yizhuo Li, Yi Wang, Yinan He, Limin Wang, and Yu Qiao. Unmasked teacher: Towards training-efficient video foundation models. In *ICCV*, 2023. 4, 5
- [21] Pandeng Li, Chen-Wei Xie, Liming Zhao, Hongtao Xie, Jiannan Ge, Yun Zheng, Deli Zhao, and Yongdong Zhang. Progressive spatio-temporal prototype matching for text-video retrieval. In *CVPR*, 2023. 2, 3, 6
- [22] Xirong Li, Chaoxi Xu, Gang Yang, Zhineng Chen, and Jianfeng Dong. W2vv++ fully deep learning for ad-hoc video search. 2019. 2, 6
- [23] Xirong Li, Fangming Zhou, Chaoxi Xu, Jiaqi Ji, and Gang Yang. Sea: Sentence encoder assembly for video retrieval by textual queries. *IEEE TMM*, 2020. 6
- [24] Ruyang Liu, Jingjia Huang, Ge Li, Jiashi Feng, Xinglong Wu, and Thomas H Li. Revisiting temporal modeling for clip-based image-to-video knowledge transferring. In *CVPR*, 2023. 1, 3, 6
- [25] Yang Liu, Samuel Albanie, Arsha Nagrai, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts. In *BMVC*, 2019. 2
- [26] Yuqi Liu, Pengfei Xiong, Luhui Xu, Shengming Cao, and Qin Jin. Ts2-net: Token shift and selection transformer for text-video retrieval. In *ECCV*, 2022. 1, 3, 6
- [27] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *NeurIPS*, 2019. 7
- [28] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 2022. 1, 2, 3, 6
- [29] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *ACM MM*, 2022. 1, 2, 3, 6
- [30] Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K Roy-Chowdhury. Learning joint embedding

- with multimodal cues for cross-modal video-text retrieval. In *ICMR*, 2018. 2
- [31] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3
- [32] Renjing Pei, Jianzhuang Liu, Weimian Li, Bin Shao, Songcen Xu, Peng Dai, Juwei Lu, and Youliang Yan. Clipping: Distilling clip-based models with a student base for video-language retrieval. In *CVPR*, 2023. 3
- [33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 3
- [34] Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Am-radio: Agglomerative vision foundation model reduce all domains into one. In *CVPR*, 2024. 5
- [35] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *NeurIPS*, 2016. 3
- [36] Kaibin Tian, Yanhua Cheng, Yi Liu, Xinglin Hou, Quan Chen, and Han Li. Towards efficient and effective text-to-video retrieval with coarse-to-fine visual representation learning. In *AAAI*, 2024. 3
- [37] Kaibin Tian, Ruixiang Zhao, Zijie Xin, Bangxiang Lan, and Xirong Li. Holistic features are almost sufficient for text-to-video retrieval. In *CVPR*, 2024. 1, 5, 6, 7, 8
- [38] Qiang Wang, Yanhao Zhang, Yun Zheng, Pan Pan, and Xian-Sheng Hua. Disentangled representation learning for text-video retrieval. *arXiv preprint arXiv:2203.07111*, 2022. 2, 6
- [39] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatec: A large-scale, high-quality multilingual dataset for video-and-language research. In *ICCV*, 2019. 2, 5
- [40] Ziyang Wang, Yi-Lin Sung, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Unified coarse-to-fine alignment for video-text retrieval. In *ICCV*, 2023. 2, 4, 6
- [41] Wenhao Wu, Haipeng Luo, Bo Fang, Jingdong Wang, and Wanli Ouyang. Cap4video: What can auxiliary captions do for text-video retrieval? In *CVPR*, 2023. 1, 6, 8
- [42] Haiyang Xu, Qinghao Ye, Ming Yan, Yaya Shi, Jiabo Ye, Yuanhong Xu, Chenliang Li, Bin Bi, Qi Qian, Wei Wang, et al. mplug-2: A modularized multi-modal foundation model across text, image and video. In *ICML*, 2023. 7, 8
- [43] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016. 2, 5
- [44] Hongwei Xue, Yuchong Sun, Bei Liu, Jianlong Fu, Ruihua Song, Houqiang Li, and Jiebo Luo. Clip-vip: Adapting pre-trained image-text model to video-language alignment. In *ICLR*, 2023. 3, 6
- [45] Xiangpeng Yang, Linchao Zhu, Xiaohan Wang, and Yi Yang. Dgl: Dynamic global-local prompt tuning for text-video retrieval. In *AAAI*, 2024. 3
- [46] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *ECCV*, 2018. 2, 5
- [47] Shuai Zhao, Linchao Zhu, Xiaohan Wang, and Yi Yang. Centerclip: Token clustering for efficient text-video retrieval. 2022. 3, 6
- [48] Xianwei Zhuang, Hongxiang Li, Xuxin Cheng, Zhihong Zhu, Yuxin Xie, and Yuexian Zou. Kdpror: A knowledge-decoupling probabilistic framework for video-text retrieval. In *ECCV*, 2024. 1