Holistic Features are almost Sufficient for Text-to-Video Retrieval

Kaibin Tian[1*†] Ruixiang Zhao[1*] Zijie Xin[1,2] Bangxiang Lan[1] Xirong Li[1‡]
[1]MoE Key Lab of DEKE, Renmin University of China
[2]College of Computer Science, Sichuan University
https://github.com/ruc-aimc-lab/TeachCLIP

Abstract

For text-to-video retrieval (T2VR), which aims to retrieve unlabeled videos by ad-hoc textual queries, CLIP-based methods currently lead the way. Compared to CLIP4Clip which is efficient and compact, state-of-the-art models tend to compute video-text similarity through fine-grained cross-modal feature interaction and matching, putting their scalability for large-scale T2VR applications into doubt. We propose TeachCLIP*, enabling a CLIP4Clip based student network to learn from more advanced yet computationally intensive models. In order to create a learning channel to convey fine-grained cross-modal knowledge from a heavy model to the student, we add to CLIP4Clip a simple Attentional frame-Feature Aggregation (AFA) block, which by design adds no extra storage / computation overhead at the retrieval stage. Frame-text relevance scores calculated by the teacher network are used as soft labels to* supervise *the attentive weights produced by AFA. Extensive experiments on multiple public datasets justify the viability of the proposed method. TeachCLIP has the same efficiency and compactness as CLIP4Clip, yet has near-SOTA effectiveness.*

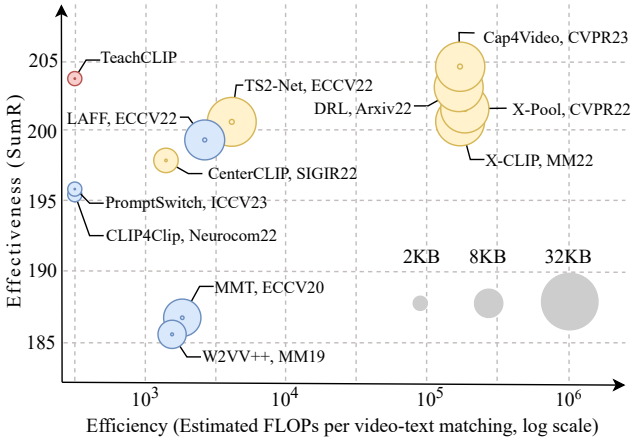

Figure 1. **Effectiveness, efficiency and video-feature storage footprint of current (CLIP based) text-to-video retrieval (T2VR) models**. Backbone: CLIP(ViT-B/32) [29]. Dataset: MSRVTT-1k [37]. Yellow circles indicate models using fine-grained cross-modal feature interaction and matching. The proposed *TeachCLIP* strikes a good balance between the three factors.

1. Introduction

Are *holistic* features, *i.e.* representing a given video / textual query by a single feature vector, sufficient for text-to-video retrieval (T2VR)? The question is scientifically interesting due to the cross-modal nature of the T2VR task: video and query have to be encoded into a semantically aligned common feature space for video-text matching [6]. The question is also practically valuable as matching by holistic features is much more scalable than matching by local features.

Due to the great success of the Contrastive Language-Image Pre-Training (CLIP) model [29] in the image domain, we see encouraging efforts on re-purposing CLIP for video-text matching [7, 9, 22, 24, 25, 36]. As an initial effort, CLIP4Clip [24] encodes a given video by first using CLIP's visual encoder to extract image features per frame. The frame-level features, enhanced by Transformer blocks, are aggregated into a video-level feature by mean pooling. The video feature, with a typical size of 512, can be computed and stored in advance, making CLIP4Clip efficient for T2VR. Recent methods, *e.g.* X-CLIP [25], TS2-Net [22] and X-Pool [9], improve over CLIP4Clip by considering fine-grained frame-text similarities. Despite their better retrieval performance on multiple benchmark datasets [1, 35], these models introduce substantial overhead w.r.t. offline storage and online computation, see Fig. 1 and Tab. 1, putting their scalability for large-scale T2VR into doubt.

Luckily, the importance of holistic features for CLIP-based T2VR has not gone unnoticed. In a contemporary work [4], a novel method termed PromptSwitch has been developed. There, a spatial-temporal prompt cube is added

*Equal contributions.
†Kaibin Tian is currently with Kuaishou.
‡Corresponding author: Xirong Li (xirong@ruc.edu.cn)

to the CLIP image encoder to model intra-frame and inter-frame visual token relationships for better video representation. We take a more simplified approach: knowledge distillation (KD) [12]. KD was originally developed to improve the classification performance of a student model by transferring dark knowledge from a relatively larger and stronger teacher model to the student. Good effort on exploiting KD for T2VR already exists before the advent of CLIP. TeachText [3] first trains an ensemble of T2VR models with varied textual encoders. The averaged video-text similarity given by the ensemble is then used as soft labels to supervise a specific student network. TeachText is designed specifically for video-level knowledge distillation. Therefore, it remains unclear how to effectively pass fine-grained cross-modal knowledge, which is crucial for the current T2VR models' good performance, on to the student.

In order to set a fine-grained learning channel for the student, we propose to add an Attentional frame-Feature Aggregation (AFA) block to CLIP4Clip, see Fig. 2. Given a sequence of frame-level features as input, the AFA block produces frame-specific weights, which will be used to aggregate the frame features into a *holistic* video-level feature. So different from fine-grained T2VR models [9, 25], we need no frame feature at the retrieval stage. The AFA block alone is not new, as similar blocks have been used for fusing diverse video/text features [13]. Our innovation is being the first to *supervise* the attentional weights with frame-text relevance scores given by the teacher network. AFA thus creates a channel to accept the fine-grained knowledge from the teacher. As our student and teacher networks are all CLIP-based, we coin the proposed method `TeachCLIP`.

In sum, our contributions are as follows. We propose TeachCLIP, letting a CLIP4Clip based network learn from more advanced yet computationally heavy T2VR models. TeachCLIP by design introduces no extra storage / computation overhead at the retrieval stage. Extensive experiments on multiple datasets, *i.e*. MSRVTT-1k [37], MSRVTT-3k [35], MSVD [1], VATEX [33], ActNetCap [10] and DiDeMo [11], justify the viability of the proposed method.

Table 1. **An overview of CLIP-based T2VR**. Visual backbone: ViT-B/32. Dataset: MSRVTT-1k. Note that we replicate existing models with their author-provided source code where applicable, so the numbers might differ (slightly) from the original papers.

Model	Per video-text (↓) matching (FLOPs)	Video feature (↓) storage (KB)	Video feature (↓) extraction (FLOPs)	R1 (↑)	SumR (↑)
CLIPPING[27]	0.5K	2	16.80G	40.7	–
CLIP4Clip[24]	0.5K	2	53.64G	42.8	195.5
PromptSwitch[4]	0.5K	2	59.28G	43.6	195.7
CenterCLIP[38]	1.5K	6	–	44.2	197.9
TS2-Net[22]	6.1K	24	54.27G	46.7	200.5
X-CLIP[25]	220.9K	26	53.64G	45.3	200.8
X-Pool[9]	275.0K	24	53.49G	46.0	201.5
DRL[32]	220.4K	26	53.64G	46.2	203.2
Cap4Video[34]	220.9K	28	–	**47.8**	**204.3**
TeachCLIP	0.5K	2	53.65G	46.8	203.7

2. Related Work

At a high level, our idea of multi-grained teaching can be viewed as knowledge distillation from a relatively heavy teacher network to an efficient student network. Hence, we briefly review progress in T2VR and knowledge distillation, accordingly interpreting our novelty in such a joint context.

2.1. Text-to-Video Retrieval

Effective T2VR. The majority of the literature is on effectiveness, aiming for better cross-modal matching networks that compute video-text similarity more accurately. Depending on whether video/text feature extractors are trained together with the cross-modal matching module, we categorize existing works into two groups. That is, feature re-learning methods and CLIP-based end-to-end methods.

Feature re-learning methods typically employ pre-trained 2D-CNNs [6, 18], 3D-CNNs [21, 26] or their combinations [8, 13] to obtain an initial feature representation of a given video. Similarly, a given text is encoded either by nontrainable bag-of-words [18], or by pre-trained text encoders including Word2Vec [5], BERT [28], GPT [3], *etc*. Feature re-learning is then performed to project the video and text features into a common latent space, wherein the video-text relevance can be measured in terms of their distance in the common space. While there is still room for improvement, *e.g*. by adding more features with novel feature fusion blocks [3, 8, 13], the performance of feature re-learning methods is largely bounded by the initial features.

The advent of CLIP [29] and its application in the video domain is reshaping the research landscape of T2VR. CLIP-based end-to-end methods have shown superior performance to their feature re-learning based predecessors on multiple public datasets [9, 22, 24, 25]. As an initial attempt in this line of research, CLIP4Clip [24] employs the visual encoder of CLIP to first extract a sequence of frame-level features. The frame features, updated by a stack of standard Transformer blocks, are averaged to produce a video-level feature. Such a video feature can be precomputed offline, while the storage footprint is linear w.r.t. the number of videos. Hence, a CLIP4Clip based T2VR system is efficient and compact. Follow-ups of CLIP4Clip, *e.g*. X-CLIP [25], TS2-Net [22], X-Pool [9] and Cap4Video [34], improve video-text matching by fine-grained cross-modal feature interaction and matching. Despite their better performance, local interaction means features used for cross-modal matching have to be computed online, while fine-grained matching results in substantial computation and storage overhead, see Tab. 1. This puts the scalability of the latest CLIP-based methods into question.

Efficient T2VR. Depending on how the term "efficiency" is interpreted, we see two lines of research: efficient video feature extraction [27, 38] and efficient video-text matching [4]. TeachCLIP belongs to the latter.

In order to accelerate video feature extraction, CenterCLIP [38] utilizes multi-segment token clustering to find the most representative tokens. Only these essential tokens will be forwarded through the entire Transformers, while those non-essential tokens will be dropped at a prespecified Transformer block. Alternatively, CLIPPING [27] reduces the inference cost by using a mobile-friendly ViT [30] trained by feature-level knowledge distillation from CLIP4Clip. Nonetheless, the video features extracted by CLIPPING are less effective than their CLIP4Clip counterparts, resulting in a clear drop in retrieval effectiveness.

For efficient video-text matching, PromptSwitch [4] introduces a prompt cube into the CLIP image encoder for iteratively incorporating global video semantics into frame-level features. The frame features are then averaged to produce the final video feature for video-text relevance estimation. The probe cube operation adds computation overhead for video feature extraction, with the number of FLOPs increased from 53.64G to 59.28G, see Tab. 1. By contrast, the proposed TeachCLIP leaves the CLIP image encoder as is, with nearly the same feature extraction cost as CLIP4Clip.

TeachCLIP is orthogonal to recent works that focus on improving their visual encoder by refining network architectures (proxy-guided attention [36] and decomposed spatial-temporal modules [20]), knowledge transfer from the image domain to the video domain by selective token alignment [17], and large-scale video-text pretraining [17, 36]. The stronger encoder will be naturally beneficial to the student and the teacher models used by TeachCLIP.

2.2. Knowledge Distillation

KD is to transfer "dark" knowledge from a large teacher model or ensemble of teacher models to a single, smaller student model, which can be practically deployed [12]. The form of the knowledge varies, which can be the output of the teacher [3, 16], its intermediate representations [27], or mutual relations of data examples [31]. In the context of video-to-video retrieval, DnS is proposed to distill knowledge, represented in the form of video-to-video relevance scores, from a big network to a smaller network [16].

As for T2VR, TeachText is developed to train a student network to mimic the averaged video-text similarity produced by an ensemble of models based on feature relearning [3]. Since TeachText focuses solely on video-level KD, how to impart fine-grained cross-modal knowledge to the student network remains unexplored. CrossKD [31] introduces a distillation loss that leverages the available structures of the video and caption domains, eliminating the need for an external teacher during training. TeachCLIP thus differs from CrossKD in motivation (external *vs* internal KD) and technology (multi-grained teaching *vs* additional loss).

3. Proposed TeachCLIP Method

3.1. Problem Setup

We are provided with two sorts of CLIP-based T2VR models. The first sort, using exclusively video-level features for video-text matching, is efficient in terms of storage footprint and retrieval speed. Meanwhile, the other sort, relying on fine-grained cross-modal matching, is more accurate than the former. Such an advantage, however, comes at the cost of substantially increased storage and computation overhead. Viewing the former as a student and the latter as a teacher, the proposed TeachCLIP method aims to improve the relatively weaker student with multi-grained dark knowledge from the teacher, as illustrated in Fig. 2. We opt for the teacher-student framework due to its high flexibility: Any advanced model can be used as a teacher model as long as it provides video-level and frame-level relevance scores w.r.t. a given textual query. In what follows, we detail the student network in Sec. 3.2, which will be trained by the proposed multi-grained teaching algorithm in Sec. 3.3.

3.2. The Student Network

Our student network is based on CLIP4Clip [24]. Given a video x represented by a sequence of m frames $\{f_1, \ldots, f_m\}$, CLIP4Clip first feeds the frames in parallel into the visual encoder of CLIP, *i.e.* a Vision Transformer (ViT), producing an array of frame-level features $\{v_1, \ldots, v_m\}$, sized $m \times d$. These features, appended with position encoding, then go through four stacked Transformer blocks for temporal modeling, resulting in m enhanced features $\{\phi_1, \phi_2, \ldots, \phi_m\}$. The video feature $\phi(x)$ is obtained by mean pooling over the enhanced features.

We improve CLIP4Clip by replacing its mean pooling layer with an Attentional frame-Feature Aggregation (AFA) block. Given $\{\phi_1, \phi_2, \ldots, \phi_m\}$ as its input, AFA is designed to produce an m-dimensional nonnegative weight vector $\{w_1, \ldots, w_m\}$, where w_i shall reflect the importance of frame f_i. More formally, the key data flow of the visual side of the student network is expressed as follows:

$$\begin{cases} \{f_1, \ldots, f_m\} & \leftarrow \text{video-to-frames}(x), \\ \{v_1, \ldots, v_m\} & \leftarrow \text{ViT}(\{f_1, \ldots, f_m\}), \\ \{\phi_1, \ldots, \phi_m\} & \leftarrow \text{Transformers} \times 4(\{v_1, \ldots, v_m\}), \\ \{w_1, \ldots, w_m\} & \leftarrow \text{AFA}(\{\phi_1, \ldots, \phi_m\}), \\ \phi(x) & \leftarrow \sum_{i=1}^{m} w_i \phi_i. \end{cases} \tag{1}$$

As illustrated in Fig. 2, we implement the AFA block with a linear layer of $d \times d$, followed by ReLU, another linear layer of $d \times 1$, and finally a softmax layer. As such, the amount of extra parameters introduced by AFA is $O(d^2)$. Such computational overhead is ignorable. Note that AFA is not novel by itself. Our innovation is to *supervise* the attention weights with frame-text relevance scores given by the teacher network, as we will describe shortly in Sec. 3.3.2.

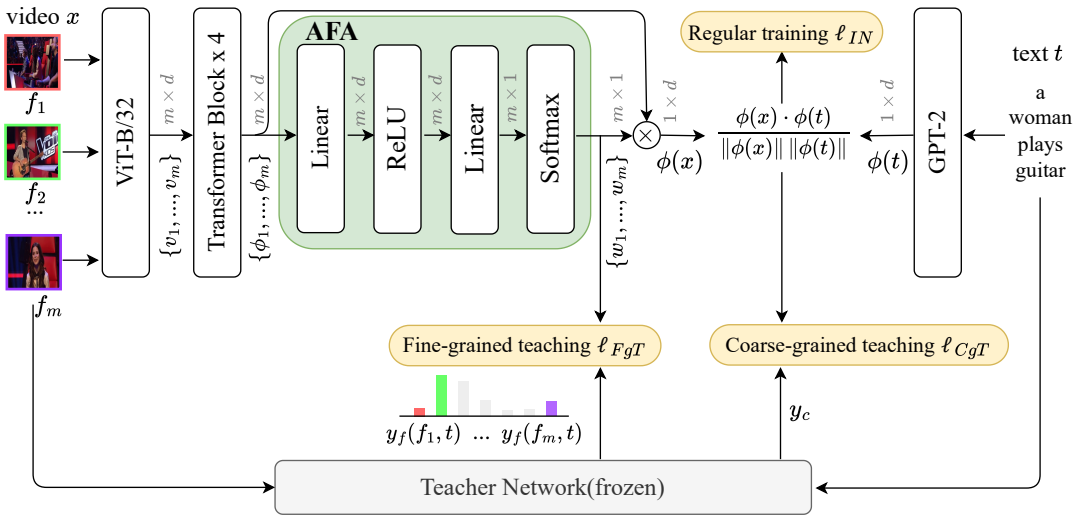

Figure 2. ***TeachCLIP* for training a CLIP4Clip based network for text-to-video retrieval**. We add to CLIP4Clip an **A**ttentional frame-**F**eature **A**ggregation (AFA) block. We term this variant CLIP4Clip-AFA. Given a sequence of m frame-level features as input, AFA outputs frame-specific weights $\{w_1, \ldots, w_m\}$, used to aggregate the frame-level features into a video-level feature. The weights are *supervised* by the frame-text relevance scores $\{y_f(f_1, t), \ldots, y_f(f_m, t)\}$ provided by a pre-trained (and frozen) teacher network. As such, AFA provides a simple yet effective channel for fine-grained knowledge distillation. End-to-end network training is conducted by jointly minimizing a coarse-grained teaching loss ℓ_{CgT}, a fine-grained teaching loss ℓ_{FgT} and a regular training loss ℓ_{IN}. CLIP4Clip-AFA trained by TeachCLIP has the same retrieval time and nearly the same GPU memory usage as CLIP4Clip, yet more effective.

AFA thus creates a simple yet effective channel to pass the fine-grained knowledge from the teacher on to the student.

3.3. Multi-grained Teaching

The proposed Multi-grained Teaching (MgT) algorithm follows a standard SGD procedure. In each training iteration, a mini-batch B consisting of b video-text pairs $\{(x_i, t_i) | i = 1, \ldots, b\}$ is randomly sampled from the given training dataset. Such randomness ensures that each video is irrelevant to other texts within the batch, and vice versa. For simplicity, we re-use B to denote a $b \times b$ video-text similarity matrix derived from the batch by the student network. Specifically, $B_{i,j}$ represents the cosine similarity between $\phi(x_i)$ and $\phi(t_j)$, where $i, j = 1, \ldots, b$. Accordingly, the i-th row $B_{i,\cdot}$ stores similarity scores of video x_i with respect to all b texts in the batch, while the j-th column $B_{\cdot,j}$ stores similarity scores of all b videos in relation to text t_j.

In order to perform MgT, we require two types of output from the teacher network concerning video x_i and text t_j. That is, a coarse-grained relevance score of x_i and t_j, denoted by $y_c(x_i, t_j)$, and a fine-grained relevance score for each frame, denoted by $y_f(f_{i,k}, t_j)$, where $k = 1, \ldots, m$. Note that both y_c and y_f are readily obtainable from the SOTA models such as X-CLIP and TS2-Net, which can compute relevance scores at both frame-level and video-level. Moreover, we assume that $\{y_f(f_{i,k}, t_j)\}$ have been adjusted using the softmax function.

3.3.1 Coarse-grained Teaching

Coarse-grained teaching is to supervise the student network with video-level soft labels predicted by the teacher network. To that end, TeachText [3] uses the element-wise Huber loss, enforcing $B_{i,j}$ to be close to $y_c(x_i, t_j)$. However, recent research on KD [14] suggests such type of loss is suboptimal, as enforcing a student model to replicate the output of a much stronger teacher model would unnecessarily increase the difficulty of KD and consequently impede transferring truly useful knowledge from the teacher to the student. Instead, [14] suggests minimizing Pearson's distance d_p (or equivalently maximizing Pearson correlation coefficient), which is known to be invariant under separate changes in scale and location in two given variables. We consider such invariance also desirable in the current task, as minimizing d_p between the output of the student and the teacher networks is adequate for the student to rank-wisely imitate the teacher. Following this thought, we opt to use d_p as the coarse-grained teaching loss ℓ_{CgT}. The loss for video x_i is computed as $d_p\big(\sigma(B_{i,\cdot}), \sigma(y_c(v_i, \cdot))\big)$, where σ is softmax. In a similar manner, the loss for text t_j is calculated as $d_p\big(\sigma(B_{\cdot,j}), \sigma(y_c(\cdot, t_j))\big)$. Accordingly, ℓ_{CgT} is defined as the following batch-level symmetric loss:

$$\ell_{CgT} := \frac{1}{b} \sum_{i=1}^{b} d_p\big(\sigma(B_{i,\cdot}), \sigma(y_c(v_i, \cdot))\big) + \frac{1}{b} \sum_{j=1}^{b} d_p\big(\sigma(B_{\cdot,j}), \sigma(y_c(\cdot, t_j))\big) \tag{2}$$

3.3.2 Fine-grained Teaching

Recall that at the visual side of the student network, we introduce a lightweight AFA block to produce m attention weights $\mathbf{w} = \{w_1, \ldots, w_m\}$ for a convex combination of the m temporally enhanced frame-level features $\{\phi_1, \ldots, \phi_m\}$. Instead of letting the network learn the weights all by itself, we propose to guide the weights-related learning process with fine-grained relevance information from the stronger teacher network. Intuitively, frames that are more relevant with respect to the given text should receive a higher weighting. Consequently, we calculate a fine-grained teaching loss, denoted as ℓ_{FgT}, for each relevant video-text pair (v_i, t_i). This is achieved by applying the cross-entropy (CE) loss between the assigned weights and the frame-text similarities provided by the teacher. A batch-level loss is then derived by averaging across all b relevant pairs within the specified batch, namely

$$\ell_{FgT} := -\frac{1}{b}\sum_{i=1}^{b}\sum_{k=1}^{m} y_f(f_{i,k}, t_i)\log w_{i,k}. \tag{3}$$

Notice that we do not consider fine-grained teaching on the textual side, *e.g.* exploiting video-word similarities to attentively aggregate word-level features. Our main concern is that in contrast to a keyframe that represents the video content to a large extent, a keyword token alone is largely insufficient to represent the corresponding sentence. Indeed, our preliminary experiment showed that adding an AFA on the textual side brings no improvement. We therefore did not go further in that direction. Following CLIP4Clip, we adopt CLIP's textual encoder, *i.e.* a GPT-2.

3.4. Training Algorithm

In addition to ℓ_{CgT} and ℓ_{FgT}, we calculate the symmetric InfoNCE loss over the similarity matrix B. Denoted as ℓ_{IN}, this loss is commonly used for training cross-modal matching networks [22, 24, 25, 38]. The student network is trained to minimize the sum of the three losses, *i.e.* $\ell_{CgT} + \ell_{FgT} + \ell_{IN}$, with the first two terms responsible for MgT and the last term for regular training. We combine the three losses equally due to their small disparity in the magnitudes. TeachCLIP is easy to implement, see Alg. 1.

Algorithm 1: TeachCLIP in a PyTorch style

```
Input: Training data loader D={(v, t)}
       Trained teacher network
Output: Trained student network

optimizer = torch.optim.Adam(student.parameters)
for e=1,2,..., MAX_EPOCHES:
    for mini-batch {(v, t)} in D:
        optimizer.zero_grad()
        y_c, y_f = teacher({(v, t)})
        B, w = student({(v, t)})
        l_CgT = pearson_distance_loss(y_c, B)
        l_FgT = cross_entropy_loss(y_f, w)
        l_IN = symmetric_InfoNCE_loss(B)
        loss = l_CgT + l_FgT + l_IN
        loss.backward()
        optimizer.step()
```

4. Experiments

4.1. Experimental Setup

Datasets. We adopt the following public datasets: MSRVTT[35], MSVD[1], VATEX[33], ActivityNet-Caption (ActNetCap) [10] and DiDeMo [11]. While the original data split of MSRVTT has nearly 3k test videos and 60k sentences, Yu *et al.* suggest another split of 9k videos for training and 1k video-text pairs for testing [37]. Probably due to the relatively smaller test-set size that makes the evaluation more efficient, the 1k edition (MSRVTT-1k) appears to be more popular than its 3k counterpart (MSRVTT-3k). We follow this practice, using MSRVTT-1k as the primary dataset for our ablation study.

For MSVD and DiDeMo, we use the official data split. For VATEX, we use the split by [2]. As for ActNetCap, we adopt the split by [8], testing on 'val1' as [22, 24, 25]. Different from the other datasets using a sentence as a query, ActNetCap and DiDeMo, with descriptions per video merged into a paragraph, essentially perform paragraph-to-video retrieval. See Tab. 2 for an overview.

Table 2. **Datasets**. MSRVTT-1k is used for ablation study.

Dataset	Training set		Validation set		Test set	
	#videos	#texts	#videos	#texts	#videos	#texts
MSRVTT-1k [37]	9,000	180,000	n.a.	n.a.	1,000	1,000
MSRVTT-3k [35]	6,513	130,260	497	9,940	2,990	59,800
MSVD [1]	1,200	48,774	100	4,290	670	27,763
VATEX [33]	25,991	259,910	1,500	15,000	1,500	15,000
ActNetCap [10]	10,009	10,009	n.a.	n.a.	4,917	4,917
DiDeMo [11]	8,392	8,392	1,065	1,065	1,004	1,004

Evaluation criteria. We report standard rank-based metrics, *i.e.* Recall at top k (k=1, 5, 10) and SumR (R1+R5+R10) as a combined metric.

Implementation details. Subject to our computation capacity (8 NVIDIA 3090 GPUs), the default setting is as follows, unless otherwise stated. We use ViT-B/32 as the visual encoder and GPT-2 as the textual encoder, initialized using OpenAI-released CLIP[1]. To prevent catastrophic forgetting, the initial learning rate is set in a module-specific manner: 1e-7 for ViT-B/32 and GPT-2, and 1e-4 for the remaining modules. Training lasts 10 epochs at maximum by an Adam optimizer [15], with the learning rate decayed by a cosine schedule strategy [23]. The input frame size is

[1] https://github.com/openai/CLIP

Table 3. **Performance of TeachCLIP with different teachers**. Visual backbone: ViT-B/32. Dataset: MSRVTT-1k.

Choice of Teacher(s)	R1	R5	R10	SumR
–	44.0	71.2	81.1	196.3
Single teacher:				
TS2-Net	45.6	72.1	81.7	199.4 (+3.1 ↑)
X-CLIP	45.2	72.3	82.3	199.8 (+3.5 ↑)
X-Pool	44.0	73.5	82.6	200.1 (+3.8 ↑)
DRL	45.9	73.9	82.3	202.1 (+5.8 ↑)
X-CLIP (ViT-B/16)	45.7	73.9	83.1	202.7 (+6.4 ↑)
Multiple teachers:				
X-CLIP & TS2-Net	46.5	72.7	83.0	202.2 (+5.9 ↑)
X-CLIP & DRL	45.8	73.9	**83.5**	203.2 (+6.9 ↑)
X-CLIP & X-Pool	**46.8**	74.3	82.6	203.7 (+7.4 ↑)
X-CLIP & TS2-Net & X-Pool	**46.8**	**74.9**	82.9	**204.6** (+8.3 ↑)

224×224. The maximum length of frame / word tokens is set to 12 and 32, respectively, with mini-batch size of 240. As ActNetCap and DiDeMo have much longer videos and texts, we use a larger maximum token length of 64 and a smaller batch size of 96. For MSRVTT-3k, MSVD, VATEX and DiDeMo, models maximizing R1 on the corresponding validation set are chosen. For MSRVTT-1k and ActNetCap without validation set, we follow [24, 25], reporting peaking performance on the test set.

4.2. Evaluating TeachCLIP

Recall that the essence of TeachCLIP is to let a stronger teacher network to teach a computationally efficient student network for better retrieval performance. Hence, TeachCLIP needs to be evaluated along multiple dimensions including the choice of the teacher, the choice of the student, and how the teaching process is executed.

Choice of the teacher. We have X-CLIP[25], TS2-Net [22], X-Pool [9] and DRL[32] in our shortlist, as they are open-source, provide both video-text and frame-text similarity scores, and report competitive performance.

The performance of TeachCLIP with different teachers is shown in Tab. 3. TeachCLIP consistently outperforms the baseline, *i.e*. the student network trained alone, with the gain of SumR ranging from 3.1 to 6.4, subject to the specific teacher in use. We also try a multi-teacher teaching strategy, where ℓ_{CgT} and ℓ_{FgT} are computed per teacher and minimized jointly. Compared to the single-teacher counterparts, even better performance is obtained with the maximum gain of 8.3 in SumR. In both single-teacher and multi-teacher settings, TeachCLIP improves over the baseline, with no extra computation / storage overhead in the retrieval stage.

Our experiments show that X-CLIP is stably reproducible on varied datasets. Hence, this model is used as the teacher in the rest of our ablation study.

Choice of the student network. As Tab. 4 shows, TeachCLIP w/o teaching (row#3) is better than CLIP4Clip w/o teaching (row#1), showing that the AFA block is helpful even under regular training. Given X-CLIP as the teacher, TeachCLIP remains better than CLIP4Clip (199.8 *vs* 196.5 in SumR). Furthermore, we experiment with a stronger backbone, substituting ViT-B/16 for ViT-B/32 as the visual encoder. TeachCLIP is again effective, lifting SumR from 201.5 to 207.4. Our choice of using CLIP4Clip-AFA as the student network is verified.

Table 4. **Performance of different students**.

Student	Teacher	R1	R5	SumR
CLIP4Clip	–	42.8	71.6	195.5
CLIP4Clip	X-CLIP	44.2	71.7	196.5 (+1.0 ↑)
CLIP4Clip-AFA	–	44.0	71.2	196.3
CLIP4Clip-AFA	X-CLIP	45.2	72.3	199.8 (+3.5 ↑)
CLIP4Clip-AFA	X-CLIP(ViT-B/16)	45.7	73.9	202.7 (+6.4 ↑)
CLIP4Clip-AFA(ViT-B/16)	–	46.3	72.8	201.5
CLIP4Clip-AFA(ViT-B/16)	X-CLIP(ViT-B/16)	**48.0**	**75.9**	**207.4** (+5.9 ↑)

Table 5. **Performance of TeachCLIP with different losses**. Teacher: X-CLIP. Dataset: MSRVTT-1k.

Loss configuration	R1	R5	R10	SumR
0: InfoNCE	44.0	71.2	81.1	196.3
1: + FgT(Huber as ℓ_{FgT})	42.5	72.6	81.8	196.9 (+0.6 ↑)
2: + FgT(Pearson as ℓ_{FgT})	44.1	72.1	81.0	197.2 (+0.9 ↑)
3: + FgT(CE as ℓ_{FgT})	44.1	71.2	82.0	197.3 (+1.0 ↑)
4: + CgT(Huber as ℓ_{CgT})	44.7	71.1	82.0	197.8 (+1.5 ↑)
5: + CgT(Pearson as ℓ_{CgT})	44.4	71.0	**82.6**	198.0 (+1.7 ↑)
6: + MgT(Pearson as ℓ_{CgT}, Huber as ℓ_{FgT})	**45.6**	71.9	81.8	199.3 (+3.0 ↑)
7: + MgT(Pearson as ℓ_{CgT}, Pearson as ℓ_{FgT})	45.4	**73.0**	81.3	199.7 (+3.4 ↑)
8: + MgT(Pearson as ℓ_{CgT}, CE as ℓ_{FgT})	45.2	72.3	82.3	**199.8** (+3.5 ↑)

Which loss for CgT? Comparing Setup#5 (the Pearson distance loss) and Setup#4 (the Huber loss as in TeachText) in Tab. 5, the former is marginally better (198.0 versus 197.8). Our experiments on other datasets also show that the Pearson loss consistently outperforms the Huber loss.

Is MgT necessary? Setup#3, Setup#5 and Setup#8 in Tab. 5 are constructed by adding FgT, CgT and MgT separately to the standard InfoNCE loss. MgT has SumR of 199.8, followed by CgT (198.0) and FgT (197.3). The necessity of MgT is verified.

Which loss for FgT? We compare CE with two alternatives, namely Huber and Pearson, see Setup#1/#2 w/o CgT and Setup#6/#7 w/ CgT. CE is the best loss for FgT.

Cross-data evaluation. To check if TeachCLIP merely fits weights specific to each dataset, we test the models trained on MSRVTT-1k directly on the other datasets. As shown in Tab. 6, TeachCLIP is consistently better than CLIP4Clip on all datasets, indicating that TeachCLIP does not learn dataset-specific weights. Moreover, the lower cross-dataset performance of CLIP4Clip-AFA as compared to CLIP4Clip shows that learning the adaptive weights by the student itself does not generalize.

Qualitative analysis. Some qualitative results are given in Fig. 3. Consider the result at the top for instance. The first frame is the most salient, as it shows key objects, *i.e*. a

Table 6. **Cross-dataset results**. Training data: MSRVTT-1k.

Model	MSVD	VATEX	ActNetCap	DiDeMo	Mean
CLIP4Clip-AFA	**201.4**	217.2	152.8	149.6	180.3
CLIP4Clip	197.5	217.7	155.7	153.8	181.2
TeachCLIP	199.5	**220.0**	158.9	156.2	183.7
X-CLIP	200.0	218.1	**159.5**	**159.5**	**184.3**

girl wearing a blue dress and a man in a black shirt, specified in the query. By computing frame-text relevance on the fly, X-CLIP successfully identifies this frame. For this frame, TeachCLIP also gives a larger weight, albeit precomputed. Similar results can be observed in the other two examples. These results further confirm the viability of TeachCLIP.

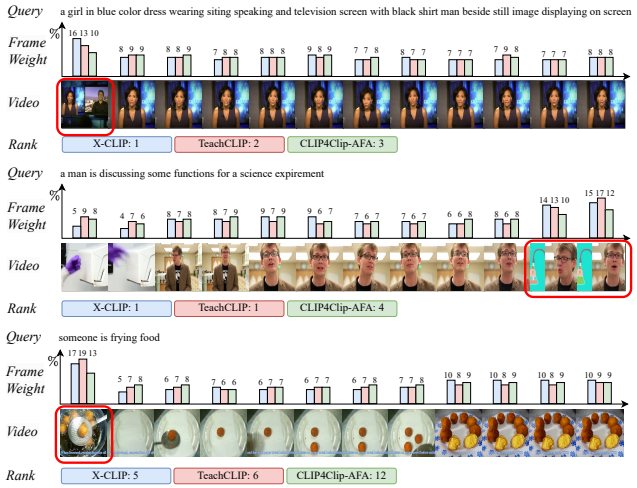

Figure 3. **Visualization of frame weights given by X-CLIP, CLIP4Clip-AFA and TeachCLIP**. The weights by TeachCLIP are closer to the query-dependent weights by X-CLIP, especially on salient frames (manually marked out by red rectangles).

4.3. Comparison with Existing Methods

Baselines. Both feature re-learning based methods and CLIP-based end-to-end methods are compared. For the purpose of reproducible research, we include the following open-sourced methods:

• *Feature re-learning*: W2VV++[18], DualE[6], CE[21], SEA[19], MMT[8], TeachText[3], and LAFF[13].

• *CLIP-based end-to-end*: Besides CLIP4Clip [24], X-CLIP [25], TS2-Net [22], X-Pool [9] and DRL [32] that have been used in our ablation study, we include CenterCLIP[38], PromptSwitch[4], Cap4Video [34], STAN [20], CLIP-ViP [36] and UMT [17].

Note that Cap4Video leverages a large vision-language model to enrich video descriptions, whilst CLIP-ViP and UMT are pretrained on large-scale video-text pairs, making the comparison somewhat unfair for the other models. We include these models mainly to show the latest progress.

Among the end-to-end methods, not all of them have results available, except for CLIP4Clip, TS2-Net, and X-CLIP, see Tab. 7. Moreover, our experiments show that the performance of TS2-Net is somewhat unstable: better than CLIP4Clip on MSRVTT-1k (SumR 200.5 *vs* 195.5) and MSRVTT-3k (153.6 *vs* 150.1), yet worse on MSVD (204.9 *vs* 206.6) and ActNetCap (190.4 *vs* 194.1). Considering the availability of the teacher models and their performance per dataset, we report TeachCLIP jointly taught by X-CLIP & X-Pool on MSRVTT-1k. For the other datasets, we report TeachCLIP taught by X-CLIP.

Effectiveness comparison. As shown in Tab. 7, the feature re-learning based methods (the top part of Tab. 7) are clearly worse than the CLIP-based end-to-end methods (the bottom part). The former can be much improved by adding pre-extracted CLIP features (the middle part, cited from [13]). Consider MMT, for instance. The inclusion of the CLIP feature brings in a clear improvement (SumR 145.7 $\rightarrow$ 186.1 on MSRVTT-1k). Nonetheless, they remain inferior to the end-to-end methods.

As for the end-to-end methods, there is no clear winner that tops the performance on every dataset. DRL is the best on MSRVTT-1k, followed by STAN and X-Pool. CenterCLIP is the best on ActNetCap, followed by X-CLIP and CLIP4Clip. X-CLIP leads on MSRVTT-3k and MSVD. Our evaluation indicates that X-CLIP has the best overall performance, with mean SumR of 203.4.

TeachCLIP outperforms X-CLIP on MSRVTT-1k. For the other datasets, the performance gap between CLIP4Clip and X-CLIP, measured in terms of the absolute difference of SumR, is as follows: MSRVTT-3k 6.6, MSVD 3.5, VATEX 0.0, ActNetCap 10.0 and DiDeMo 11.0. TeachCLIP, as efficient as CLIP4Clip, reduces the above gap as follows: MSRVTT-3k 6.6 $\rightarrow$ 0.7, ActNetCap 10.0 $\rightarrow$ 4.0, DiDeMo 11.0 $\rightarrow$ 4.2. Even more, TeachCLIP (marginally) surpasses X-CLIP on MSVD (210.2 *vs* 210.1) and VATEX (251.6 *vs* 248.5). While one would normally not expect the student to beat the teacher, our interpretation of this counter-intuitive result is as follows. When the teacher and the student have distinct network structures yet with relatively close performance, as in the cases of MSVD and VATEX, the teacher may provide complementary information that the student cannot learn by itself. As such, MgT has an effect on ensemble learning to train a better model. TeachCLIP, with mean SumR of 202.9, is almost comparable to X-CLIP.

Storage comparison. Given a 512-d feature vector per video and 4 bytes per floating point, CLIP4Clip, PromptSwitch, and TeachCLIP have the smallest storage footprint of 2KB per video, see Tab. 1.

Efficiency comparison. We assess the number of FLOPs[2] required for pre video-text matching. As Tab. 1 shows, CLIP4Clip, PromptSwitch, and TeachCLIP are the most efficient. We also assess the cost of video fea-

[2]https://github.com/sovrasov/flops-counter.pytorch

Table 7. **T2VR Performance of different methods on multiple datasets**. Note that we replicate existing methods with their author-provided source code where applicable, so the numbers might differ (slightly) from their original papers. We de-emphasize Cap4Video, CLIP-ViP* and UMT *using gray and italic font* as Cap4Video resorts to an external large vision-language model to generate video captions for retrieval, while CLIP-ViP* and UMT are pretrained on large-scale video-text pairs.

Model	MSRVTT-1k			MSRVTT-3k			MSVD			VATEX			ActNetCap			DiDeMo			Mean
	R1	R5	SumR	R1	R5	SumR	R1	R5	SumR	R1	R5	SumR	R1	R5	SumR	R1	R5	SumR	
Feature re-learning w/o CLIP feature:																			
W2VV++ [18]	18.9	45.3	121.7	11.1	29.6	81.2	22.4	51.6	138.8	–	–	–	–	–	–	–	–	–	–
DualE [6]	21.1	48.7	130.0	11.6	30.3	83.2	–	–	–	36.8	73.6	194.1	–	–	–	–	–	–	–
CE [21]	20.9	48.8	132.1	10.0	29.0	80.2	19.8	49.0	132.6	–	–	–	17.7	46.6	–	–	–	–	–
SEA [19]	23.8	50.3	137.9	13.1	33.4	91.5	24.6	55.0	147.5	–	–	–	–	–	–	–	–	–	–
MMT [8]	24.6	54.0	145.7	–	–	–	–	–	–	–	–	–	22.7	54.2	–	–	–	–	–
TeachText [3]	29.6	61.6	165.4	15.0	38.5	105.2	25.4	56.9	153.6	53.2	87.4	233.9	23.5	57.2	–	–	–	–	–
Feature re-learning with CLIP feature:																			
SEA	37.2	67.1	182.6	19.9	44.3	120.7	34.5	68.8	183.8	52.4	90.2	238.5	–	–	–	–	–	–	–
W2VV++	39.4	68.1	185.6	23.0	49.0	132.7	37.8	71.0	190.4	55.8	91.2	243.0	–	–	–	–	–	–	–
MMT	39.5	68.3	186.1	24.9	50.5	137.4	40.6	72.0	194.3	54.4	89.2	238.6	–	–	–	–	–	–	–
LAFF [13]	45.8	71.5	199.3	29.1	54.9	149.8	45.4	70.6	200.6	59.1	91.7	247.1	–	–	–	–	–	–	–
CLIP-based end-to-end (visual backbone: ViT-B/32):																			
CenterCLIP [38]	44.2	71.6	197.9	–	–	–	47.3	76.8	209.7	–	–	–	43.9	**74.6**	**204.3**	–	–	–	–
CLIP4Clip [24]	42.8	71.6	195.5	29.4	54.9	150.1	45.6	76.1	206.6	61.6	91.1	248.5	39.7	71.0	194.1	42.0	69.0	189.2	197.3
TS2-Net [22]	46.7	72.6	200.5	29.9	56.4	153.6	44.6	75.8	204.9	61.1	91.5	248.6	37.3	69.9	190.4	40.2	69.4	188.4	197.7
X-CLIP [25]	45.3	73.7	200.8	**31.2**	**57.4**	**156.7**	47.2	77.0	210.1	62.2	90.9	248.5	**44.4**	**74.6**	204.1	45.0	73.1	200.2	**203.4**
X-Pool [9]	46.0	72.8	201.5	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
DRL [32]	46.2	74.0	203.2	–	–	–	–	–	–	–	–	–	–	–	–	**47.9**	**73.8**	**204.4**	–
PromptSwitch [4]	43.6	71.5	195.7	–	–	–	46.3	75.8	206.6	–	–	–	–	–	–	–	–	–	-
CLIP-ViP [36]	46.5	72.1	201.1	–	–	–	–	–	–	–	–	–	–	–	–	40.6	70.4	190.3	–
STAN [20]	**46.9**	72.8	202.5	–	–	–	–	–	–	–	–	–	–	–	–	46.5	71.5	198.9	–
Cap4Video [34]	*47.8*	*73.8*	*204.3*	–	–	–	–	–	–	–	–	–	–	–	–	*52.0*	*79.4*	*218.9*	–
*CLIP-ViP**	*50.1*	*74.8*	*209.5*	–	–	–	–	–	–	–	–	–	–	–	–	*48.6*	*77.1*	*210.1*	–
UMT [17]	*51.0*	*76.5*	*211.7*	–	–	–	*71.9*	*94.5*	*264.2*	–	–	–	*58.3*	*83.9*	*233.7*	*61.6*	*86.8*	*239.9*	–
TeachCLIP	46.8	**74.3**	**203.7**	30.9	57.1	156.0	**47.4**	**77.3**	**210.2**	**63.6**	**91.9**	**251.6**	42.2	72.7	200.1	43.7	71.2	196.0	202.9

ture extraction. The computational demands, in terms of FLOPs per video, are 53.64G for CLIP4Clip, 59.28G for PromptSwitch, and 53.65G for TeachCLIP. Note that the use of the frozen teacher (X-CLIP) produces extra computation overhead during training. Given 12 frames per video and batch size of 120, per batch the GPU memory increases from 40.0GB to 42.7GB, while the forward / backward computation increases from 0.9s to 1.2s. As training is done offline, we consider such overhead affordable.

5. Conclusions and Remarks

We propose TeachCLIP with multi-grained teaching (MgT) for efficient text-to-video retrieval (T2VR). Extensive experiments on multiple public datasets allow us to conclude as follows. While coarse-grained teaching and fine-grained teaching are helpful even when used separately, their joint use, namely MgT, is the best. TeachCLIP has the same efficiency and compactness as CLIP4Clip, yet has near-SOTA effectiveness. Our work provides a new outlook on the practical use of fine-grained T2VR models deemed to be useful but inefficient in real-world applications.

Limitation of the current study. The conclusion that holistic features are almost sufficient are based on our experimental data, most of which are short video clips in 10 seconds. Subject to our computation power, the majority of the experiments are conducted with ViT-B/32 as the visual backbone. While using ViT-B/16 yields a larger gain on MSRVTT-1k, see Tab. 4, the benefit of using a stronger backbone on the other datasets needs further studying.

Acknowledgments. This research was supported by NSFC (No. 62172420) and Tencent Marketing Solution Rhino-Bird Focused Research Program. We thank H. Hu, R. Xie, F. Lian and Z. Kang from Tencent for helpful discussion on the topic.

References

[1] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, pages 190–200, 2011. 1, 2, 5

[2] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *CVPR*, pages 10638–10647, 2020. 5

[3] Ioana Croitoru, Simion-Vlad Bogolin, Marius Leordeanu, Hailin Jin, Andrew Zisserman, Samuel Albanie, and Yang Liu. TeachText: Crossmodal generalized distillation for text-video retrieval. In *ICCV*, pages 11583–11593, 2021. 2, 3, 4, 7, 8

[4] Chaorui Deng, Qi Chen, Pengda Qin, Da Chen, and Qi Wu. Prompt Switch: Efficient CLIP adaptation for text-video retrieval. In *ICCV*, pages 15648–15658, 2023. 1, 2, 3, 7, 8

[5] Jianfeng Dong, Xirong Li, and Cees G. M. Snoek. Predicting visual features from text for image and video caption retrieval. *TMM*, 20(12):3377–3388, 2018. 2

[6] Jianfeng Dong, Xirong Li, Chaoxi Xu, Xun Yang, Gang Yang, Xun Wang, and Meng Wang. Dual encoding for video retrieval by text. *TPAMI*, 44(8):4065–4080, 2021. 1, 2, 7, 8

[7] Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. CLIP2Video: Mastering video-text retrieval via image CLIP. *Arxiv*, 2021. 1

[8] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *ECCV*, pages 214–229, 2020. 2, 5, 7, 8

[9] 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*, pages 5006–5015, 2022. 1, 2, 6, 7, 8

[10] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015. 2, 5

[11] 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

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv*, 2015. 2, 3

[13] 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*, pages 444–461, 2022. 2, 7, 8

[14] Tao Huang, Shan You, Fei Wang, Chen Qian, and Chang Xu. Knowledge distillation from a stronger teacher. In *NeurIPS*, 2022. 4

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv*, 2014. 5

[16] Giorgos Kordopatis-Zilos, Christos Tzelepis, Symeon Papadopoulos, Ioannis Kompatsiaris, and Ioannis Patras. DnS: Distill-and-select for efficient and accurate video indexing and retrieval. *IJCV*, 130(10):2385–2407, 2022. 3

[17] 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. 3, 7, 8

[18] Xirong Li, Chaoxi Xu, Gang Yang, Zhineng Chen, and Jianfeng Dong. W2VV++: Fully deep learning for ad-hoc video search. In *ACMMM*, pages 1786–1794, 2019. 2, 7, 8

[19] Xirong Li, Fangming Zhou, Chaoxi Xu, Jiaqi Ji, and Gang Yang. SEA: Sentence encoder assembly for video retrieval by textual queries. *TMM*, 23:4351–4362, 2021. 7, 8

[20] 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. 3, 7, 8

[21] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts. In *BMVC*, page 279, 2019. 2, 7, 8

[22] Yuqi Liu, Pengfei Xiong, Luhui Xu, Shengming Cao, and Qin Jin. TS2-Net: Token shift and selection transformer for text-video retrieval. In *ECCV*, pages 319–335, 2022. 1, 2, 5, 6, 7, 8

[23] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 5

[24] 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*, 508:293–304, 2022. 1, 2, 3, 5, 6, 7, 8

[25] 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 *ACMMM*, pages 638–647, 2022. 1, 2, 5, 6, 7, 8

[26] 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*, pages 19–27, 2018. 2

[27] 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*, pages 18983–18992, 2023. 2, 3

[28] Ladislav Peška, Gregor Kovalčík, Tomáš Souček, Vít Škrhák, and Jakub Lokoč. W2VV++ BERT model at VBS 2021. In *MMM*, 2021. 2

[29] 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*, pages 8748–8763, 2021. 1, 2

[30] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018. 3

[31] Vinh Tran, Niranjan Balasubramanian, and Minh Hoai. From within to between: Knowledge distillation for cross modality retrieval. In *ACCV*, pages 3223–3240, 2022. 3

[32] Qiang Wang, Yanhao Zhang, Yun Zheng, Pan Pan, and Xian-Sheng Hua. Disentangled representation learning for text-video retrieval. *arXiv*, 2022. 2, 6, 7, 8

[33] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. VATEX: A large-scale, high-quality multilingual dataset for video-and-language research. In *ICCV*, pages 4581–4591, 2019. 2, 5

[34] Wenhao Wu, Haipeng Luo, Bo Fang, Jingdong Wang, and Wanli Ouyang. Cap4video: What can auxiliary captions do for text-video retrieval? In *CVPR*, 2023. 2, 7, 8

[35] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*, pages 5288–5296, 2016. 1, 2, 5

[36] 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. 1, 3, 7, 8

[37] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *ECCV*, pages 471–487, 2018. 1, 2, 5

[38] Shuai Zhao, Linchao Zhu, Xiaohan Wang, and Yi Yang. CenterCLIP: Token clustering for efficient text-video retrieval. In *SIGIR*, page 970–981, 2022. 2, 3, 5, 7, 8