Dual Learning with Dynamic Knowledge Distillation for Partially Relevant Video Retrieval

Jianfeng Dong¹,², Minsong Zhang¹, Zheng Zhang¹, Xianke Chen¹, Daizong Liu³, Xiaoye Qu⁴, Xun Wang¹,², Baolong Liu¹,² †
¹Zhejiang Gongshang University, ²Zhejiang Key Lab of E-Commerce
³Peking University, ⁴Huazhong University of Science and Technology
https://github.com/HuiGuanLab/DL-DKD

Abstract

Almost all previous text-to-video retrieval works assume that videos are pre-trimmed with short durations. However, in practice, videos are generally untrimmed containing much background content. In this work, we investigate the more practical but challenging Partially Relevant Video Retrieval (PRVR) task, which aims to retrieve partially relevant untrimmed videos with the query input. Particularly, we propose to address PRVR from a new perspective, i.e., distilling the generalization knowledge from the large-scale vision-language pre-trained model and transferring it to a task-specific PRVR network. To be specific, we introduce a Dual Learning framework with Dynamic Knowledge Distillation (DL-DKD), which exploits the knowledge of a large vision-language model as the teacher to guide a student model. During the knowledge distillation, an inheritance student branch is devised to absorb the knowledge from the teacher model. Considering that the large model may be of mediocre performance due to the domain gaps, we further develop an exploration student branch to take the benefits of task-specific information. In addition, a dynamical knowledge distillation strategy is further devised to adjust the effect of each student branch learning during the training. Experiment results demonstrate that our proposed model achieves state-of-the-art performance on ActivityNet and TVR datasets for PRVR.

1. Introduction

With the explosion of online videos, searching the videos of interest has been an indispensable activity in people’s daily lives. Meanwhile, text-to-video retrieval (T2VR), retrieving videos w.r.t. a textual query from a large number of unlabeled videos, attracts growing attention recently [1, 6, 25, 35, 50, 59]. One basic assumption prerequisite for mainstream T2VR is that videos are pre-trimmed with short duration and supposed to be fully relevant to the query [29, 48, 61]. However, in practical applications, the majority of the existing videos are untrimmed. Besides, as queries are not known a priori, pre-trimmed video clips may not contain sufficient content to fully meet the query. Therefore, there is a huge gap between the literature and the real world for the mainstream T2VR task [8].

To fill the gap, a new text-to-video retrieval subtask, i.e., Partially Relevant Video Retrieval (PRVR), has been proposed recently in [8]. Different from previous T2VR, PRVR aims to retrieve the partially relevant untrimmed videos that contain at least one internal moment relevant to the given

![Query: A man opens the door and enters the room.](image)

Figure 1. (a) An illustrative example of the PRVR task; (b) Mean values of video duration of different datasets; (c) Performance comparisons between the state-of-the-art (SOTA) [8] and the vanilla CLIP-based [2] methods, where CLIP shows huge performance divergences for different PRVR datasets.
query (as exemplified in Fig. 1(a)). Besides, videos used for PRVR are much longer than that for T2VR (see Fig. 1(b)). In this work, we target the PRVR task, considering it is more consistent with practical video retrieval scenarios.

Recently, we notice an increasing use of large-scale pre-trained vision and language models, e.g., Contrastive Language-Image Pre-training (CLIP) [46], for various cross-modal tasks, recognition [49, 53], semantic segmentation [55, 65] and person Re-identification [20, 56], such as text-image retrieval [19, 47], visual question answer [4, 13], and achieving dominant performances. For text-to-video retrieval task, current works [2, 15, 36, 39] mainly focus on the learning of temporal aggregation layers on top of CLIP features, this is because the videos are mainly composed of image sequences while CLIP is trained only on image-text pairs. Different from short videos in these works, the PRVR task contains more complicated untrimmed videos with longer-duration moments of mixed query-relevant and query-irrelevant activities. Therefore, directly treating PRVR as mainstream text-to-video retrieval and aggregating the CLIP features across all the frames may lead to huge performance divergences on PRVR datasets. As shown in Fig. 1 (c), a vanilla CLIP performs superior on ActivityNet but depressing on TVR. Therefore, how to effectively transfer the knowledge of CLIP to PRVR models is still an open problem.

To this end, we propose a Dual Learning framework with Dynamic Knowledge Distillation (DL-DKD) to purify the knowledge of CLIP into the PRVR. Specifically, we develop an effective teacher-student network where the CLIP model is adopted as the teacher and a dual-branch student model is devised to acquire the knowledge. The reason why we introduce two student branches is that CLIP may suffer from domain gap issues due to complicated datasets. Therefore, one inheritance student branch is introduced to directly absorb the beneficial knowledge of the teacher model on a specific domain, while another exploration student branch is utilized to only explore the task-specific property of the training data. In addition, motivated by the fact that human beings first learn from teachers and slowly carry out self-living evolutionary learning once they have formed their own preliminary cognition. Thus, a dynamic knowledge distillation strategy is devised, namely, the inheritance branch takes the prime position at the beginning and the exploration branch gradually becomes more prominent during the training process. In this manner, our DL-DKD is able to take the advantage of both the powerful generalization-ability of CLIP and the benefits of task-specific model convergence on the PRVR data while alleviating their limitations, achieving more robust and effective retrieval. To sum up, the contributions of this work are threefold:

- We propose a knowledge distillation framework that contains a dual-branch student network to acquire appropriate knowledge selectively for partially relevant video retrieval. Meanwhile, our framework supports single-teacher and multiple-teacher distillation.
- We explore how to take the advantage of the powerful generalization-ability of the large model and the benefits of the task-specific model simultaneously while alleviating their limitations, and propose a dynamical knowledge distillation strategy.
- Extensive experiments demonstrate the effectiveness of the above contributions, and our proposed model achieves state-of-the-art performance on the challenging PRVR task.

2. Related Work

2.1. Text-to-video Retrieval

Given a textual query, the task of T2VR [1, 25, 32, 34, 35, 58, 61] aims to retrieve relevant videos with the query from a set of pre-trimmed video clips. The dominant methods typically project videos and queries into a common space for measuring the cross-modal similarity [17, 18, 35, 52]. They usually learn the cross-modal similarity using a large amount of video-text pairs, based on the initial video features extracted by pre-trained vision models and the text features obtained by pre-trained language models. Additionally, we observe an increasing use of large-scale pre-training vision-language models, such as CLIP [46], for text-to-video retrieval [2, 15, 23, 36, 39]. For instance, Hu et al. [23] utilize CLIP to extract both video and text features as extra features. Other works adapt CLIP for text-to-video retrieval by introducing the similarity calculation module between the representation of text and video frames [39], frame-wise attentions [2], and a temporal difference block for capturing motions between frames [15].

In practice, videos are generally untrimmed containing much background content [24, 45, 54, 60, 64]. However, in the traditional T2VR, videos are typically pre-trimmed with short duration and are supposed to be fully relevant to the query [29–31, 33, 48, 61], which leads to a huge gap between the literature and the real world. To overcome this limitation, a new text-to-video retrieval subtask, i.e., PRVR, has been proposed [8]. By contrast, videos in PRVR are typically untrimmed, and it aims to retrieve partially relevant videos with the query. An untrimmed video is considered to be partially relevant to a given textual query if it contains a moment relevant to the query. Although it is more consistent with real applications, PRVR had been neglected for a long time.

2.2. Knowledge Distillation

Knowledge distillation is the process of transferring knowledge from a large model (teacher) to a smaller one
3. Method

We propose a dual learning framework with dynamic knowledge distillation for the PRVR task, which exploits the knowledge of CLIP as the teacher guidance to dynamically distill and balance the learning of dual student branches. As shown in Fig. 2, the whole architecture mainly consists of two parts: 1) Teacher model: To leverage the powerful generalization ability of the large model trained on the large-scale data, a vision-language pre-training model like CLIP [46] is taken to serve as the teacher model to guide the task-specific model converge on the PRVR data. 2) Student model: Since the vanilla pre-trained large models may have huge performance differences on different datasets due to their task-specific domain gaps, a single-branch student model may simply get stuck into the underfitting problem by the above teacher model. To learn to selectively acquire appropriate knowledge from the teacher model when it performs differently on various data, we propose a two-branch, i.e., Inheritance-Exploration student network. This student model is split into two branches, in which one branch is utilized to inherit the beneficial knowledge of the teacher model on a specific data domain, while the other is to explore and fit the domain-specific property on the training data. A dynamical knowledge distillation strategy is further applied on the teacher-student framework to adjust the effect of each student branch learning during the training. In the following, we will illustrate the details of each component.

3.1. CLIP Teacher Model

The large-scale vision-language model CLIP [46] was pre-trained on a great amount of image-text data, and is now commonly employed as a strong vision-language backbone enabling zero-shot knowledge transfer to various downstream tasks [36, 51]. Therefore, we also resort to CLIP, utilizing it as the teacher model to appropriately guide our student model training. Note that other vision-language pre-training models, such as TCL [39], can also be employed here. We conduct experiments with different teacher models in Section 4.4, showing the remarkable generalizability of our proposed framework.

As depicted in the top of Fig. 2, given a video-text pair \((V, Q)\) consisting of a video \(V = \{I_k\}_{k=1}^K\) of \(K\) frames and a textual query \(Q\) as input, we feed them into the CLIP’s image and text encoders to obtain the corresponding video feature \(F^t = \{f^t_k\}_{k=1}^K \in \mathbb{R}^{d \times K}\) and query feature \(q^t \in \mathbb{R}^d\), respectively. The video feature is comprised
of a sequence of $k$ frame features, and the dimensions of the frame and the query features are both $d$. Considering that semantic-aligned similarity matters a lot in our retrieval task, we aim to transfer the collected knowledge of video-query semantic-aware similarity distribution from the teacher model to the student model. Formally, for a pair of video $V$ and query $Q$, their semantic similarity distribution $C^s \in \mathbb{R}^k$ is formulated as:

$$C^s = [\cos(f^s_1, q^s), \cos(f^s_2, q^s), \ldots, \cos(f^s_k, q^s)],$$

where $\cos$ denotes the cosine similarity.

Different from many works [21, 40] devoting to distilling knowledge from the semantic similarity distribution of the teacher model, our teacher model is committed to guide the student model by constraining the consistent semantic similarity distributions between the teacher-student models.

### 3.2. Dual-branch Student Model

To inherit the beneficial knowledge of the teacher model on a specific data domain while learning to explore and fit the domain-specific property on the training data, we develop a dual-branch student model. As illustrated in Fig. 2, the student model contains two branches, i.e., an inheritance branch and an exploration branch. Specifically, the inheritance student branch is devised to absorb the large-scale knowledge from the teacher branch. Besides, the exploration student branch is introduced to learn the data-specific property by fitting the training data to alleviate the teacher’s performance-drop problem due to the domain gaps. By jointly training the two student branches, we can obtain prime performance not only on datasets with a similar distribution to the training data of CLIP but also on datasets with distinct domain gaps.

#### 3.2.1 Inheritance Student Branch

As for the inheritance branch, it is expected to learn the collected knowledge from the semantic similarity distribution $C^s$ of the teacher model.

**Multi-Modal Encoding.** Given an input video $V = \{I_i\}_{i=1}^{n_v}$, a pre-trained 2D CNN with an FC layer is employed to extract the higher-level CNN features of the video as $F^v = \{f^v_i\}_{i=1}^{n_v} \in \mathbb{R}^{2 \times k}$, where each video frame is represented as a $z$-dimensional feature vector. Then, after an operation of a standard Transformer with positional embedding and another FC layer, $F^v$ is projected into the joint latent space for the latter multi-modal similarity measurement. This encoded visual feature $F^s = \{f^s_1, f^s_2, \ldots, f^s_k\} = FC(Trans(FC(F^v) + PE))$, where $PE$ stands for positional embedding, and Trans is a standard Transformer.

For an input query $Q$, following [8, 27], we utilize the pre-trained RoBERTa [38] with an FC layer to generate the word-level features $Q^v = \{w^v_i\}_{i=1}^{n_q} \in \mathbb{R}^{2 \times n_q}$. To further obtain the contextual features of the query text, we first feed $Q^v$ into a standard Transformer to obtain $Q^s = \{w^s_i\}_{i=1}^{n_q} \in \mathbb{R}^{2 \times n_q}$, and then employ an attention layer to generate the sentence-level feature $q^s \in \mathbb{R}^2$ via attentive aggregation as:

$$q^s = \sum_{i=1}^{n_s} \alpha_i \times w^s_i, \quad \alpha = \text{Softmax}(WQ^s),$$

where Softmax denotes the softmax layer, $W \in \mathbb{R}^{1 \times z}$ is a trainable variable, and $\alpha \in \mathbb{R}^{1 \times n_q}$ indicates the attention vector. $q^s$ is in the joint latent space with $F^s$.

**Transferring Knowledge from Teacher to Student.** To absorb knowledge from the teacher by learning the consistency of video-query semantic similarity distribution between the teacher and student branches further, we first calculate the similarity distribution $C^s \in \mathbb{R}^k$ of current student branch between $F^s$ and $q^s$:

$$C^s = [\cos(f^s_1, q^s), \cos(f^s_2, q^s), \ldots, \cos(f^s_k, q^s)].$$

Then, we design a distribution distillation to transfer knowledge from the pre-trained teacher model to the inheritance student branch. Specifically, our distribution distillation strategy is to capture the consistency of the similarity distributions of the teacher and the student. A similarity consistency constraint is configured to guide the learning of the inheritance branch with the teacher model [51]. In detail, given the teacher-similarity distribution $C^t$ and the student-similarity distribution $C^s$ of a video-text pair $(V, Q)$, the semantic consistency loss $L_c$ is formulated by exploiting the KL divergence as:

$$L_c = D_{KL}(C^s || C^t) = \sum_{i=1}^{k} C^s_i \log \frac{C^s_i}{C^t_i},$$

where the subscript $i$ indicates the $i$-th elements in the corresponding similarity distribution.

Besides, we also employ self-similarity learning to make the partially relevant video-text pairs near and irrelevant pairs far away in the learned space. Following the previous work [8], by constructing the positive and negative video-text samples, both triplet ranking loss [11, 14] and InfoNCE loss [42, 63] are jointly utilized to self-train the inheritance branch, which can be noted as $L_t$. Overall, to train our inheritance student branch, we simultaneously optimize both $L_s$ and $L_c$ to learn the self-similarity and the similarity consistency. The final loss $L_1$ of this branch can be termed as a weighted sum of $L_s$ and $L_c$ as:

$$L_1 = L_s + wL_c,$$

where $w$ is a hyper-parameter to balance the contribution of $L_s$ and $L_c$. 

11305
3.2.2 Exploration Student Branch

Since the teacher model can not always perform well on various data due to the domain gap, the inheritance student branch may be prone to its mistake when the teacher model is of mediocre performance. Therefore, we design another student branch, called the exploration branch, to only learn the data-specific property of the training set without any guidance from the teacher branch. By jointly optimizing the two branches in a dual learning manner, we can effectively take advantage of the teacher models on well-performing data while mitigating the negative impact of the teacher model’s performance degradation on certain data. In contrast to the inheritance branch that updates its similarity distribution by referring to the teacher knowledge, the exploration branch is devised to learn the data-specific knowledge directly from the on-site training data. As shown in Fig. 2, both two branches share the same network architecture. We also utilize the triplet ranking loss [11, 14] and InfoNCE loss [42, 63] to jointly train the exploration branch, and note the overall loss as $L_E$.

3.3. Dynamic Knowledge Distillation

Although we can directly joint learn the two student branches, this training process remains two-aspect concerns: (1) Firstly, as we mentioned, the CLIP teacher model may have huge performance differences on different datasets due to their task-specific domain gaps. Therefore, when the teacher model is of mediocre performance, how to reduce the impact of the inheritance branch while strengthening the exploration branch learning is important. (2) Secondly, it is worth noticing that continuously pushing the student model to learn more knowledge from the teacher at the beginning of the training, we set a larger initial value to $w$, while learning more from the on-site data gradually otherwise when the student model getting stronger. Specifically, to obtain a more balanced and better distillation result from the dual-branch learning, a dynamic distillation strategy is introduced. It is devised to tune the hyper-parameter $w$ in Eq. (7) online during the model training instead of setting it to a fixed constant one like most previous works. At the beginning of the training, we set a larger initial value to $w$ to learn more knowledge from the teacher, then we decay $w$ smoothly according to the training epochs. Formally, $w$ is computed as:

$$w = w_0 g(t), \quad (7)$$

where $w_0$ is an initial weight, $t$ indicates $t$-th epoch during the training, and $g(\cdot)$ is the decay strategy function.

Particularly, inspired by [3, 37], we implement the dynamic distillation strategy with three different types of decay strategy functions, and the experiments demonstrate the results are all promising. As illustrated in Fig. 3, the three decay strategy functions are:

- Exponential decay: $g(t) = k^t$, where $k < 1$ is the factor that control the decay trend.
- Linear decay: $g(t) = kt + b$, where $k < 0$ and $b$ is for controlling the downward trend of the slash.
- Sigmoid decay: $g(t) = \frac{k}{1 + e^{kt}}$, where $k$ is a hyper-parameter to control the decay.

3.4. Training and Inference

During the whole framework training, we only optimize the student model by jointly minimizing the losses of the inheritance branch and the exploration branch. The overall training loss $\mathcal{L}$ is formulated as:

$$\mathcal{L} = \mathcal{L}_I + \mathcal{L}_E. \quad (8)$$

During the inference, only the student model is utilized to obtain the retrieval results. Given a video-text pair, we first calculate their similarities from both inheritance and exploration branches, resulting in $S_I(Q, V)$ and $S_E(Q, V)$. The final similarity is computed as:

$$S(Q, V) = (1 - \beta)S_I(Q, V) + \beta S_E(Q, V), \quad (9)$$

where $\beta$ is a hyper-parameter to balance the two similarities. Given a textual query, all candidate videos are sorted in terms of their final similarities with the query.

4. Experiments

4.1. Experimental Setup

4.1.1 Datasets

In order to validate the effectiveness of our model, we adopt the long untrimmed video datasets ActivityNet Captions [26] and TVR [27]. Note that the pre-trained CLIP
Table 1. The effectiveness of dual learning with both inheritance and exploration branches. Our proposed model not only outperforms the single-branch counterparts but also performs better than simple two-branch baselines.

<table>
<thead>
<tr>
<th>Branch</th>
<th>ActivityNet</th>
<th>TVR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Inheritance</td>
<td>Exploration</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>7.6</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>5.9</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>8.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Double-Inheritance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Double-Exploration</td>
</tr>
</tbody>
</table>

performs well on ActivityNet Captions, but mediocrely on TVR. Here we briefly introduce these two datasets.

**ActivityNet Captions** [26] is originally developed for dense video captioning task. As captions are partially relevant with the corresponding videos (a caption is typically associated with a specific moment in a video), it has been re-purposed for partially relevant video retrieval. It contains around 20K videos from YouTube, and the average length of videos is around 118 seconds. On average, each video has around 3.7 moments with a corresponding sentence description. For a fair comparison, we adopt the same data partition used in [8]. For ease of reference, we refer to the dataset as ActivityNet.

**TV show Retrieval (TVR)** [27] is originally developed for video corpus moment retrieval, and now can be also used for partially relevant video retrieval. It contains 21.8K videos collected from 6 TV shows, and the average length of videos is around 76 seconds. Each video is associated with 5 natural language sentences that describe a specific moment in the video. As a moment is typically a part of a video, sentences are partially relevant to videos. We utilize the same data partition in [8, 62, 63].

### 4.1.2 Evaluation Metrics

Following the previous work [8], we utilize the rank-based metrics, namely \( \text{R@}K \) (\( K = 1, 5, 10, 100 \)). \( \text{R@}K \) stands for the fraction of queries that correctly retrieve desired items in the top \( K \) of the ranking list. The performance is reported in percentage (%). The SumR is also utilized as the overall performance, which is defined as the sum of all recall scores. Higher scores indicate better performance.

### 4.1.3 Implementation Details

For the CLIP teacher model, we adopt a Vision Transformer based ViT-B/32 provided by OpenAI\(^1\), and encode video frames and query sentences to 512-D features. For the student model, we directly utilize the video and sentence features provided by [8] as the input. For the model training, we set the initial learning rate to 0.00025 and use the same learning schedule as [27]. We use the early stop schedule that the model will stop when the evaluated SumR exceeds 10 epochs without promotion. The maximum number of epochs is set to 100. We choose exponential decay as the default one unless otherwise stated, where the initial weight \( w_0 \) is 0.1 and the hyper-parameter \( \epsilon \) in exponential decay is 0.95. During the inference, we empirically set the weights of the inheritance branch and the exploration branch to 0.3 and 0.7 for similarity fusion. Additionally, we use PyTorch to build the model framework and train models on NVIDIA RTX 3090 GPU with a batch size of 128.

### 4.2. Ablation Studies

#### 4.2.1 Effectiveness of Dual Learning

In order to verify the effectiveness of our proposed dual learning with both inheritance and exploration branches, we compare it to the counterparts with the inheritance branch or exploration branch only. The results on both ActivityNet and TVR are summarized in Table 1. Note that the teacher model CLIP we used performs pretty well on ActivityNet, while it performs mediocrely on TVR. On both datasets, the model with both branches consistently performs the best, which demonstrates the effectiveness of our proposed dual learning structure with both inheritance and exploration branches. Especially on TVR where the pre-trained CLIP only achieves SumR score of 110, the model using dual learning obtains a relative SumR gain of around 12% when compared to single-branch counterparts, which is more significant than that on ActivityNet. The result demonstrates that dual learning is more important when the teacher model is of mediocre performance.

Additionally, we also try to verify whether the improvements come from the combination of two branches. We compare our model to the baselines of simply combining two exploration branches (Dual-exploration) or two inheritance branches (Dual-inheritance) without our dynamic distillation strategy. Their worse performance compared to ours demonstrates that the architecture of our dual-branch exploration and inheritance with the dynamic distillation contributes a lot to the final performance.

\(^1\)https://github.com/openai/CLIP
Table 2 shows the ablation study results of dynamic knowledge distillation for both single-branch and dual-branch networks. We compare it to the counterparts without any knowledge distillation or using knowledge distillation with a fixed weight. The fixed weight is set to 0.1 which is the same as the initial weight used in our full model. To ease of reference, we refer to the latter as fixed knowledge distillation. For the single-branch, we observe a phenomenon that the fixed knowledge distillation is beneficial on ActivityNet, but it hurts the performance on TVR. It allows us to conclude that the common knowledge distillation with a fixed weight is not suitable when the performance of the teacher model is mediocre. By contrast, on both datasets our proposed dynamic knowledge distillation consistently achieves performance gain over the ones without the distillation. For the dual-branch, the fixed knowledge distillation also consistently improves the performance on both datasets, but still worse than our dynamic knowledge distillation. The results not only again confirm the effectiveness of our dual learning with two branches, but also demonstrate the advantage of the dynamic knowledge distillation.

### 4.2.2 Effectiveness of Dynamic Knowledge Distillation

Table 2 shows the effectiveness of dynamic knowledge distillation. Note that Fixed indicates the model using knowledge distillation with a fixed weight during the training.

<table>
<thead>
<tr>
<th>Branch</th>
<th>Distillation</th>
<th>ActivityNet</th>
<th></th>
<th>TVR</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>R@1</td>
<td>R@5</td>
<td>R@10</td>
<td>R@100</td>
</tr>
<tr>
<td>Single</td>
<td></td>
<td>5.9</td>
<td>20.0</td>
<td>32.3</td>
<td>73.7</td>
</tr>
<tr>
<td></td>
<td>(Fixed)</td>
<td>7.6</td>
<td>23.4</td>
<td>35.4</td>
<td>76.2</td>
</tr>
<tr>
<td></td>
<td>(Dynamic)</td>
<td>7.5</td>
<td>24.1</td>
<td>36.2</td>
<td>76.0</td>
</tr>
<tr>
<td>Dual</td>
<td></td>
<td>6.8</td>
<td>22.3</td>
<td>34.5</td>
<td>75.6</td>
</tr>
<tr>
<td></td>
<td>(Fixed)</td>
<td>7.7</td>
<td>24.9</td>
<td>36.6</td>
<td>77.1</td>
</tr>
<tr>
<td></td>
<td>(Dynamic)</td>
<td>8.0</td>
<td>25.0</td>
<td>37.5</td>
<td>77.1</td>
</tr>
</tbody>
</table>

### 4.2.3 Influence of Decay Strategies

In this section, we explore three decay strategies with various initial weights, and also include the model using the distillation with the fixed weight as the baseline. The results on ActivityNet and TVR are demonstrated in Fig. 4. On the whole, the three decay strategies give similar results. Considering the relatively more stable performance of exponential decay, we choose it as the default decay strategy. Additionally, all three variants with dynamic knowledge distillation consistently outperform the baseline, which again confirms the effectiveness of our proposed model. What is more, we find that dynamic knowledge distillation is much less sensitive to the initial weight than the baseline. It makes the dynamic knowledge distillation more appealing, as it alleviates the cumbersome efforts of hyper-parameter tuning.

### 4.3. Comparison with the State-of-the-Art

Table 3 summarizes the comparison results with other methods on ActivityNet. Our proposed model outperforms all the competitor models with clear margins. Among all methods, only our model utilizes the knowledge distillation, the results justify the viability of using the knowledge distillation for partially relevant video retrieval. In addition, although the previous best-performing model MS-SL also utilizes two branches, their two branches solely learn from the training data without extra knowledge. By contrast, in our model with the dual learning paradigm where one branch
Table 3. Performance comparison on ActivityNet. Models are sorted in ascending order in terms of their overall performance.

<table>
<thead>
<tr>
<th>Model</th>
<th>R@1</th>
<th>R@5</th>
<th>R@10</th>
<th>R@100</th>
<th>SumR</th>
</tr>
</thead>
<tbody>
<tr>
<td>W2VV [9]</td>
<td>2.2</td>
<td>9.5</td>
<td>16.6</td>
<td>45.5</td>
<td>73.8</td>
</tr>
<tr>
<td>HTM [43]</td>
<td>3.7</td>
<td>13.7</td>
<td>22.3</td>
<td>66.2</td>
<td>105.9</td>
</tr>
<tr>
<td>HGR [6]</td>
<td>4.0</td>
<td>15.0</td>
<td>24.8</td>
<td>63.2</td>
<td>107.0</td>
</tr>
<tr>
<td>RIVRL [12]</td>
<td>5.2</td>
<td>18.0</td>
<td>28.2</td>
<td>66.4</td>
<td>117.8</td>
</tr>
<tr>
<td>VSE++ [14]</td>
<td>4.9</td>
<td>17.7</td>
<td>28.2</td>
<td>67.1</td>
<td>117.9</td>
</tr>
<tr>
<td>DE++ [11]</td>
<td>5.3</td>
<td>18.4</td>
<td>29.2</td>
<td>68.0</td>
<td>121.0</td>
</tr>
<tr>
<td>DE [10]</td>
<td>5.6</td>
<td>18.8</td>
<td>29.4</td>
<td>67.8</td>
<td>121.7</td>
</tr>
<tr>
<td>W2VV++ [28]</td>
<td>5.4</td>
<td>18.7</td>
<td>29.7</td>
<td>68.8</td>
<td>122.6</td>
</tr>
<tr>
<td>CE [35]</td>
<td>5.5</td>
<td>19.1</td>
<td>29.9</td>
<td>71.1</td>
<td>125.6</td>
</tr>
<tr>
<td>ReLoCLNet [63]</td>
<td>5.7</td>
<td>18.9</td>
<td>30.0</td>
<td>72.0</td>
<td>126.6</td>
</tr>
<tr>
<td>XML [27]</td>
<td>5.3</td>
<td>19.4</td>
<td>30.6</td>
<td>73.1</td>
<td>128.4</td>
</tr>
<tr>
<td>MS-SL [8]</td>
<td>7.1</td>
<td>22.5</td>
<td>34.7</td>
<td>75.8</td>
<td>140.1</td>
</tr>
<tr>
<td>DL-DKD (Ours)</td>
<td>8.0</td>
<td>25.0</td>
<td>37.5</td>
<td>77.1</td>
<td>147.6</td>
</tr>
</tbody>
</table>

Table 4. Performance comparison on the TVR dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>R@1</th>
<th>R@5</th>
<th>R@10</th>
<th>R@100</th>
<th>SumR</th>
</tr>
</thead>
<tbody>
<tr>
<td>W2VV [9]</td>
<td>2.6</td>
<td>5.6</td>
<td>7.5</td>
<td>20.6</td>
<td>36.3</td>
</tr>
<tr>
<td>HGR [6]</td>
<td>1.7</td>
<td>4.9</td>
<td>8.3</td>
<td>35.2</td>
<td>50.1</td>
</tr>
<tr>
<td>HTM [43]</td>
<td>3.8</td>
<td>12.0</td>
<td>19.1</td>
<td>63.2</td>
<td>98.2</td>
</tr>
<tr>
<td>CE [35]</td>
<td>3.7</td>
<td>12.8</td>
<td>20.1</td>
<td>64.5</td>
<td>101.1</td>
</tr>
<tr>
<td>W2VV++ [28]</td>
<td>5.0</td>
<td>14.7</td>
<td>21.7</td>
<td>61.8</td>
<td>103.2</td>
</tr>
<tr>
<td>VSE++ [14]</td>
<td>7.5</td>
<td>19.9</td>
<td>27.7</td>
<td>66.0</td>
<td>121.1</td>
</tr>
<tr>
<td>DE [10]</td>
<td>7.6</td>
<td>20.1</td>
<td>28.1</td>
<td>67.1</td>
<td>123.4</td>
</tr>
<tr>
<td>RIVRL [12]</td>
<td>9.4</td>
<td>23.4</td>
<td>32.2</td>
<td>70.6</td>
<td>135.6</td>
</tr>
<tr>
<td>XML [27]</td>
<td>10.0</td>
<td>26.5</td>
<td>37.3</td>
<td>81.3</td>
<td>155.1</td>
</tr>
<tr>
<td>ReLoCLNet [63]</td>
<td>10.7</td>
<td>28.1</td>
<td>38.1</td>
<td>80.3</td>
<td>157.1</td>
</tr>
<tr>
<td>MS-SL [8]</td>
<td>7.1</td>
<td>22.5</td>
<td>34.7</td>
<td>75.8</td>
<td>140.1</td>
</tr>
<tr>
<td>DL-DKD (Ours)</td>
<td>8.0</td>
<td>25.0</td>
<td>37.5</td>
<td>77.1</td>
<td>147.6</td>
</tr>
</tbody>
</table>

Figure 5. Performance of different models on different types of queries. Queries are grouped according to their M/V. The smaller M/V indicates more challenging of queries.

Table 5. Performance with different teacher models. Our proposed framework supports various teacher models, and also allows for distilling for multiple teachers jointly.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Teacher</th>
<th>R@1</th>
<th>R@5</th>
<th>R@10</th>
<th>R@100</th>
<th>SumR</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActivityNet</td>
<td>CLIP</td>
<td>8.0</td>
<td>25.0</td>
<td>37.5</td>
<td>77.1</td>
<td>147.6</td>
</tr>
<tr>
<td></td>
<td>TCL</td>
<td>7.3</td>
<td>24.1</td>
<td>36.2</td>
<td>76.4</td>
<td>144.0</td>
</tr>
<tr>
<td></td>
<td>CLIP+TCL</td>
<td>8.1</td>
<td>25.3</td>
<td>37.7</td>
<td>77.6</td>
<td>148.6</td>
</tr>
<tr>
<td>TVR</td>
<td>CLIP</td>
<td>14.4</td>
<td>34.9</td>
<td>45.8</td>
<td>84.9</td>
<td>179.9</td>
</tr>
<tr>
<td></td>
<td>TCL</td>
<td>13.5</td>
<td>33.1</td>
<td>44.6</td>
<td>84.1</td>
<td>175.3</td>
</tr>
<tr>
<td></td>
<td>CLIP+TCL</td>
<td>15.1</td>
<td>35.4</td>
<td>46.5</td>
<td>84.5</td>
<td>181.6</td>
</tr>
</tbody>
</table>

By contrast, our proposed model achieves more balanced performance in all groups, which shows that our model is less sensitive to irrelevant content in videos.

4.4. Extension to Multi-Teacher Distillation

While the focus of our work in the Method Section is only the single-teacher distillation, it is natural to consider whether the framework can be extended to multi-teacher distillation. Therefore, we adopt another vision-language pre-training model TCL [57] as an extra teacher model. As shown in Table 5, utilizing TCL as the teacher model still gives better performance than the previous state-of-the-art works. Additionally, with the joint use of CLIP and TCL as teacher models (their output distributions are fused by simple summation), it brings a further performance boost over the single-teacher distillation. We believe this extension may be useful in scenarios where various vision-language pre-training models are available during training.
4.5. Complementarity between the two branches

Recall that our proposed model consists of an inheritance branch and an exploration branch, here we explore their complementarity. We measure the complementarity via Pearson correlation coefficient between the similarity distributions of two branches, i.e., $C^t$ and $C^e$. Besides our model, we also compute the correlation coefficient of the common two-branch baseline without distillation (i.e., Double-Exploration). On the ActivityNet dataset, our model achieves a coefficient of 0.622, while the baseline obtains a coefficient of 0.749. Note that the lower coefficient indicates the less correlated between two branches and more complementary. The result demonstrates that the two branches of our model are more complementary, which to some extent illustrates why our model is better than the two-branch baseline.

5. Conclusions

In this paper, we have investigated the meaningful but challenging text-to-video subtask of PRVR from a new perspective of knowledge distillation. A novel framework, i.e., KL-DKD, has been proposed to distill the generalization knowledge from the large-scale vision-language pre-trained model to a task-specific network. Extensive experiments on both ActivityNet and TVR datasets support the following conclusions: (1) Dual learning of an inheritance branch and an exploration branch is necessary for knowledge distillation. (2) Besides, the dynamic knowledge distillation further improves performance, especially when the teacher model is of mediocre performance. (3) For state-of-the-art performance, we recommend dual learning with dynamic knowledge distillation for PRVR.

Acknowledgements. This work was supported by the “Pioneer” and “Leading Goose” R&D Program of Zhejiang (No.2023CO1212), National Natural Science Foundation of China (No. 61976188), Young Elite Scientists Sponsorship Program by CAST (No. 2022QNRC001), the open research fund of The State Key Laboratory of Multimodal Artificial Intelligence Systems, and the Fundamental Research Funds for the Provincial Universities of Zhejiang.

References


[16] Sheng Fang, Shuhui Wang, Junbao Zhuo, Xinzhe Han, and Qingming Huang. Learning linguistic association towards


[41] Antoine Micich, Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Andrew Zisserman. Thinking fast and slow: Effi-


