Neurocomputing 614 (2025) 128851

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

S2CA: Shared Concept Prototypes and Concept-level Alignment for text–video retrieval

Yuxiao Li, Yu Xin *, Jiangbo Qian, Yihong Dong

The Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo, China

ARTICLE INFO

Keywords:
Video–text retrieval
Cross-modal learning
Concept prototypes learning
Represents decoupling

ABSTRACT

Text–video retrieval, as a fundamental task of cross-modal learning, relies on effectively establishing the semantic association between text and video. At present, mainstream semantic alignment methods for text–video adopt instance-level alignment strategies, ignoring the fine-grained concept association and the "concept-level alignment" characteristics of text–video. In this regard, we propose **S**hared **C**oncept Prototypes and **C**oncept-level **A**lignment (**S2CA**) to achieve concept-level alignment. Specifically, we utilize the text–video **Shared Concept Prototypes** mechanism to bridge the correspondence between text and video. On this basis, we use cross-attention and Gumbel-softmax to obtain **Discrete Concept Allocation Matrices** and then assign text and video tokens to corresponding concept prototypes. In this way, texts and videos are decoupled into multiple **Conceptual Aggregated Features**, thereby achieving **Concept-level Alignment**. In addition, we use CLIP as the teacher model and adopt the Align-Transform-Reconstruct distillation framework to strengthen the multimodal semantic learning ability. The extensive experiments on MSR-VTT, DiDeMo, ActivityNet and MSVD prove the effectiveness of our method.

1. Introduction

With the rapid development of digital media technology and the popularization of the internet, large-scale text and video data are constantly emerging at an astonishing speed. These data contain a vast amount of information and knowledge, and how to quickly and accurately retrieve the needed information from these massive data has become one of the urgent problems to be solved. Text–Video Retrieval (TVR) models [1–5] rely on cross-modal learning approaches such as Video–Text Contrastive Loss (VTC) [3], Video–Text Matching (VTM) [6], and Masked Language Modeling (MLM) [7] to model the semantic associations between text and video, achieving cross-modal text–video retrieval and video–text retrieval, making it possible to automatically extract effective information from massive texts and videos [8–11]. However, most TVR models currently have two issues: (1) TVR models that train from scratch (we call it "cold start") [3,12–14] converge slowly and require a large amount of computational resources, posing significant obstacles in further improving the performance. (2) The mainstream TVR models adopt either global alignment [9,15,16] or local alignment [17] (two different implementation methods of VTC), both of which assume that the paired text and video have completely corresponding feature semantics, neglecting the inherent "concept-level alignment" characteristics of text and video. For these two existing issues, we will elaborate on them in the following **Motivation** paragraph.

Motivation-1. For issue (1), many related works propose finetuning the well-known text-image multimodal model, *i.e.*, CLIP [18], on text–video data, thereby avoiding the high computational demands and limitations of text–video dataset required for training TVR models from scratch. However, CLIP-ViP [15] observes that increasing the amount of text–video data during CLIP finetuning can paradoxically lead to a decline in model performance. A credible explanation for this is that CLIP is trained solely on text-image data. Due to the inherent domain gap between text–video data and text-image data, namely the additional temporal information in videos compared to images, finetuning CLIP with excessive text–video data may disrupt the originally learned multimodal knowledge within CLIP. Thus, simply finetuning CLIP seems to have encountered a bottleneck. Fortunately, UMT [10] proposes a text–video retrieval model that learns CLIP's powerful multimodal comprehension ability through distilling CLIP's features during training. This approach allows the model to retain CLIP's multimodal semantic comprehension of static images while learning temporal information from text–video data, enabling the model to achieve satisfactory results under acceptable computational and dataset constraints, thereby offering a new direction for TVR. Based on this, we follow the approach

* Corresponding author.
E-mail addresses: 2211100091@nbu.edu.cn (Y. Li), xinyu@nbu.edu.cn (Y. Xin).

https://doi.org/10.1016/j.neucom.2024.128851
Received 15 July 2024; Received in revised form 28 September 2024; Accepted 31 October 2024
Available online 8 November 2024

of UMT by distilling CLIP features to acquire preliminary multimodal knowledge. Additionally, we utilize the Align-Transform-Reconstruct training framework to enhance the distillation effect. For details, please refer to Section 3.1.

Motivation-2. We will further elaborate on issue (2). Global alignment aligns the overall features of text and video [9,15,16], which results in equal treatment of the discriminative regions of text and video, making it challenging to capture local details [19]. On the other hand, local alignment aligns each word in the text with each frame of the video. Essentially, both methods treat text and video as an instance, achieving instance-level alignment, but neglect the inherent "concept-level alignment" characteristics of text and video, which leads to suboptimal performance. We believe that both text and video are composed of a series of concepts, where a particular concept in the text may only correspond to a specific concept in the video and be unrelated to others, and vice versa. This characteristic, which we term the "concept-level alignment" property of text and video, is overlooked by both global alignment and local alignment. As shown in (Fig. 1, top), the text is composed of three concepts: 1. husky dog, 2. tree around, 3. girl, and the video contains corresponding concepts as well. However, both global alignment and local alignment tend to draw the concept of "tree around" in the text closer to those of "husky dog" and "girl" in the video within the semantic space. Intuitively, the "tree around" concept in the text should only align with the "tree around" concept in the video, and not with the "husky dog" concept or the "girl" concept in the video. Clearly, this incorrect matching can lead the model to overlook detailed information, resulting in a decrease in retrieval performance.

To address the aforementioned issues, a crucial step is decoupling the concept features embedded in text and video. Two relevant works DiCoSA [19] and GroupViT [20] have inspired us. After extracting the overall features of text and video respectively, DiCoSA uses Multi-Layer Perceptron (MLP) to map the coarse-grained overall features of text and video into multiple latent factors. However, this decoupling method completely relies on the performance of MLP. Even though it employs two additional loss functions to constrain and optimize the decoupling ability of MLP, its suboptimal performance indicates that simple, modality-independent MLP is difficult to fully achieve fine-grained decoupling. These limitations also compel us to seek other decoupling methods.

GroupViT is a notable work in unsupervised image semantic segmentation, which utilizes group tokens as clustering centers within the image encoder. It maps image tokens into group tokens, and then averages the features of group tokens to align them with the overall textual features. Ultimately, similar features in the image will be mapped into the same group token, enabling the matching of group tokens through the input text of the target segmentation object, and subsequently obtaining the corresponding segmentation results, we believe that this framework is capable of achieving concept decoupling effectively. The objective of GroupViT is to cluster images and decouple only the image domain. However, our goal differs in that we aim to decouple both video and text simultaneously, hence we employ a concept decoupling architecture for both the video and text domains. Furthermore, to determine the correspondence between decoupled text concepts and video concepts, we innovatively utilize shared concept prototypes for decoupling text and video separately. These shared concept prototypes serve as the common clustering centers for both the text and video domains, facilitating communication between them, without the need for additional loss functions to constrain the decoupling process, as in DiCoSA. As shown in (Fig. 1, bottom), text and video can be measured and compared through multiple **Shared Concept Prototypes**. Each prototype represents a class of similar concepts, text and video features are clustered based on their semantic similarity to the prototypes, allowing the concept features related to "tree around" in the text to be pulled closer to those related to "tree around" in the video in the semantic space, while not being pulled closer to the concept features

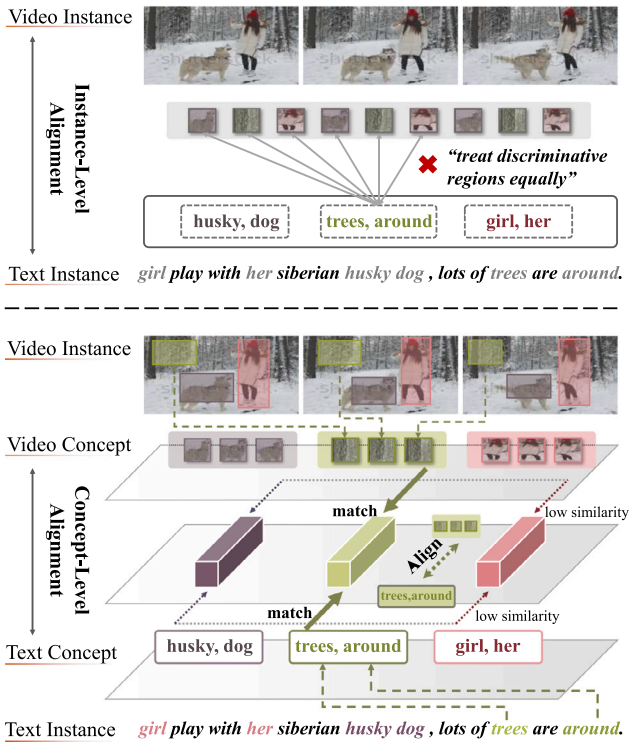

Fig. 1. Top: Gray lines represent instance-level alignment aligns text features "trees around" with all video features. **Bottom:** The three colored bars in the middle represent the concept prototypes shared by text and video.

related to "husky dog" and "girl", thereby achieving **Concept-level Alignment**.

Work Summary. Based on the above, we propose **S**hared **C**oncept Prototypes and **C**oncept-level **A**lignment (**S2CA**), as shown in Fig. 2. According to **Motivation-1**, we divide the training of S2CA into two stages: **Stage 1: Distillation Learning.** In stage 1, we adopt a student–teacher architecture for distillation learning. The video encoder (vanilla ViT [21]) serves as a student to learn the rich multimodal semantic knowledge of the teacher model (*i.e.*, CLIP [18]), additionally, we utilize the Align-Transform-Reconstruct training framework to enhance the distillation effect. For details, please refer to Section 3.1. **Stage 2: Text–video cross-modal pretraining.** In stage 2, we use the distilled video encoder for text–video cross-modal pretraining. According to **Motivation-2**, to perform concept decoupling on text and video, as shown in (Fig. 2, bottom), we set **Shared Concept Prototypes** for the text and video modalities as the shared cluster centers. We concatenate video tokens and text tokens with shared concept prototypes and interact with each other through their respective encoder. Then, cross-attention and `Gumbel-Softmax` [22,23] are used to obtain **Discrete Concept Allocation Matrices** and then map each token to the corresponding prototype, and the tokens mapped to the same prototype are averaged to obtain **Conceptual Aggregated Features**. The shared concept prototypes ensures that the same benchmark is used to measure the conceptual features in both text and video, and guarantees a one-to-one correspondence between text concepts and video concepts. Ultimately, we employ contrastive learning [24–27] to align the conceptual aggregated features of text and video on a one-to-one basis, achieving **Concept-level Alignment**. In the two-stage training process, we draw inspiration from UMT [10] and apply a large proportion of masking to video tokens, significantly reducing computational complexity. Our contributions are summarized as follows:

- We propose a text–video shared concept prototypes learning strategy to decouple conceptual semantics for concept-level alignment. Shared concept prototypes serve as common benchmarks

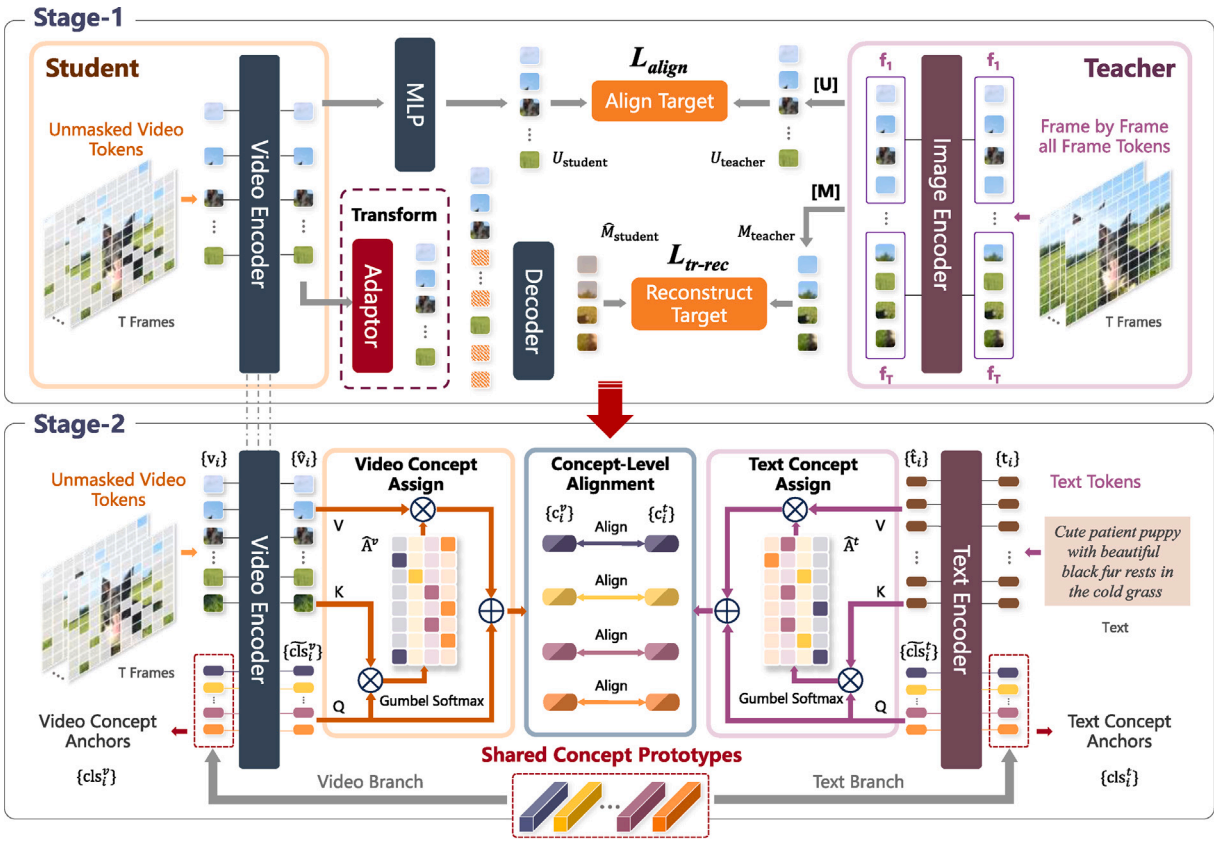

Fig. 2. Overall architecture. In stage 1, the teacher is CLIP-ViT, and the student is a randomly initialized vanilla ViT. The video encoder of stage 2 is initialized by the video encoder of stage 1. We mask out a large number of tokens in both stages for efficient training. **[U]** represents the CLIP tokens corresponding to unmasked tokens, and **[M]** represents the CLIP tokens corresponding to masked tokens.

for text and video comparison, and measure the similarity between text and video from multiple perspectives.

- To address the "cold start" problem, we adopt a two-stage training framework. In stage 1, we adopt an Align-Transform-Reconstruct distillation framework to provide initial knowledge for stage 2. Both stages we mask video tokens at a high ratio, greatly saving computational resources and making S2CA resource-friendly.
- During inference, we simply calculate the cosine similarity between text and video, but exceed the methods of using the two-stage re-ranking retrieval. Extensive experiments on four benchmarks demonstrate the effectiveness of our method.

2. Related work

We introduce three types TVR models in turn based on their development sequence: (1) Training from Scratch, (2) Finetuning based on CLIP, and the latest (3) Knowledge Distillation Based on CLIP. Finally, we elaborate on the content related to Concept Decoupling.

Training from Scratch Early TVR models often underwent cross-modal pretraining from scratch on large-scale text–video datasets, followed by finetuning on downstream text–video retrieval datasets. However, the difficulty in acquiring text–video datasets, coupled with the significantly higher computational cost of training video models compared to image models, constrained the development of TVR models trained from scratch. ClipBert [14] is a pioneer in end-to-end text–video pretraining, which proposes sparse sampling of frames from videos to reduce computational complexity. Frozen [3] uniformly samples video frames and treats the images as "frozen in time" videos, enabling joint training on both image-text and video–text datasets.

Finetuning based on CLIP The emergence of the CLIP [18] model has significantly accelerated the progress in the field of TVR. CLIP underwent multimodal pretraining on 400M text-image pairs, endowing it with exceptional capabilities in understanding multimodal text-image representations. By directly finetuning CLIP on downstream text–video datasets, outstanding performance can be achieved with reduced computational and time costs. CLIP4Clip [9] is the first to transfer the knowledge of the CLIP model into an end-to-end text–video retrieval model. DRL [28] adopts fine-grained contrastive learning, which aligns the features of each frame and each word separately. X-CLIP [8] proposes cross-grained contrastive learning, modeling the feature associations across multiple grained of text (word, sentence) and video (frame, video). X-Pool [2] extracts text-relevant features from the video, using the text as a condition. Essentially, none of the above methods take into account the "concept-level alignment" characteristics between text and video, which we have elaborated on in motivation-2 of the introduction.

Knowledge Distillation Based on CLIP Recently, UMT [10] learns CLIP's multimodal comprehension ability through feature-based distillation [29–32] while being trained on text–video data. Specifically, UMT masks a significant portion of video tokens and enforces the model to encode the unmasked tokens in such a way that their representations, when compared to the corresponding CLIP tokens, are similar, as measured by a Mean Squared Error (MSE) loss. We refer to this objective of aligning the unmasked tokens with CLIP's as the "Align Target". In this way, UMT can maintain CLIP's powerful semantic understanding ability while pretraining TVR models, without encountering the issue of damaging CLIP's original knowledge, which may arise from excessively finetuning CLIP with video data, as mentioned in the introduction and CLIP-Vip [15]. However, it was found that reconstructing CLIP features through a decoder resulted in a decrease in the model performance.

UMT believe that this result was due to the difficulty of reconstructing high semantic features for the model. However, we believe that the performance degradation is due to the conflict between the Align Target and the Reconstruct Target, the proper use of the Reconstruct Target can help to model temporal information in the video and improve the effectiveness of distillation learning. For details, please refer to Section 3.1.

Concept Decoupling The current mainstream training pipeline for TVR models involves encoding the features of text and video, mapping them into a shared semantic space. In this semantic space, contrastive learning is employed to pull paired text and video features closer together while pushing unpaired ones apart. As mentioned in motivation-2 of the introduction, the specific implementation of pulling features closer together can be primarily categorized into two approaches: Global Alignment and Local Alignment. Both of these approaches essentially achieve instance-level alignment, but in reality, the alignment should be at a concept-level. Achieving concept-level alignment hinges critically on the ability to decouple text and video concept, which remains an urgent problem to solve. However, some works have made contributions in this direction. DiCoSA [19] maps the overall features of text and video to multiple latent factors through MLP, achieving disentangled conceptualization of two modalities and set-to-set alignment. However, its decoupling process is only based on the overall features and only uses a simple MLP, which is not sufficient. GroupViT [20] uses text-image contrastive learning to associate the semantics of images and text while also enabling image tokens with similar features to be grouped together through learning. We follow the idea of grouping and apply it to the fields of TVR, and decouple both video and text based on concept prototypes simultaneously.

3. Methodology

The overall architecture of S2CA is shown in Fig. 2. The training is divided into two stages (*i.e.*, stage 1 and stage 2). Given a video clip, we embed it into a sequence of video tokens with a patch size of $16 \times 16 \times 1$, where the "1" signifies the patch size along the temporal dimension. Following UMT [10], for both stages of our approach, we mask video tokens in order to enhance training efficiency. However, the masking ratios differ between the two stages, with the specific ratios and more configurations outlined in Tables 4 and 5. For simplicity, we represent the video input as $\boldsymbol{V} \in \mathbb{R}^{L\times C}$, where L represents the number of unmasked tokens, C represents the feature dimension. In stage 1, S2CA (the video encoder, a vanilla ViT without CLS) as the student learns the cross-modal understanding ability of CLIP through (i) Align Target, (ii) Feature Transform, and (iii) Reconstruct Target. The details will be introduced in Section 3.1. In stage 2, Shared Concept Prototypes $\{\mathbf{cls}_1, \mathbf{cls}_2, \ldots, \mathbf{cls}_M\}$ are randomly initialized, and video conceptual aggregated features (VCAF) and text conceptual aggregated features (TCAF) are obtained through Video Concept Assign (VCA) and Text Concept Assign (TCA). Through contrastive learning, VCAF and TCAF are aligned one-to-one at the concept-level with specific details introduced in Section 3.2.

3.1. Stage 1: Distillation learning architecture of Align-Transform-Reconstruct

As shown in (Fig. 2, top), the unmasked tokens, after being encoded by the video encoder, are aligned with the corresponding tokens from CLIP, which we refer to as the Align Target. At the same time, S2CA can perform token reconstruction through the encoder–decoder architecture and align the reconstructed masked tokens with the corresponding CLIP tokens as Reconstruct Target. In addition, to better balance the two objectives, we design an Adaptor module as Feature Transform to alleviate the negative impact of conflicting learning objectives.

In terms of Align Target For S2CA, the output of $\boldsymbol{V}$ through the video encoder and MLP is recorded as $U_{student} \in \mathbb{R}^{L\times C'}$. For the teacher model CLIP, it encodes each frame with a patch size of 16×16 individually, taking as input the complete sequence of tokens for each frame along with the CLS that comes with CLIP, without performing any masking operation. While for the output, we remove all CLS and filter out tokens corresponding to $U_{student}$, denoted as $U_{teacher}$. We compute MSE between normalized token pairs of $U_{student}$ and $U_{teacher}$, the loss is represented as L_{align}.

In terms of Reconstruct Target Similar to VideoMAE [33], the encoded unmasked tokens $U_{student}$ and mask tokens (learnable tokens, denoted as $\overline{M}_{learn}$) are concatenated to obtain the complete tokens M_{all}. Then, M_{all} reconstructs the masked tokens through the *Decoder*, and the reconstructed tokens are denoted as $M_{student}$:

$$\begin{aligned} M_{all} &= \text{concatenate}(U_{student}, \overline{M}_{learn}), \\ M_{student} &= \text{filter}(\text{Decoder}(M_{all})), \end{aligned} \tag{1}$$

where $filter$ represents the selection of the reconstructed masked tokens from the M_{all} after passing through the *Decoder*. The CLIP tokens corresponding to $M_{student}$ are denoted as $M_{teacher}$, we compute MSE between normalized token pairs of $M_{student}$ and $M_{teacher}$, the loss function is represented as L_{rec}. Simple use of L_{align} and L_{rec} observed performance degradation in UMT [10], we will explain below.

In terms of Feature Transform The goal of L_{align} is to make $U_{student}$ and $U_{teacher}$ as consistent as possible, while the goal of L_{rec} is to enable $U_{student}$ to reconstruct $M_{teacher}$ optimally. Optimizing L_{align} and L_{rec} simultaneously for $U_{student}$ is a multiobjective optimization problem, where L_{align} and L_{rec} collide, making it difficult to obtain a global optimal solution and reducing the effectiveness of distillation.

In this regard, we use a simple yet effective Adaptor module (transformer encoder layers) for feature transformation to alleviate conflicts caused by differences between these two objectives. Specifically, we rewrite Eq. (1) as:

$$\begin{aligned} U_{adapt} &= \text{Adaptor}(U_{student}), \\ \hat{M}_{all} &= \text{concatenate}(U_{adapt}, \overline{M}_{learn}), \\ \hat{M}_{student} &= \text{filter}(\text{Decoder}(\hat{M}_{all})). \end{aligned} \tag{2}$$

While using L_{align} to optimize $U_{student}$, we use the Adaptor for feature transformation to encode features specific to Reconstruct Target. In this way, L_{align} and L_{rec} are decoupled to a certain extent through the Adaptor, thereby reducing the conflict between L_{align} and L_{rec}. We compute MSE between normalized token pairs of $\hat{M}_{student}$ and $M_{teacher}$, the modified loss function is represented as L_{tr-rec}. The final distillation learning loss is obtained by weighted summation of L_{align} and L_{tr-rec}:

$$L_{distill} = \alpha L_{align} + \beta L_{tr-rec}, \tag{3}$$

where α and β are the trade-off hyper-parameter of L_{align} and L_{tr-rec}.

3.2. Stage 2: Shared concept prototypes based concept decoupling for text–video alignment

To decouple the concept in text and video, and achieve concept-level alignment, we establish a simple yet effective framework in stage 2. As shown in (Fig. 2, bottom), the model framework can be divided into three modules: Shared Concept Prototypes (SCP), Text and Video Concept Assign (TCA and VCA), and Concept-Level Alignment (CLA).

Shared Concept Prototypes As shown in (Fig. 2, bottom), we initialize a large number of M learnable vectors $\{\mathbf{cls}_i\}_{i=1}^{M} \in \mathbb{R}^{M\times C}$ as the shared concept prototypes between text and video modalities. Similar to stage 1, we mask out most of the video tokens, the input unmasked video tokens input are represented as $\{\mathbf{v}_i\}_{i=1}^{L} \in \mathbb{R}^{L\times C}$ and the input text tokens are represented as $\{\mathbf{t}_i\}_{i=1}^{K} \in \mathbb{R}^{K\times C}$. For simplicity, we simplify $\{\mathbf{cls}_i\}_{i=1}^{M}$ to $\{\mathbf{cls}_i\}$, $\{\mathbf{v}_i\}_{i=1}^{L}$ to $\{\mathbf{v}_i\}$, and $\{\mathbf{t}_i\}_{i=1}^{K}$ to $\{\mathbf{t}_i\}$. The shared concept prototypes $\{\mathbf{cls}_i\}$ will interact with text and video modalities respectively. Specifically, $\{\mathbf{v}_i\}$ and $\{\mathbf{t}_i\}$ are concatenated with concept prototypes $\{\mathbf{cls}_i\}$ as inputs to the video encoder and text encoder, respectively:

$$\begin{aligned} \{\hat{\mathbf{v}}_i\}, \{\hat{\mathbf{cls}}_i^{v}\}_{i=1}^{M} &= \text{VideoEncoder}[\{\mathbf{v}_i\}; \{\mathbf{cls}_i\}], \\ \{\hat{\mathbf{t}}_i\}, \{\hat{\mathbf{cls}}_i^{t}\}_{i=1}^{M} &= \text{TextEncoder}[\{\mathbf{t}_i\}; \{\mathbf{cls}_i\}], \end{aligned} \tag{4}$$

where [;] denotes the concatenation operator. The shared concept prototypes $\{\mathbf{cls}_i\}$ can be regarded as the shared cluster centers for text and video, to obtain preliminary cluster features $\{\hat{\mathbf{cls}}_i^v\}_{i=1}^M$ and $\{\hat{\mathbf{cls}}_i^t\}_{i=1}^M$. To further obtain fewer but more specific cluster features with more semantic information, similar to the approach in [20], we map M cluster features to fewer N cluster features through MLP:

$$\begin{aligned}\{\widetilde{\mathbf{cls}}_i^v\}_{i=1}^N &= \mathrm{MLP}(\{\hat{\mathbf{cls}}_i^v\}_{i=1}^M),\\ \{\widetilde{\mathbf{cls}}_i^t\}_{i=1}^N &= \mathrm{MLP}(\{\hat{\mathbf{cls}}_i^t\}_{i=1}^M),\end{aligned} \tag{5}$$

where MLP maps on the dimension of the number of cluster features, *i.e.*, we map $\{\hat{\mathbf{cls}}_i^t\}_{i=1}^M \in \mathbb{R}^{M\times C}$ to $\{\widetilde{\mathbf{cls}}_i^t\}_{i=1}^N \in \mathbb{R}^{N\times C}$ and $\{\hat{\mathbf{cls}}_i^v\}_{i=1}^M \in \mathbb{R}^{M\times C}$ to $\{\widetilde{\mathbf{cls}}_i^v\}_{i=1}^N \in \mathbb{R}^{N\times C}$. For simplicity, we simplify $\{\widetilde{\mathbf{cls}}_i^v\}_{i=1}^N$ to $\{\widetilde{\mathbf{cls}}_i^v\}$, and $\{\widetilde{\mathbf{cls}}_i^t\}_{i=1}^N$ to $\{\widetilde{\mathbf{cls}}_i^t\}$.

Text and Video Concept Assign After the above steps, we obtain $\{\widetilde{\mathbf{cls}}_i^v\}$ and $\{\widetilde{\mathbf{cls}}_i^t\}$ for the text and video modalities, respectively. $\{\widetilde{\mathbf{cls}}_i^v\}$ and $\{\widetilde{\mathbf{cls}}_i^t\}$ can be regarded as cluster centers for N latent concept semantics of video and text, respectively. Then, we assign text and video tokens to corresponding concept prototypes based on similarity through Video Concept Assign (VCA) and Text Concept Assign (TCA) to achieve aggregation of similar concept semantic features. The implementation of VCA and TCA is similar. To simplify, we take VCA as an example to introduce. We follow [20] and use $\{\widetilde{\mathbf{cls}}_i^v\}$ as the query and $\{\hat{\mathbf{v}}_i\}$ as the key and value. We calculate the similarity matrix $\mathbf{A}_{i,j}^v$ of $\{\widetilde{\mathbf{cls}}_i^v\}$ and $\{\hat{\mathbf{v}}_i\}$ through cross-attention. After that, we discretize $\mathbf{A}_{i,j}^v$ through `Gumbel-Softmax` [20,22,23,34–36] and a straight-through trick [20,37] so that each video token is only assigned to one video concept prototype (*i.e.*, performing a one-hot operation on $\mathbf{A}_{i,j}^v$ and ensuring that the one-hot operation is differentiable through the straight-through trick), thus obtaining the **Discrete Concept Allocation Matrices** $\hat{\mathbf{A}}^v$:

$$\begin{aligned}\mathbf{A}_{i,j}^v &= \frac{\exp(W_q\widetilde{\mathbf{cls}}_i^v \cdot W_k\hat{\mathbf{v}}_j + \gamma_i)}{\sum_{k=1}^N \exp(W_q\widetilde{\mathbf{cls}}_k^v \cdot W_k\hat{\mathbf{v}}_j + \gamma_k)},\\ \hat{\mathbf{A}}^v &= \mathtt{one-hot}(\mathbf{A}_{\mathtt{argmax}}^v) + \mathbf{A}^v - \mathtt{sg}(\mathbf{A}^v),\end{aligned} \tag{6}$$

where $\{\gamma_i\}$ are independent identically distributed random variables sampled from the `Gumbel(0,1)` distribution, W_q and W_k are the mapping layers, and `sg` represents stopping the gradient operation. $\hat{\mathbf{A}}^v$ is a discrete concept allocation matrices, and its gradient is equivalent to $\mathbf{A}^v$ through the straight-through trick. Finally, the video tokens assigned to the same concept prototype are weighted average and added to the concept prototype through residual connections to obtain the Video Concept Aggregation Features (VCAF) $\{\mathbf{c}_i^v\}_{i=1}^N$:

$$\mathbf{c}_i^v = W_c(\widetilde{\mathbf{cls}}_i^v + W\frac{\sum_{j=1}^L \hat{\mathbf{A}}_{i,j}^v W_v\hat{\mathbf{v}}_j}{\sum_{j=1}^L \hat{\mathbf{A}}_{i,j}^v}), \tag{7}$$

where W, W_c and W_v are the mapping layers. Similarly, through TCA, we can obtain Text Concept Aggregation Features (TCAF) $\{\mathbf{c}_i^t\}_{i=1}^N$.

Concept-Level Alignment Concept-level alignment leverages contrastive learning to enhance the cosine similarity between paired texts and videos. What distinguishes this approach from global alignment and local alignment is that, when calculating the cosine similarity between texts and videos, we compute the similarity between each TCAF and its corresponding VCAF, ignoring the similarity with other non-corresponding VCAF, achieving concept-level alignment. Ultimately, the final similarity between the text and video is obtained by averaging all the calculated concept-level similarities. Specifically, the formula for calculating the similarity $S_{i,j}$ between the i_{th} text and the j_{th} video is as follows:

$$S_{i,j} = \sum_{k=1}^N \frac{(\mathbf{c}_k^t)^\top \mathbf{c}_k^v}{\left\|\mathbf{c}_k^t\right\| \left\|\mathbf{c}_k^v\right\|}. \tag{8}$$

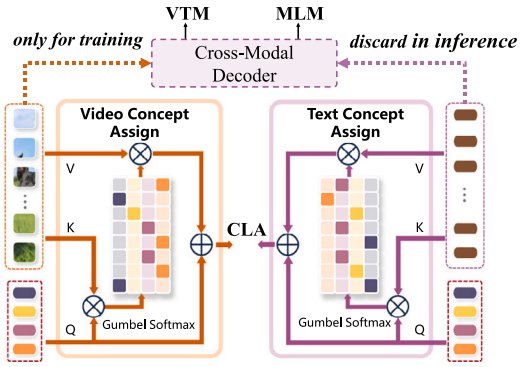

Fig. 3. In addition to CLA, we use VTM and MLM to enhance the learning of multimodal semantics. During the inference stage, the joint cross-modal decoder with VTM and MLM loss is discarded.

Table 1
The effectiveness of Adaptor module. The first line represents model that only use Align Target.

Adaptor	Decoder	Text->Video			
		R@1	R@5	R@10	Mean
–	–	39.7	68.0	**79.4**	62.37
–	2	40.2	68.3	78.4	62.30
1	2	**41.7**	67.4	77.9	62.33
2	2	**41.7**	**69.0**	78.6	**63.10**
3	2	41.0	67.0	79.1	62.37

Table 2
Ablation study for the number of concept prototypes on the MSR-VTT dataset. The initial concepts and output concepts represent M and N in Eq. (5), respectively. "Baseline" denotes the global alignment.

Initial Concepts	Output Concepts	Text->Video			
		R@1	R@5	R@10	Mean
Baseline		43.0	70.4	79.5	64.30
32	1	43.2	70.5	80.0	64.57
32	2	43.8	71.2	80.9	65.30
32	3	**44.7**	**71.7**	81.2	**65.87**
32	4	44.2	71.0	**81.3**	65.50
32	8	43.7	71.2	80.2	65.03
16	3	44.1	70.3	80.7	65.03
48	3	43.9	70.5	80.8	65.07

We use InfoNCE loss [38] to optimize the similarity between text and video modalities:

$$\begin{aligned}L_{CLA} = -\frac{1}{2}(&\frac{1}{B}\sum_{i=1}^B \log\frac{\exp(S_{i,i}/\tau)}{\sum_{j=1}^B \exp(S_{i,j}/\tau)} +\\ &\frac{1}{B}\sum_{j=1}^B \log\frac{\exp(S_{j,j}/\tau)}{\sum_{i=1}^B \exp(S_{i,j}/\tau)}),\end{aligned} \tag{9}$$

where B is the batch size, τ is the learnable temperature coefficient, and $S_{i,j}$ represents the similarity between the i_{th} text and the j_{th} video.

Other Objectives and Inference Strategy As shown in Fig. 3, we attempt to use two other commonly used target losses in text–video cross-modal learning to promote learning of concept-level alignment: (**i**) Video–Text Matching (**VTM**) enhances cross-modal fusion by determining whether a text–video pair is paired. We follow [6,39] and use a hard negative mining strategy. (**ii**) Masked Language Modeling (**MLM**) predicts masked words through a cross-modal decoder based on unmasked text information and video feature information, we follow the text masking strategy in BERT [7]. In addition, most models that use VTM will use a cross-modal decoder to rerank high similarity candidates during inference, which incurs efficiency issues in large-scale retrieval systems as the video representations must be recomputed

Table 3
Effects of VTM and MLM on MSR-VTT dataset. We only use VTM and MLM during training, and only use CLA to calculate similarity in the inference stage for fast retrieval. "Baseline" denotes the global alignment. We report the post-processing time for retrieval in the 1k test setup (1k videos and 1k texts).

CLA	VTM	MLM	Text->Video				
			R@1	R@5	R@10	Mean	Time (s)
One-stage direct calculation of cosine similarity							
Baseline			43.0	70.4	79.5	64.30	0.83
Local Alignment			43.4	70.8	80.9	65.03	1.29
✓			44.7	71.7	81.2	65.87	1.02
✓	✓		44.9	72.0	80.9	65.93	1.02
✓	✓	✓	**46.3**	**73.1**	**81.7**	**67.03**	1.02
Two-stage using VTM re-ranking							
✓	✓		46.1	70.5	81.3	65.97	204.73
✓	✓	✓	45.4	71.9	81.1	66.13	204.73

Table 4
Stage-1 pretraining settings.

config	Kinetics
optimizer	AdamW
optimizer momentum	β_1, β_2 = 0.9,0.95
weight decay	0.05
learning rate schedule	cosine decay
learning rate	1.2e−3
batch size	2048
warmup epochs	40
total epochs	200
mask ratio	80%
input frame	8
drop path	0.1
flip augmentation	yes
augmentation	MultiScaleCrop[0.66, 0.75, 0.875, 1]

Table 5
Stage-2 pretraining settings.

config	5.5M & 17.5M
optimizer	AdamW
optimizer momentum	β_1, β_2 = 0.9,0.95
weight decay	0.02
learning rate schedule	cosine decay
learning rate	6.25e−5
batch size	2560
warmup epochs	1
total epochs	10
mask ratio	50% (image),80% (video)
input frame	4
drop path	0.1
flip augmentation	yes
augmentation	MultiScaleCrop[0.5, 1]

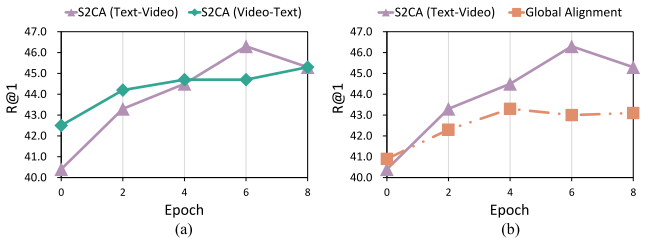

Fig. 4. The impact of S2CA training epochs on performance and the comparison with global alignment.

online for every text query [40]. In order to fully demonstrate the superiority of CLA and achieve faster retrieval that is more in line with real-world applications, we discard the cross-modal decoder during the inference phase and only calculate the similarity between VCAF and TCAF for efficient retrieval.

Table 6
Text–video retrieval finetuning settings.

config	MSRVTT	MSVD	ActivityNet	DiDeMo
optimizer	AdamW			
optimizer momentum	β_1, β_2 = 0.9,0.999			
weight decay	0.02			
learning rate schedule	cosine decay			
learning rate	1e−5	1e−5	2e−5	1e−5
batch size	128			
warmup epochs	1			
total epochs	10	10	20	12
mask ratio	30%			
input frame	12			
max text length	32	64	150	64
drop path	0.2	0.2	0.1	0.1
loss	CLA+VTM+MLM			
flip augmentation	yes			
augmentation	MultiScaleCrop[0.5, 1]			

4. Experiment

4.1. Implementation

Pretraining Datasets. In stage 1, we use Kinetics-700 [41] (**0.57 M**) to perform distillation learning on CLIP. In stage 2, we follow [16] and use two corpora: **(i) 5.5M** Corpus comprises WebVid-2M [3] video–text pairs and CC3M [42] image-text pairs. **(ii) 17.5M** Corpus adds four other image-text datasets: COCO [43], SBU Captions [44], Visual Genome [45] and CC12M [46].

Downstream Tasks and Metrics. We evaluate our method on four mainstream text–video retrieval datasets, including MSR-VTT [47], MSVD [48], Didemo [49] and ActivityNet [50]. **MSR-VTT** contains 10k videos with 200k captions. We follow previous works [51,52] to split into 9K and 1k videos for training and testing. **MSVD** contains 1970 videos with about 120K captions, where the train, validation, and test splits contain 1200, 100 and 670 videos, respectively. **DiDeMo** contains 10,000 videos and 40,000 captions. Following [3], we concatenate all captions to perform paragraph-to-video retrieval. **ActivityNet** contains 20K videos with 100K captions. Following [14], We concatenate all captions to perform paragraph-to-video retrieval. We report the results of Recall@K (R@K, K = 1, 5, 10), and average of Recall@K (Mean) for quantitative evaluation.

Settings. In stage 1, we use ViT-B/16 [21] without [CLS] as the video encoder and CLIP-ViT-B/16 [18] for distillation learning. For Kinetics-700, we sparsely sample 8 frames and masked tokens based on their similarity to CLIP [CLS] tokens as in [10], with a masking rate of 80%. We follow [33] and train for 200 epochs using a 2048 batch size. For the align target, we follow [10] to align the last 6 layers. For the reconstruction target, we only perform on the last layer of features. We set α and β to 1 and 0.05, respectively; In stage 2, we use the distilled video encoder from stage 1 and add BERT-base [7].

Like most methods [10,16,56], we use the first 9 layers of BERT as the text encoder and the last 3 layers as the cross-modal decoder. We follow [16] and sparsely sample 4 frames of video data, treating the image data as a single frame video. According to [10], we mask 50% image tokens and 80% video tokens using random masking [57]. The concept prototypes is randomly initialized from a Gaussian distribution with a zero mean and 0.02 std. We train for 10 epochs using a 2560 batch size, please refer to Tables 4,5,6 for more details. We conducted all experiments on 4 NVIDIA Tesla V100 32 GB GPUs with python 3.8.19, cuda 12.1, pytorch 2.1.2, torchvision 0.16.2, and development environment is pyCharm.

4.2. Ablation study

We verify the effectiveness of our proposed method through experimental analysis. We use K700 pretrained models and further pretrain

Table 7
Comparisons to current state-of-the-art methods on the MSRVTT. * denotes that the model using cross-modal decoder for reranking during inference. "#Pairs" denotes the number of pretraining pairs. CLIP-base models are noted in gray.

Method	#Pairs	Text->Video				Video->Text			
		R@1	R@5	R@10	Mean	R@1	R@5	R@10	Mean
ClipBERT* [14]	5.4M	22.0	46.8	59.9	42.90	–	–	–	–
Frozen [3]	5M	31.0	59.5	70.5	53.67	–	–	–	–
All-in-one* [13]	138M	37.9	68.1	77.1	61.03	37.5	66.1	77.4	60.33
VINDLU* [16]	17M	45.3	69.9	79.6	64.93	–	–	–	–
UMT* [10]	5.6M	46.3	72.7	82.0	67.00	44.4	72.8	80.7	65.97
CLIP4Clip [9]	400M	44.5	71.4	81.6	65.83	42.7	70.9	80.6	64.73
X-Pool [2]	400M	46.9	72.8	82.2	67.30	44.4	73.3	84.0	67.23
DiCoSA [19]	400M	47.5	74.7	83.8	68.67	46.7	75.2	84.3	68.73
Prompt Switch [40]	400M	46.1	72.8	81.8	66.90	44.8	73.7	82.4	66.97
S2CA (Ours)	5.5M	46.3	73.1	81.7	67.03	44.7	73.1	82.0	66.60
	17.5M	**47.7**	**73.4**	**83.1**	**68.07**	**45.8**	**73.4**	**83.1**	**67.43**

Table 8
Comparisons to current state-of-the-art methods on the MSVD. * denotes that the model using cross-modal decoder for reranking during inference. "#Pairs" denotes the number of pretraining pairs. We use the UMT latest modified results on MSVD (see [53,54]).

Method	#Pairs	Text->Video				Video->Text			
		R@1	R@5	R@10	Mean	R@1	R@5	R@10	Mean
LAVENDER* [53]	30M	50.1	79.6	87.2	72.30	–	–	–	–
UMT* [10]	5.6M	47.4	76.8	84.0	69.40	69.1	85.8	92.1	82.33
UMT* [10]	25.6M	50.8	79.7	86.2	72.23	**73.3**	89.6	93.7	**85.53**
CLIP4Clip [9]	400M	46.2	76.1	84.6	68.97	62.0	87.3	92.6	80.63
CenterCLIP [54]	400M	50.6	80.3	88.4	73.10	68.4	90.1	95.0	84.50
X-Pool [2]	400M	47.2	77.4	86.0	70.20	66.4	90.0	94.2	83.53
X-CLIP [8]	400M	50.4	80.6	–	–	66.8	90.4	–	–
DiCoSA [19]	400M	47.4	76.8	86.0	70.07	–	–	–	–
Prompt Switch [40]	400M	47.1	76.9	86.1	70.03	68.5	**91.8**	**95.6**	85.30
S2CA (Ours)	5.5M	49.6	80.0	88.1	72.57	67.5	87.6	93.6	82.90
	17.5M	**52.7**	**82.2**	**89.0**	**74.63**	69.4	88.7	94.2	84.10

Table 9
Text-to-video retrieval performance on the ActivityNet and DiDeMo. * denotes that the model using cross-modal decoder for reranking during inference. "#Pairs" denotes the number of pretraining pairs (see [55]).

Method	#Pairs	ActivityNet				DiDeMo			
		R@1	R@5	R@10	Mean	R@1	R@5	R@10	Mean
LAVENDER* [53]	5.5M	–	–	–	–	47.4	74.7	82.4	68.17
CLIP4Clip [9]	400M	40.5	72.4	–	–	43.4	70.2	80.6	64.73
CenterCLIP [54]	400M	46.2	77.0	87.6	–	–	–	–	–
X-CLIP [8]	400M	46.2	75.5	–	–	47.8	79.3	–	–
CLIP-ViP [15]	500M	–	–	–	–	49.4	74.9	84.5	69.60
DiCoSA [19]	400M	42.1	73.6	84.6	66.77	45.7	74.6	83.5	67.93
HBI [55]	400M	42.2	73.0	84.6	66.60	46.9	74.9	82.7	68.17
S2CA (Ours)	5.5M	46.2	76.0	87.0	69.73	51.2	78.3	85.3	71.60
	17.5M	**49.6**	**79.4**	**89.1**	**72.70**	**54.3**	**82.3**	**89.4**	**75.33**

it for 10 epochs on 5.5M Corpus. We report retrieval performance on MSR-VTT. For the ablation experiments of the **Distillation Framework**, considering efficiency, we trained the model on K700 with a batch size of 512 for 20 epochs. For all other experiments, we trained the full 200-epoch model on K700 with a batch size of 2048 as the basis.

Distillation Framework. Table 1 represents the impact of Reconstruction Target and Adaptor module on retrieval performance in distillation framework of stage 1. Like UMT [10], we observe that simply adding Reconstruct Target (the second row in Table 1) resulted in a decrease in average Recall accuracy. When we use the Adaptor module (lightweight two-layer Transformer encoder layers), the R@1, R@5 are significantly improved (*i.e.*, from 39.7 to 41.7 and from 68.0 to 69.0). Finally, we set the Adaptor and Decoder to be lightweight 2-layer Transformer encoder.

The Number of Concept Prototypes. Table 2 represents the impact of the number of initial concept prototypes and output Conceptual Aggregated Features (CAF) (*i.e.*, M and N in Eq. (5)). Intuitively, a larger initial concept prototypes provide more learnable clustering space for the model, and the number of output CAF controls the number of clusters that the text and video are divided into. We observe that excessive output CAF will reduce performance. Too many or too few initial concept prototypes can lead to performance degradation, possibly due to the difficulty of optimizing too many concept prototypes and too few concept prototypes make it difficult to fully leverage clustering effects. In addition, regardless of the configuration, the significant improvement after adding the concept prototypes compared to the baseline demonstrates the superiority of our framework. Based on the significant improvement of R@1, we set concept prototypes $M = 32$ and output CAF $N = 3$.

Effects of CLA, VTM and MLM. During the training process, VTM and MLM serve as auxiliary supervisions to promote text–video semantic alignment. As shown in Table 3, although we only use VTM and MLM for training, and we do not use VTM for reranking during inference, retrieval performance is improved from 44.7 to 46.3 at R@1. In addition, compared to baseline, CLA brings significant improvements

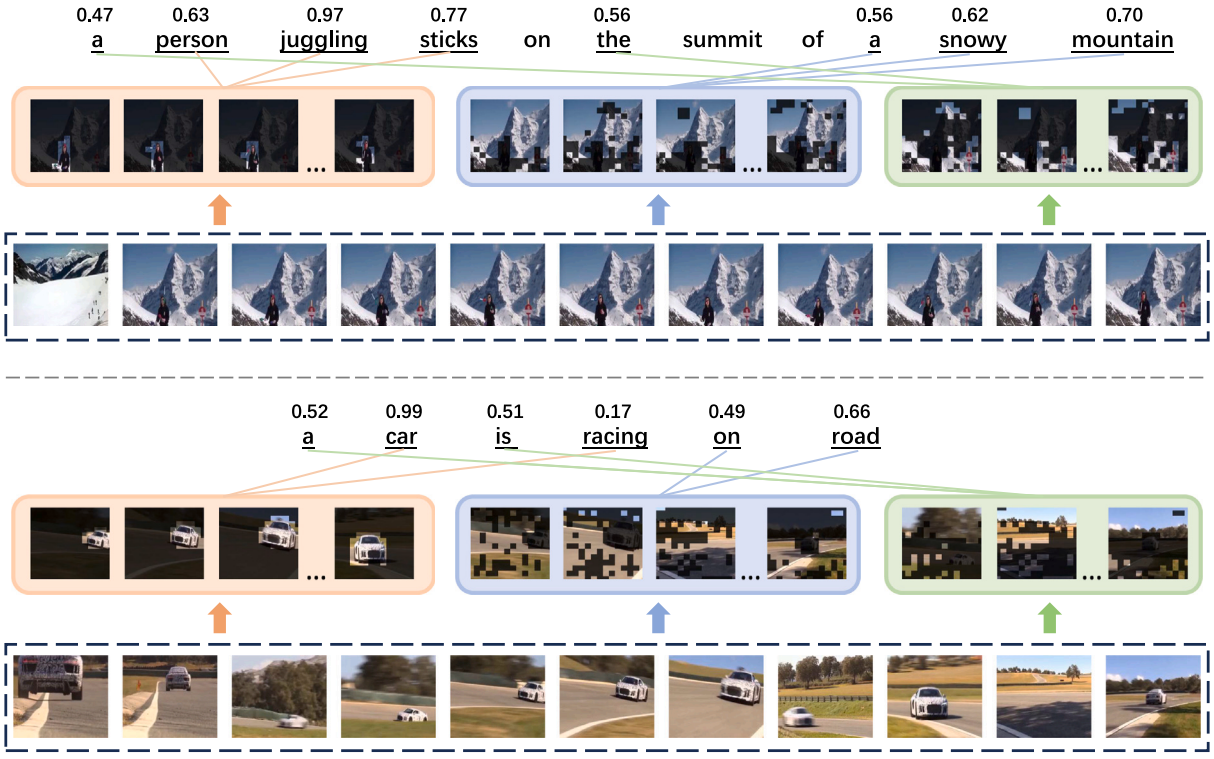

Fig. 5. Visualization of Concept Allocation. Red, blue, and green represent different CAF, while lines represent the most similar words and indicate the corresponding similarity.

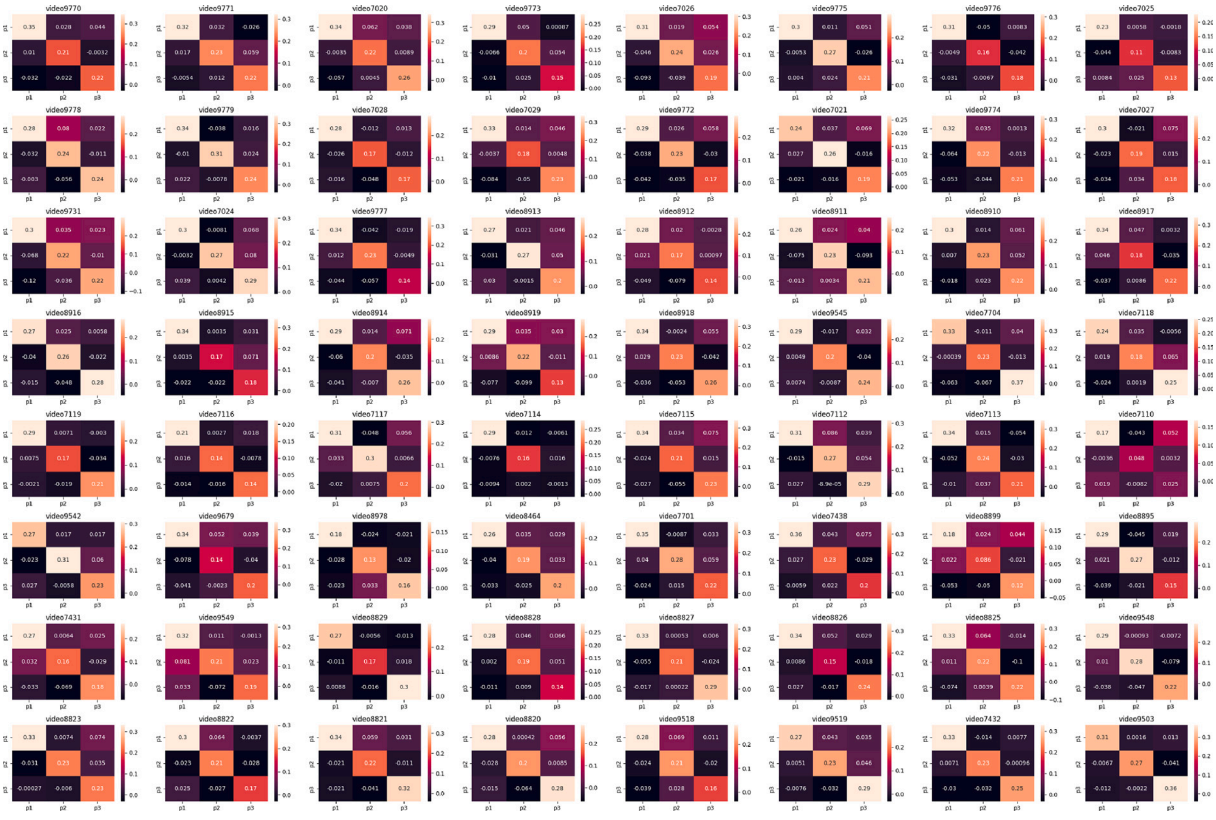

Fig. 6. Heat map of pairwise cosine similarity between TCAF and VCAF. Light colors indicate high similarity, while dark colors indicate low similarity.

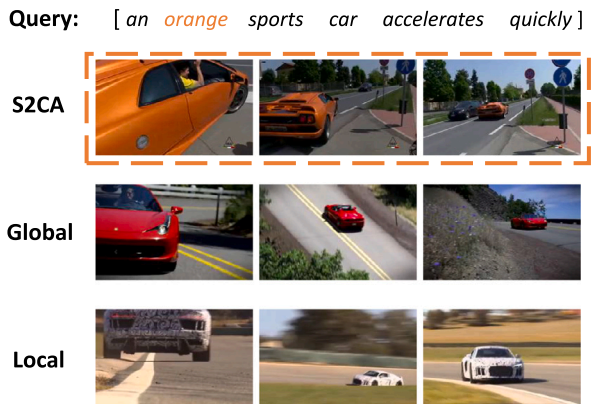

Fig. 7. Comparison of retrieval performance with other methods. "Global" denotes the global alignment."Local" denotes the local alignment. The orange box indicates the correct video.

(43.0 to 44.7 at R@1). Please note that the effectiveness of our CLA is primarily compared with the baseline (global alignment), which is a method based on VTC. CLA, in essence, is a special form of VTC, while VTM and MLM are two frameworks that differ from VTC.

The Impact of Training Epochs. We demonstrate the changes in retrieval performance during the S2CA training process in Fig. 4(a) and compare it with global alignment in Fig. 4(b). S2CA significantly outperformed global alignment in most epochs.

Comparison of Retrieval Time. As shown in Table 3, we compare the performance and post-retrieval processing time of various methods. For one-stage methods, they generally refer to retrieval using the cosine similarity between text and video (optimized through VTC). These methods offer faster retrieval speeds. Among them, global alignment only calculates the similarity between one text feature and one video feature, thus achieving the fastest speed. Local alignment calculates the similarity between all words and all frames, leading to slower speed. Our S2CA needs to compute the one-to-one similarity between TCAF and VCAF, with its speed lying between the two but outperforming them in terms of performance (note that in the one-stage approach, S2CA only uses VTM and MLM for auxiliary training, and VTM re-ranking is not used during inference). For two-stage methods, we follow other works [10,16] where S2CA first uses CLA for initial ranking and then applies VTM for reranking. This requires the query text and all videos to predict similarity through a cross-modal decoder, resulting in slow speed that is not suitable for practical applications, and suboptimal performance. Ultimately, S2CA adopts the one-stage method.

4.3. Comparison with the state-of-the-arts

We compare the proposed S2CA with other methods on four benchmarks. In Tables 7 and 8, we show the results of our method on the MSR-VTT and MSVD. For the 5.5M datasets, S2CA only efficiently calculates similarity during inference, but exceed the methods using VTM for reranking such as VINDLU and UMT on MSR-VTT. S2CA significantly surpasses both the models that utilize VTM for reranking and the models based on CLIP on MSVD (**52.7%** (+1.9%)). Table 9 shows text-to-video retrieval results on the ActivityNet and DiDeMo. We clearly surpass the other models. S2CA achieves consistent improvements across different datasets, which demonstrates the effectiveness and generalization ability of our method.

4.4. Visualization

Visualization of CAF for Video and Text. We firstly show the visualization of conceptual aggregated features of text and video in Fig. 5, we present videos with relatively small temporal changes (top) and videos with more drastic changes (bottom). It can be observed that the concept marked in red pays more attention to prominent objects in the video, while the concept marked in blue pays more attention to objects with smaller changes in the video, such as the background. The green concept is visually difficult to explain, but from the text, we can see that it pays more attention to some adverbs. In addition, as can be seen from the video example below, S2CA can easily track the rapidly moving prominent object in the video, namely the white car driving at top speed, and can correspond with the text, which reflects the effectiveness of the concept decoupling proposed in this paper.

Visualization of Similarity between TCAF and VCAF. As shown in Fig. 6, we visualize the heat map obtained from the pairwise cosine similarity of three TCAF and three VCAF. It can be intuitively and clearly seen that the conceptual aggregated features of the text and the video correspond one-to-one. For example, the first TCAF has the highest similarity with the first VCAF, and the second TCAF has the highest similarity with the second VCAF. Therefore, the conceptual aggregated features of the text and the video are corresponded one-to-one to achieve concept-level alignment.

More Visualization of VCAF. To further demonstrate the effectiveness of concept decoupling based on concept prototypes, more examples are shown in Fig. 8, we have selected multiple distributed videos to demonstrate the robustness of decoupling, including single person videos, hand drawn videos, anime videos, multiplayer videos, and videos with drastic temporal changes. As discussed above, the first VCAF places more emphasis on prominent physical targets, while the second VCAF places more emphasis on background information. As can be clearly seen from the example from the second to last video, when there are only cars in the video, the VCAF can easily focus on prominent objects such as cars, even if the target occupies a small volume. As the car gradually disappears, the VCAF can focus on new prominent objects such as dogs.

Retrieval Result. We compare the retrieval performance with two commonly used retrieval training frameworks, global alignment and local alignment, as shown in Fig. 7. Our S2CA pays more attention to the detailed features of retrieval, namely the "orange car".

4.5. Discussion

Retrieval Performance. As shown in Tables 3, 7, 8 and 9, despite being pretrained on only 17.5M data, S2CA is still able to outperform models that finetuning CLIP, which was pretrained on 400M data. Additionally, S2CA's efficiency stems from the absence of complex architectures and time-consuming two-stage retrieval, making it both flexible and effective. Notably, on the MSVD dataset, S2CA significantly surpasses all methods to achieve state-of-the-art (SOTA) performance. This might be attributed to the characteristics of the MSVD dataset, where text descriptions often adhere to a standard format of "who did what to what where", such as the following three descriptions for a single video: "a man demonstrating how to clean a flower", "a man brushes off the roots of a sunflower", and "a man is brushing dirt off a sunflower". In contrast, descriptions for the same video in MSR-VTT vary considerably. This may indicate that concept-level alignment may be more suitable for text queries with more fixed formats. S2CA's performance also demonstrates the feasibility of distilling CLIP for text–video retrieval models, a direction that has yet to be extensively explored.

However, we also noticed that the improvement in retrieval performance of S2CA on the 17.5M dataset over the 5.5M dataset is not particularly significant. This could be attributed to the fact that while S2CA, as a one-stage retrieval model, ensures efficiency, it sacrifices the benefits of fine-grained two-stage retrieval using VTM. As shown in the experiments in Table 3, we found that there seems to be a certain conflict between CLA and VTM, as using both for retrieval does not lead

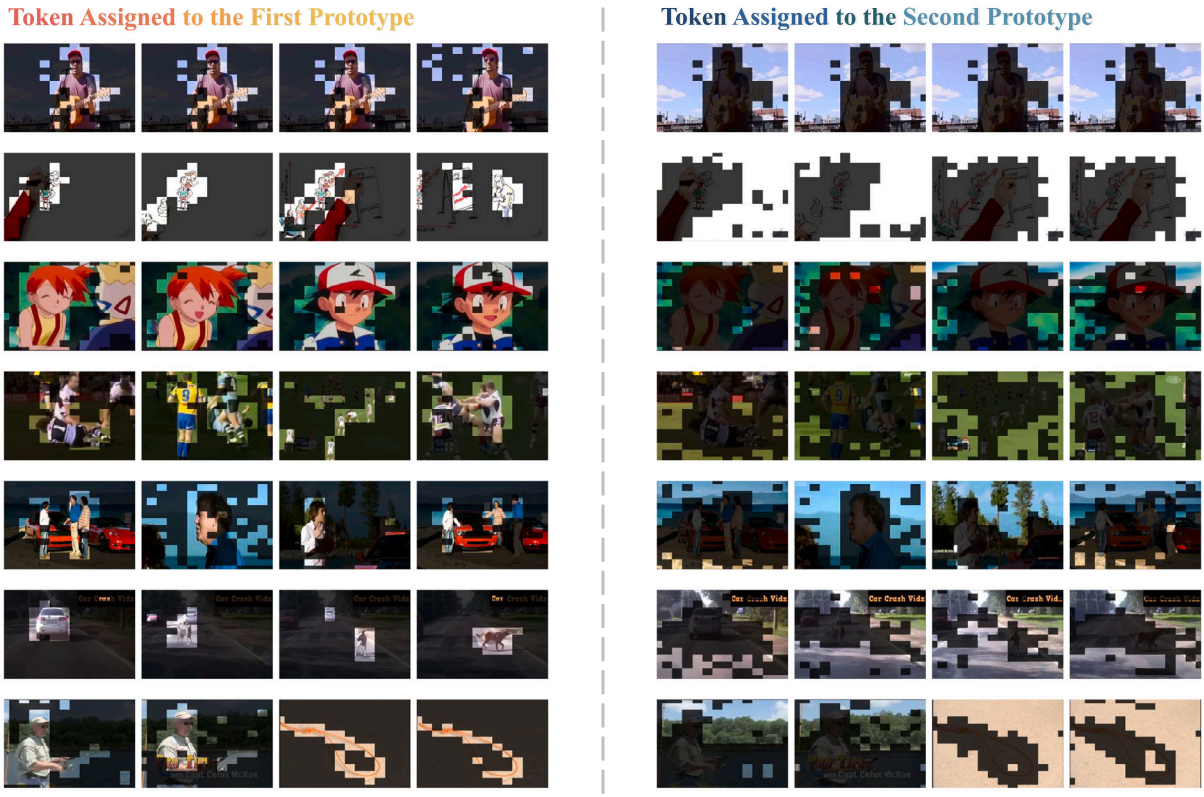

Fig. 8. More Visualization of VCAF.

to a notable performance boost. Further work and research are needed to explain the underlying reasons for this phenomenon.

Concept-level Decoupling. We would like to emphasize that the core of S2CA lies in decoupling different concepts from text and video, achieving concept decoupling and concept-level alignment. As discussed in the introduction, we believe that concept-level alignment is more intuitive than conventional global alignment and local alignment. In simple terms, "liking someone does not necessarily mean liking everything about them". However, there is still relatively little work in the field of text–video retrieval that focuses on concept-level alignment, even though the methods for achieving it could be endless. For instance, DiCoSA [19] attempts decoupling with a simple MLP. We consider the research on concept-level alignment a promising direction.

Furthermore, initially, we assumed that the optimal number of CAF might be 5, 6, or more. However, through experimentation, we surprisingly found that the optimal number of decoupled CAF is 3, significantly fewer than anticipated. Yet, what exactly is the optimal number of decoupled CAF? Can it be explained theoretically? These remain open questions. Lastly, from the visualization of CAF in Fig. 5, we observe that among the three CAF, one focuses more on the subject matter in the foreground, another emphasizes background information, while the last one possibly attends to less intuitive adverbial information. The heat map in Fig. 6 also demonstrates that S2CA can indeed deconstruct the semantic features of three distinct concepts within videos and texts, successfully achieving the concept decoupling and concept-level alignment we have pointed out.

5. Conclusion

In this paper, we propose the Shared Concept Prototypes and Concept-level Alignment (S2CA), which balances the align and reconstruct target and improves the effectiveness of distillation learning and establishes a one-to-one correspondence between text and video concepts, achieving concept-level alignment. S2CA achieves the decoupling of heterogeneous concept from texts and videos, which is more reasonable and intuitive compared to traditional methods. We demonstrate the feasibility of concept decoupling through abundant visualizations. Additionally, S2CA does not introduce complex modules or time-consuming two-stage retrieval, ensuring flexibility and achieving remarkable performance with high efficiency that is more suitable for real-world applications, particularly achieving state-of-the-art (SOTA) results on the MSVD dataset. However, we also observe that when the amount of pretraining data increases, the improvement in retrieval performance of S2CA is not particularly significant, and there is a lack of more theoretical justification for determining the number of decoupled concepts.

Lastly, we believe that decoupling concepts from texts and videos and then performing fine-grained concept-level alignment based on these concepts represents a promising research direction in text–video retrieval. There are diverse methods for concept decoupling and concept-level alignment, and S2CA merely presents a feasible framework. More work can be carried out in this direction. In the future, we will explore more ways to achieve concept decoupling, investigate the rules governing data volume limitations and the determination of the number of decoupled concepts, or seek a dynamic approach to adaptively determine the number of decoupled concepts.

CRediT authorship contribution statement

Yuxiao Li: Writing – original draft, Visualization, Resources, Methodology, Formal analysis, Conceptualization. **Yu Xin:** Writing – review & editing, Supervision, Funding acquisition. **Jiangbo Qian:** Supervision. **Yihong Dong:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We acknowledge the support of the Natural Science Foundation of Zhejiang Province, China (Grant No. LY22F020001, No. LZ20F020001), the 3315 Plan Foundation of Ningbo (Grant No. 2019B-18-G), China Natural Science Foundation under Grant 62271274 and the support of Research and Application of A Multi Billion Parameter Monitoring Video Model for domestic full stack AI infrastructure, China (Grant No. 20242004).

Data availability

Data will be made available on request.

References

[1] Y. Chen, J. Wang, L. Lin, Z. Qi, J. Ma, Y. Shan, Tagging before alignment: Integrating multi-modal tags for video-text retrieval, 2023, arXiv:2301.12644.
[2] S.K. Gorti, N. Vouitsis, J. Ma, K. Golestan, M. Volkovs, A. Garg, G. Yu, X-pool: Cross-modal language-video attention for text-video retrieval, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5006–5015.
[3] M. Bain, A. Nagrani, G. Varol, A. Zisserman, Frozen in time: A joint video and image encoder for end-to-end retrieval, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 1728–1738.
[4] S. Chen, Y. Zhao, Q. Jin, Q. Wu, Fine-grained video-text retrieval with hierarchical graph reasoning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10638–10647.
[5] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, J. Sivic, Howto100m: Learning a text-video embedding by watching hundred million narrated video clips, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2630–2640.
[6] J. Li, R. Selvaraju, A. Gotmare, S. Joty, C. Xiong, S.C.H. Hoi, Align before fuse: Vision and language representation learning with momentum distillation, Adv. Neural Inf. Process. Syst. 34 (2021) 9694–9705.
[7] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.
[8] Y. Ma, G. Xu, X. Sun, M. Yan, J. Zhang, R. Ji, X-CLIP: End-to-end multi-grained contrastive learning for video-text retrieval, in: Proceedings of the 30th ACM International Conference on Multimedia, MM '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 638–647, http://dx.doi.org/10.1145/3503161.3547910.
[9] H. Luo, L. Ji, M. Zhong, Y. Chen, W. Lei, N. Duan, T. Li, Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning, Neurocomputing 508 (2022) 293–304.
[10] K. Li, Y. Wang, Y. Li, Y. Wang, Y. He, L. Wang, Y. Qiao, Unmasked teacher: Towards training-efficient video foundation models, 2023, arXiv preprint arXiv:2303.16058.
[11] F. Shu, B. Chen, Y. Liao, S. Xiao, W. Sun, X. Li, Y. Zhu, J. Wang, S. Liu, Masked contrastive pre-training for efficient video-text retrieval, 2022, arXiv preprint arXiv:2212.00986.
[12] J. Lei, T.L. Berg, M. Bansal, Revealing single frame bias for video-and-language learning, 2022, arXiv preprint arXiv:2206.03428.
[13] J. Wang, Y. Ge, R. Yan, Y. Ge, K.Q. Lin, S. Tsutsui, X. Lin, G. Cai, J. Wu, Y. Shan, et al., All in one: Exploring unified video-language pre-training, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 6598–6608.
[14] J. Lei, L. Li, L. Zhou, Z. Gan, T.L. Berg, M. Bansal, J. Liu, Less is more: Clipbert for video-and-language learning via sparse sampling, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 7331–7341.
[15] H. Xue, Y. Sun, B. Liu, J. Fu, R. Song, H. Li, J. Luo, Clip-vip: Adapting pre-trained image-text model to video-language representation alignment, 2022, arXiv preprint arXiv:2209.06430.
[16] F. Cheng, X. Wang, J. Lei, D. Crandall, M. Bansal, G. Bertasius, Vindlu: A recipe for effective video-and-language pretraining, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 10739–10750.
[17] X. Wang, L. Zhu, Y. Yang, T2vlad: global-local sequence alignment for text-video retrieval, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 5079–5088.
[18] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International Conference on Machine Learning, PMLR, 2021, pp. 8748–8763.
[19] P. Jin, H. Li, Z. Cheng, J. Huang, Z. Wang, L. Yuan, C. Liu, J. Chen, Text-video retrieval with disentangled conceptualization and set-to-set alignment, 2023, arXiv preprint arXiv:2305.12218.
[20] J. Xu, S. De Mello, S. Liu, W. Byeon, T. Breuel, J. Kautz, X. Wang, Groupvit: Semantic segmentation emerges from text supervision, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 18134–18144.
[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16×16 words: Transformers for image recognition at scale, 2020, arXiv preprint arXiv:2010.11929.
[22] E. Jang, S. Gu, B. Poole, Categorical reparameterization with gumbel-softmax, 2016, arXiv preprint arXiv:1611.01144.
[23] C.J. Maddison, A. Mnih, Y.W. Teh, The concrete distribution: A continuous relaxation of discrete random variables, 2016, arXiv preprint arXiv:1611.00712.
[24] S. Zhang, F. Zhu, J. Yan, R. Zhao, X. Yang, Zero-cl: Instance and feature decorrelation for negative-free symmetric contrastive learning, in: International Conference on Learning Representations, 2021.
[25] S. Zhang, M. Liu, J. Yan, H. Zhang, L. Huang, X. Yang, P. Lu, M-mix: Generating hard negatives via multi-sample mixing for contrastive learning, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 2461–2470.
[26] S. Zhang, L. Qiu, F. Zhu, J. Yan, H. Zhang, R. Zhao, H. Li, X. Yang, Align representations with base: A new approach to self-supervised learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16600–16609.
[27] S. Zhang, F. Zhu, R. Zhao, J. Yan, Patch-level contrasting without patch correspondence for accurate and dense contrastive representation learning, 2023, arXiv preprint arXiv:2306.13337.
[28] Q. Wang, Y. Zhang, Y. Zheng, P. Pan, X.-S. Hua, Disentangled representation learning for text-video retrieval, 2022, arXiv preprint arXiv:2203.07111.
[29] Z. Yang, Z. Li, M. Shao, D. Shi, Z. Yuan, C. Yuan, Masked generative distillation, in: European Conference on Computer Vision, Springer, 2022, pp. 53–69.
[30] R. Wang, D. Chen, Z. Wu, Y. Chen, X. Dai, M. Liu, L. Yuan, Y.-G. Jiang, Masked video distillation: Rethinking masked feature modeling for self-supervised video representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 6312–6322.
[31] Z. Yang, Z. Li, A. Zeng, Z. Li, C. Yuan, Y. Li, Vitkd: Practical guidelines for vit feature knowledge distillation, 2022, arXiv preprint arXiv:2209.02432.
[32] Y. Bai, Z. Wang, J. Xiao, C. Wei, H. Wang, A.L. Yuille, Y. Zhou, C. Xie, Masked autoencoders enable efficient knowledge distillers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 24256–24265.
[33] Z. Tong, Y. Song, J. Wang, L. Wang, Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training, Adv. Neural Inf. Process. Syst. 35 (2022) 10078–10093.
[34] T. Radhika, A. Chandrasekar, V. Vijayakumar, Q. Zhu, Analysis of Markovian jump stochastic Cohen–Grossberg BAM neural networks with time delays for exponential input-to-state stability, Neural Process. Lett. 55 (8) (2023) 11055–11072.
[35] T. Radhika, A. Chandrasekar, V. Vijayakumar, Finite-time H_∞ synchronization of semi-Markov jump neural networks with two delay components with stochastic sampled-data control, Bulletin des Sciences Mathématiques 195 (2024) 103482.
[36] A. Chandrasekar, T. Radhika, Q. Zhu, Further results on input-to-state stability of stochastic Cohen–Grossberg BAM neural networks with probabilistic time-varying delays, Neural Process. Lett. (2022) 1–23.
[37] A. Van Den Oord, O. Vinyals, et al., Neural discrete representation learning, Adv. Neural Inf. Process. Syst. 30 (2017).
[38] A.v.d. Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, 2018, arXiv preprint arXiv:1807.03748.
[39] A. Rohrbach, A. Torabi, M. Rohrbach, N. Tandon, C. Pal, H. Larochelle, A. Courville, B. Schiele, Movie description, Int. J. Comput. Vis. 123 (2017) 94–120.
[40] C. Deng, Q. Chen, P. Qin, D. Chen, Q. Wu, Prompt switch: Efficient CLIP adaptation for text-video retrieval, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 15648–15658.
[41] J. Carreira, E. Noland, C. Hillier, A. Zisserman, A short note on the kinetics-700 human action dataset, 2019, arXiv preprint arXiv:1907.06987.
[42] P. Sharma, N. Ding, S. Goodman, R. Soricut, Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 2556–2565.

[43] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer, 2014, pp. 740–755.
[44] V. Ordonez, G. Kulkarni, T. Berg, Im2text: Describing images using 1 million captioned photographs, Adv. Neural Inf. Process. Syst. 24 (2011).
[45] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D.A. Shamma, et al., Visual genome: Connecting language and vision using crowdsourced dense image annotations, Int. J. Comput. Vis. 123 (2017) 32–73.
[46] S. Changpinyo, P. Sharma, N. Ding, R. Soricut, Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3558–3568.
[47] J. Xu, T. Mei, T. Yao, Y. Rui, Msr-vtt: A large video description dataset for bridging video and language, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5288–5296.
[48] D. Chen, W.B. Dolan, Collecting highly parallel data for paraphrase evaluation, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, pp. 190–200.
[49] L. Anne Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, B. Russell, Localizing moments in video with natural language, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5803–5812.
[50] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, J. Carlos Niebles, Dense-captioning events in videos, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 706–715.
[51] V. Gabeur, C. Sun, K. Alahari, C. Schmid, Multi-modal transformer for video retrieval, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16, Springer, 2020, pp. 214–229.
[52] Y. Yu, J. Kim, G. Kim, A joint sequence fusion model for video question answering and retrieval, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 471–487.
[53] L. Li, Z. Gan, K. Lin, C.-C. Lin, Z. Liu, C. Liu, L. Wang, Lavender: Unifying video-language understanding as masked language modeling, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 23119–23129.
[54] S. Zhao, L. Zhu, X. Wang, Y. Yang, Centerclip: Token clustering for efficient text-video retrieval, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 970–981.
[55] P. Jin, J. Huang, P. Xiong, S. Tian, C. Liu, X. Ji, L. Yuan, J. Chen, Video-text as game players: Hierarchical banzhaf interaction for cross-modal representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 2472–2482.
[56] J. Li, R. Selvaraju, A. Gotmare, S. Joty, C. Xiong, S.C.H. Hoi, Align before fuse: Vision and language representation learning with momentum distillation, Adv. Neural Inf. Process. Syst. 34 (2021) 9694–9705.
[57] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked autoencoders are scalable vision learners, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16000–16009.

Yuxiao Li is pursuing a master's degree in Faculty of Electrical Engineering and Computer Science, Ningbo University, China. His research interests include deep learning, multimodal learning, text-image retrieval, recommendation systems, diagnostic analysis and text–video retrieval.

Yu Xin received the Ph.D. degree in computer science and technology from Harbin Engineering University, China. He is currently an Associate Professor with the Faculty of Electrical Engineering and Computer Science, Ningbo University, China. His current research interests include multiple classifier and prediction systems, processing and modeling of uncertainty in predictive modeling, recommendation systems, diagnostic analysis, and decision support systems.

Jiangbo Qian received the Ph.D. degree in computer science from Southeast University, China, in 2006. He was a Visiting Scholar with the Department of Computer and Information Science, The University of Michigan–Dearborn, USA. He is currently a Professor with the Faculty of Electrical Engineering and Computer Science, Ningbo University, China. His research interests include database management, streaming data processing, deep learning, computer vision, and hardware/software codesign.

Yihong Dong received the Ph.D. degree in computer science from Zhejiang University, China, in 2007. He is currently a Professor with the Faculty of Electrical Engineering and Computer Science, Ningbo University, China. His research interests include big data, data mining, and artificial intelligence.