CLIPPING: Distilling CLIP-Based Models with a Student Base for Video-Language Retrieval

Renjing Pei, Jianzhuang Liu, Weimian Li,
Bin Shao, Songcen Xu, Peng Dai, Juwei Lu, Youliang Yan
Huawei Noah's Ark Lab
{peirenjing,liu.jianzhuang,liweimian,shaobin3,
xusongcen,peng.dai,juwei.lu,yanyouliang}@huawei.com

Abstract

Pre-training a vision-language model and then fine-tuning it on downstream tasks have become a popular paradigm. However, pre-trained vision-language models with the Transformer architecture usually take long inference time. Knowledge distillation has been an efficient technique to transfer the capability of a large model to a small one while maintaining the accuracy, which has achieved remarkable success in natural language processing. However, it faces many problems when applying KD to the multi-modality applications. In this paper, we propose a novel knowledge distillation method, named CLIPPING[1]*, where the plentiful knowledge of a large teacher model that has been fine-tuned for video-language tasks with the powerful pre-trained CLIP can be effectively transferred to a small student only at the fine-tuning stage. Especially, a new layer-wise alignment with the student as the base is proposed for knowledge distillation of the intermediate layers in CLIPPING, which enables the student's layers to be the bases of the teacher, and thus allows the student to fully absorb the knowledge of the teacher. CLIPPING with MobileViT-v2 as the vision encoder without any vision-language pre-training achieves **88.1%–95.3%** of the performance of its teacher on three video-language retrieval benchmarks, with its vision encoder being **19.5x** smaller. CLIPPING also significantly outperforms a state-of-the-art small baseline (ALL-in-one-B) on the MSR-VTT dataset, obtaining relatively **7.4%** performance gain, with **29%** fewer parameters and **86.9%** fewer flops. Moreover, CLIPPING is comparable or even superior to many large pre-training models.*

[1]In this paper, CLIPPING means cutting something to make it smaller through distilling.

1. Introduction

Recently, pre-training a vision-language model and then fine-tuning it on downstream tasks are a popular paradigm [12, 16, 20, 21, 30]. Pre-trained vision-language models (PVLMs) have achieved great success in many multi-modality tasks (e.g., video-text retrieval). However, PVLMs with the Transformer architecture (especially the vision stream) usually consume a huge amount of computation, making them difficult to be deployed on edge devices such as mobile phones. Recent small models [8, 14, 19, 22, 27, 40] show that combining Convolutional Neural Networks (CNNs) and Transformers as a hybrid architecture gets the best of both architectures, but the overall performance of these works is still far from satisfactory when compared to large pre-training models. Apparently, knowledge distillation (KD) [15] is an efficient technique to transfer the capability of a large model to a small one while maintaining the accuracy, which has achieved remarkable success in natural language processing (NLP) [17, 18]. In NLP, the knowledge distillation methods are usually performed at both the pre-training and the fine-tuning stages. However, the collection of the pre-training data for the pre-training knowledge distillation cost huge manpower in multi-modality applications (e.g., the pre-training dataset of CLIP is hard to obtain). Therefore, an efficient knowledge distillation method need to be explored for multi-modality applications. To this end, we propose a novel knowledge distillation method, named CLIPPING, where the plentiful knowledge of a large teacher model Clip4clip [26] that has been fine-tuned for video-language tasks with the powerful pre-trained CLIP [30] can be effectively transferred to a small student only at the fine-tuning stage. The contributions of CLIPPING are summarized below:

1) We introduce an efficient approach to distill both the vision knowledge and the cross-modality knowledge from teacher to a small model at the fine-tuning stage. The resulting model shows strong performance on multi-modality

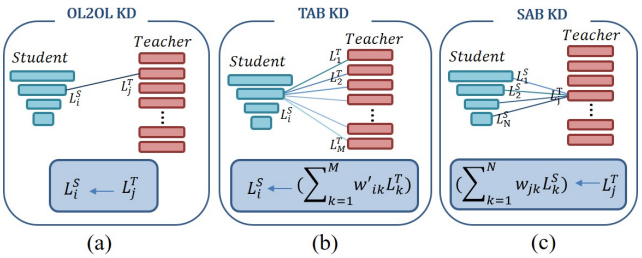

Figure 1. (a) and (b) are previous methods of intermediate features' knowledge distillation. L_j^T, $j = 1, 2, ..., M$, and L_i^S, $i = 1, 2, ..., N$, are the outputs of the j^{th} and i^{th} layers of the teacher and the student, respectively. OL2OL is the most common One-Layer-to-One-Layer KD, where it selects some layers of the teacher and then distills the layers into the student one by one (e.g., from L_j^T to L_i^S). Teacher-As-Base (TAB) KD uses all the teacher's layers to supervise each layer in the student, where each layer of the student (e.g., L_i^S) learns the knowledge from all the selected layers of the teacher ($\sum_{k=1}^{M} w'_{ik} L_k^T$, where w'_{ik}, $k = 1, 2, ..., M$, are the knowledge selection weights with $\sum_{k=1}^{M} w'_{ik} = 1$). (c) Our Student-As-Base (SAB) KD enables each of the teacher's layers to pass its knowledge to all the student's layers (e.g., $L_j^T \rightarrow \sum_{k=1}^{N} w_{jk} L_k^S$, $\sum_{k=1}^{N} w_{jk} = 1$). More detailed analysis is provided in Section 3.1.

text-video retrieval task.

2) We propose a new layer-wise alignment scheme, called Student-As-Base (SAB), for knowledge distillation of the intermediate layers from the Transformer to the CNN in CLIPPING, where the student's layers can be regarded as the bases of the teacher's feature space, forcing the student model to full absorb the knowledge of the teacher. In our experience, the SAB layer-wise alignment significantly surpasses the previous knowledge distillation methods. We believe it will become a popular paradigm for knowledge distillation of intermediate features.

3) We present an effective cross-modal knowledge distillation, which includes knowledge from both the global and local video-caption distributions. We use the video-caption distributions of the teacher to guide the training of the student. Besides, the student can also benefit from the powerful pre-trained CLIP that obtains certain local frame-word attention ability via learning from massive data. Experiments show that this pre-training knowledge can be effectively transferred to the student when jointly trained with our SAB KD only at the fine-tuning stage.

4) CLIPPING with MobileViT-v2 [27] as the vision encoder without any vision-language pre-training achieves 91.5%–95.3% of the performance of its teacher on MSR-VTT video-language retrieval benchmark, with its vision encoder being 19.5x smaller. CLIPPING also significantly outperforms a state-of-the-art small baseline (ALL-in-one-B) on the MSR-VTT dataset, obtaining relatively 7.4% performance gain, with 29% fewer parameters and 86.9% fewer flops. Moreover, CLIPPING is comparable or even superior to many large pre-training models.

2. Relative Work

Vision-Language Retrieval. Learning from web-collected image-text data, large-scale Vision-Language Pre-training (VLP) models such as CLIP [30] have recently demonstrated great success across various downstream tasks. Nowadays, models such as Clip4clip [26] and MDMMT [11] extended from the pre-trained model CLIP keep appearing for vision-language retrieval. There are also some end-to-end trainable models [3, 37], which are designed to take advantage of both large-scale image and video captioning datasets. All-in-one [34] is the first work to consider both efficiency and performance for video-language retrieval tasks. It introduces a unified backbone that enables the representation learning of both video-text multimodal and unimodal inputs.

Knowledge Distillation. Knowledge distillation (KD) [15] is a simple yet effective technique to improve the performance of a learning model. Earlier works transfer knowledge embedded in the "logits" learned in a large teacher model to a small student model without sacrificing much performance. Recent works [5, 6] use multiple layers of the teacher to supervise each layer in the student, where each layer of the student learns the knowledge from multiple layers of the teacher. [23] proposes a target-aware Transformer and enables the student to mimic each spatial component of the teacher in each distilled layer to boost the student's performance. For multi-modality KD, [36] designs a fusion-encoder model as the teacher and introduces cross-modal attention knowledge to train the dual-encoder student model. The distillation objective is applied at both the pre-training and the fine-tuning stages and helps the dual-encoder model learn interactions of different modalities. TinyBert [17] also introduces a two-stage learning framework that performs Transformer distillation at both the pre-training and the task-specific fine-tuning stages. In this paper, we also focus on KD for transferring the knowledge of a pre-training vision-language model to a small one but only at the fine-tuning stage.

Transformers and CNNs. Usually, the pre-training model is a Transformer-based model [10, 25] and the small model is a CNN-based or hybrid architecture [19, 27]. For KD, most existing methods distill knowledge either from a Transformer to another Transformer (T2T) or from a CNN to another CNN (C2C) that computes the loss in the OL2OL or TAB style (see Fig. 1). To the best of our knowledge, KD from a Transformer to a CNN (T2C) has not been explored yet. The most related work [7] distills knowledge from a CNN to a Transformer (C2T) in the OL2OL way. However, previous works (e.g., [31]) have investigated the internal representation structures of ViTs and CNNs, and found striking differences between the two models, such as

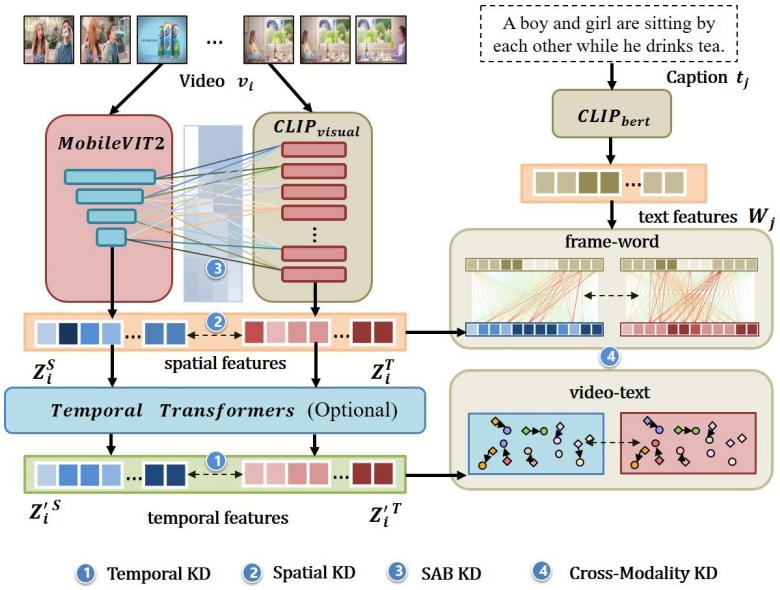

Figure 2. Overview of CLIPPING. There are mainly four knowledge distillation (KD) parts: (1) Temporal KD. (2) Spatial KD. (3) SAB (Student-As-Base) KD. (4) Cross-modality KD. The CLIP's vision encoder is a Transformer. The temporal Transformer is an optional module for CLIPPING.

ViTs having highly similar representations throughout the model's layers, while CNNs showing obvious distinction of representations between lower and higher layers [31]. Considering such striking differences between ViTs and CNNs, the previous KD (OL2OL and TAB) may not be good distillation ways for our T2C task, which is verified by our experiments.

3. Methodology

We propose an efficient approach to distill both the vision knowledge and the cross-modality knowledge from Clip4clip to a small model with MobileViT-v2 as its vision encoder for text-video retrieval task, which can get the best from both sides: powerful multimodal representations of CLIP and efficient mobile vision Transformer of MobileViT-v2. The overall architecture is illustrated in Fig. 2.

3.1. Analysis of TAB and SAB

We first show the advantage of our SAB KD over traditional TAB KD. For simplicity, suppose $M = N = 2$. Then, for SAB,

$$\begin{cases} L_1^T = w_{11}L_1^S + w_{12}L_2^S,\ w_{11} + w_{12} = 1 \\ L_2^T = w_{21}L_1^S + w_{22}L_2^S,\ w_{21} + w_{22} = 1 \end{cases}, \quad (1)$$

and for TAB,

$$\begin{cases} L_1^S = w'_{11}L_1^T + w'_{12}L_2^T,\ w'_{11} + w'_{12} = 1 \\ L_2^S = w'_{21}L_1^T + w'_{22}L_2^T,\ w'_{21} + w'_{22} = 1 \end{cases}. \quad (2)$$

Obviously, in Eq. 1, the knowledge of the teacher L_1^T or L_2^T is the linear combination of L_1^S and L_2^S. In other words, L_1^S and L_2^S are the bases of L_1^T and L_2^T of the teacher. In Eq. 2, it can been seen that the teacher's L_1^T and L_2^T are as the bases for L_1^S and L_2^S. Eq. 2 can be converted to:

$$L_1^T = A \cdot L_1^S + B \cdot L_2^S\ ,\ L_2^T = C \cdot L_1^S + D \cdot L_2^S, \quad (3)$$

where $A = \frac{w'_{22}}{w'_{11}w'_{22} - w'_{12}w'_{21}}$, $B = \frac{-w'_{12}}{w'_{11}w'_{22} - w'_{12}w'_{21}}$, $C = \frac{-w'_{21}}{w'_{11}w'_{22} - w'_{12}w'_{21}}$ and $D = \frac{w'_{11}}{w'_{11}w'_{22} - w'_{12}w'_{21}}$, which looks like Eq. 1. It seems that in this case, the student's L_1^S and L_2^S could also serve as the bases for L_1^T and L_2^T. However, (i) $A + B \neq 1$ and $C + D \neq 1$, but it is usually required in the previous KD that all the coefficients in each equation should be summed to be 1 [5,6,23]; (ii) when learning with TAB KD in Eq. 2, the coefficients A, B , C and D are uncontrollable, which may be arbitrary large during learning.

For the reasons above, although it seems that SAB and TAB have the same matrix representation, TAB is unable to get a similar result as SAB and more likely to get to a local minimum during training. Therefore, TAB KD is not equivalent to our SAB KD. When well trained, the student with this SAB KD fully learns the knowledge of the teacher because the teacher's L_i^T can be obtained by the linear combination of the student's L_i^S, $i = 1, 2, ..., N$. Our experiment also shows this property (see Section 4.2).

3.2. Preliminaries

Given a batch of videos V and captions T, Clip4clip learns a similarity function $s(v_i, t_j)$ to calculate the similar-

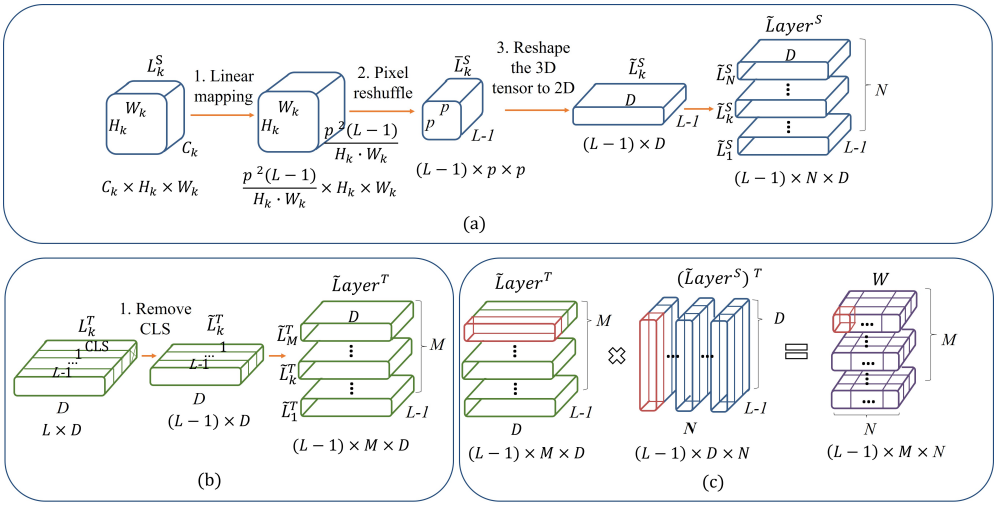

Figure 3. Reshaping the features of (a) the student L_k^S and (b) the teacher L_k^T to the same shape. (c) Similarity Computation.

ity between a video $v_i \in V$ and a caption $t_j \in T$. It obtains the frames' representation and the caption representation in a multi-modal embedding space via CLIP. The frames' representation is denoted as $Z_i = \{z_{i1}, z_{i2}, ..., z_{in}\}$, where n is the number of frames in v_i and z_{ik} $(k = 1, 2, ..., n)$ is the class token from the output of the CLIP's vision encoder corresponding to the k^{th} input frame. The caption representation is denoted as $W_j = \{w_{j1}, w_{j2}, ..., w_{jm}\}$, where m is the number of words in t_j (including [CLS] and [SEP]) and w_{jm} is used as the representation of t_j in Clip4clip. Clip4clip uses a temporal Transformer encoder to obtain the sequential feature, denoted as $Z_i' = \{z_{i1}', z_{i2}', ..., z_{in}'\} = TemporalTransformer(Z_i)$. Then, the mean pooling is used to aggregate the features of all frames to obtain the video representation, $z_i = \frac{1}{n}\sum_{k=1}^{n} z_{ik}'$. Finally, the similarity functions $s(v_i, t_j)$ and $s(t_j, v_i)$ are defined as:

$$s(v_i, t_j) = w_{jm}^{\top} z_i, \quad s(t_i, v_j) = z_j^{\top} w_{im}. \tag{4}$$

Nowadays, models such as Clip4clip based on the pre-trained model CLIP have been applied to many tasks. They pursue better performance with CLIP but have a heavy vision encoder, which makes them difficult to be deployed on edge devices such as mobile phones. Since MobileViT-v2 [27] is a light-weight and mobile-friendly hybrid network, we employ it as the vision encoder of the student model and maintain the original text encoder and temporal Transformer of Clip4clip. Note that the temporal Transformer is an optional module for the CLIPPING model structure, and the text encoder ($CLIP_{bert}$) can be further compressed through the existing methods (details are shown in the Section 4).

3.3. CLIPPING

To distill multimodal knowledge from Clip4clip to the MobileViT-v2-based model, we use four kinds of knowledge transfer: 1) temporal KD, 2) spatial KD, 3) SAB KD, and 4) cross-modality KD.

3.3.1 Temporal and Spatial Knowledge Distillation

The temporal KD is motivated by the previous finding that aligning multi-frame dependency from the teacher to the student can enhance the performance of the student [24], which encodes the multi-frame dependency into a latent embedding by using a recurrent unit ConvLSTM. In our architecture, multi-frame dependency can be modeled naturally through the temporal Transformer (Temporal Transformer) in Fig. 2, so we define the temporal KD loss as:

$$L_{TKD} = \frac{1}{B}\sum_{i=1}^{B} D_{KL}(Z_i'^S, Z_i'^T), \tag{5}$$

where B is the batch size, the superscripts S and T denote the student and the teacher, respectively, and $D_{KL}()$ is the KL divergence loss function. In addition to imitating the behavior of CLIP for each frame, we use spatial KD to align the spatial embeddings of the student and the teacher by:

$$L_{SKD} = \frac{1}{B}\sum_{i=1}^{B} D_{KL}(Z_i^S, Z_i^T). \tag{6}$$

The Temporal Transformer is an optional module for the CLIPPING model structure in our method. In this case, $Z_i'^S$ is exactly equal to Z_i^S in TKD.

3.3.2 SAB Knowledge Distillation

Recently, it is discovered that distilling intermediate features is more effective. So in addition to the spatial and temporal KD, we also force the vision encoder of the student to mimic the intermediate layers of the teacher. In the traditional OL2OL KD, some layers of the teacher are selected

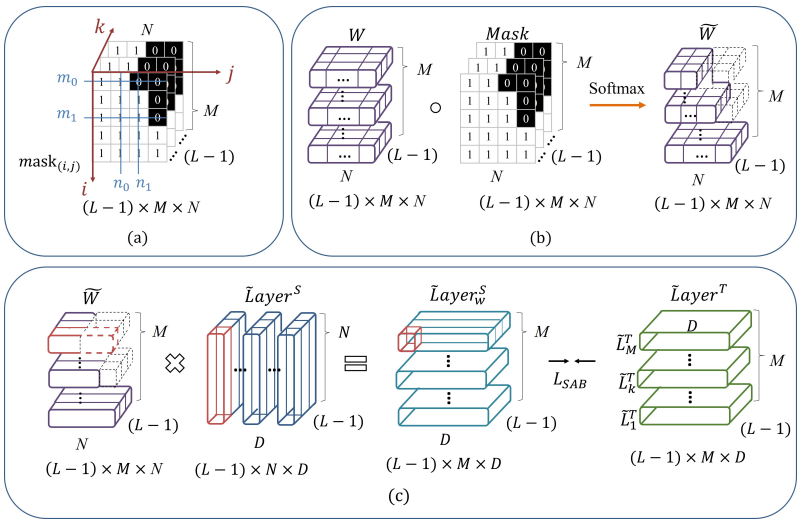

Figure 4. (a) Masks obtained with Eq. 7, where all the $L-1$ masks along the k dimension are the same. (b) Masking the similarity tensor W. (c) SAB alignment. L_{SAB} is the SAB loss.

and distilled into the student one by one [17]. However, it is difficult to find the optimal correspondence between the student and the teacher. The recent TAB works (e.g., [5]) still cannot help the student learn enough knowledge from the teacher when there is a large architecture gap between them, which is verified in our experiments. For our task, the student (MobileViT-v2) is a hybrid architecture, which has convolution layers in the shallow stages and separable self-attention layers in the later blocks. And a separable self-attention layer differs significantly from the traditional self-attention in two ways: 1) it does not learn an explicit attention map; 2) its input and output are still 3D, which are more similar to CNN features. Except for the separable self-attention in the later blocks, the shallow stages with convolution layers undoubtedly have a huge difference from the Transformer structure in CLIP, which has been investigated in previous works [31,41]. Therefore, we design SAB layer-wise alignment for KD of the intermediate layers from the Transformer to the CNN.

SAB Property. As shown in Fig. 2, the student's layers in our SAB KD can be explained as the bases of the feature space, and each layer of the teacher is a linear combination of the bases. After training, if the student's layers do ideally form the bases, the knowledge of the teacher's layers is learned completely by the student. Our experiment in Section 4.2 shows that the teacher's features in different layers can be well recovered from the features of the student's layers. Next, we describe how to implement SAB KD.

Feature Reshaping and Similarity Computation. Let the layers of the teacher and the student be $Layer^T = [L_1^T, L_2^T, \dots, L_M^T]$ and $Layer^S = [L_1^S, L_2^S, \dots, L_N^S]$ (usually $M > N$), respectively. $L_k^T \in R^{D \times L}$ is a 2D tensor, where D is the dimensionality of the token feature and L is the number of tokens, while $L_k^S \in R^{C_k \times H_k \times W_k}$ is a 3D tensor, where C_k, H_k and W_k are the channel number, height and width of the k^{th} CNN layer's feature. We calculate a similarity tensor W between L_k^S and L_k^T, which need to be reshaped first as shown in Figs. 3(a) and (b), respectively. For L_k^S, we first apply a learnable linear operator and then do pixel reshuffle on the result, obtaining $\bar{L}_k^S$, $k = 1, 2, ..., N$. Next, $\bar{L}_k^S$ is reshaped to $\tilde{L}_k^S$. As for L_k^T, we remove the class token and obtain a $(L-1) \times D$ tensor, denoted as $\tilde{L}_k^T$, $k = 1, 2, ..., M$. Finally, all $\tilde{L}_k^S$, $k = 1, 2, ..., N$, and $\tilde{L}_k^T$, $k = 1, 2, ..., M$, are represented as $\tilde{Layer}^S = [\tilde{L}_1^S, \tilde{L}_2^S, ..., \tilde{L}_N^S]$ and $\tilde{Layer}^T = [\tilde{L}_1^T, \tilde{L}_2^T, ..., \tilde{L}_M^T]$, respectively. After this reshaping, as shown in Fig. 3(c), we are able to measure the similarities between the intermediate layers of the student and the teacher through $W = \tilde{Layer}^T \times (\tilde{Layer}^S)^\top$, $W \in R^{(L-1) \times M \times N}$.

Masking and SAB Alignment. To speed up the training procedure, we design SAB KD with sequential masks, which follow this formula:

$$mask_{(i,j)} = \begin{cases} 0, \text{ if } j > n_0 \text{ and } i \leq m_0 \\ 0, \text{ if } j > n_1 \text{ and } m_0 < i \leq m_1 \\ 1, \text{ else} \end{cases}, \quad (7)$$

where $i = 1, 2, ..., M$, $j = 1, 2, ..., N$, $1 \leq m_0 < m_1 < M$, $1 \leq n_0 < n_1 < N$. And the final attention mask $Mask \in R^{(L-1) \times M \times N}$ is composed of $L-1$ same masks, Fig. 4(a) shows one example with $m_0 = 1, n_0 = 2, m_1 = 3, n_1 = 3$. $mask_{(i,j)} = 0$ means to mask all the elements along the k dimension (Fig. 4(a)) at (i, j) in W, while $mask_{(i,j)} = 1$ means to maintain their similarities, as shown in Fig. 4(b). This masking is based on our experimental finding: The student's lower-level layers

should learn from the teacher's lower-layers, while the student's higher layers should learn from all the teacher's layers. In Fig. 4(b), we also normalize the masks by performing softmax along the j dimension on each row, obtaining $\tilde{W} = \sigma(W \circ Mask)$, where σ is the softmax function and $\circ$ denotes the element-wise product. Guided by $\tilde{W}$, as shown in Fig. 4(c), the weighted student layers are calculated as: $\tilde{Layer}^S_w = \tilde{W} \times \tilde{Layer}^S$. Each component of $\tilde{Layer}^S_w$ is the linear combination of N student layers' features. Finally, the SAB KD loss is defined as:

$$L_{SAB} = D_{KL}(\tilde{Layer}^S_w, \tilde{Layer}^T). \tag{8}$$

3.3.3 Cross-Modality Knowledge distillation

To further adapt the student's features to the teacher's multimodal feature space, we employ the teacher's knowledge of cross-modality distributions to guide the training of the student. Specifically, we characterize the cross-modal distributions from two perspectives, global video-caption distributions of the teacher and local video-caption distributions of CLIP.

Global Video-Caption Distribution Alignment. We consider both the video-to-caption distribution A_{GVC} and the caption-to-video distribution A_{GCV}, the elements ($s(v_i, t_j)$ and $s(t_j, v_i)$) of which are defined via the similarities obtained by Eq. 4:

$$\begin{aligned} A_{GVC} &= \sigma \begin{pmatrix} s(v_1,t_1) & ... & s(v_1,t_B) \\ ... & ... & ... \\ s(v_B,t_1) & ... & s(v_B,t_j) \end{pmatrix}, \\ A_{GCV} &= \sigma \begin{pmatrix} s(t_1,v_1) & ... & s(t_1,v_B) \\ ... & ... & ... \\ s(t_B,v_1) & ... & s(t_B,v_B) \end{pmatrix}. \end{aligned} \tag{9}$$

, where σ is the softmax function along the first dimension of the similarity matrixs. Finally, the global video-caption distribution alignment loss is defined as:

$$L_G = D_{KL}(A^S_{GVC}, A^T_{GVC}) + D_{KL}(A^S_{GCV}, A^T_{GCV}). \tag{10}$$

Local Video-Caption Distribution Alignment. CLIP uses text prompts (such as "A picture of a ()") for zero-shot image classification. CLIP fills them with different words (e.g., "cat" and "dog") and results in different captions (e.g., "A picture of a cat" and "A picture of a dog"). It can match the captions to the corresponding images, showing some image-word alignment ability. We transfer this pre-training knowledge to the student through a local frame-word alignment as follows. Recall that $Z'_i = \{z'_{i1}, z'_{i2}, ..., z'_{in}\}$ and $W_j = \{w_{j1}, w_{j2}, ..., w_{jm}\}$ are the features of the video v_i and the caption t_j, respectively, with n being the number of frames in v_i and m the number of words in t_j (Section 3.1). The similarity between the k^{th} frame and the r^{th} word is defined as $s_{fw}(w_{jr}, z'_{ik}) = (w_{jr})^\top z'_{ik}$. Then we respectively define the local video-to-caption and caption-to-video similarities as:

$$\begin{aligned} s'(v_i, t_j) &= \frac{1}{n}\sum_{k=1}^{n} \max_{1\le r\le m} \{s_{fw}(w_{jr}, z'_{ik})\}, \\ s'(t_j, v_i) &= \frac{1}{m}\sum_{r=1}^{m} \max_{1\le k\le n} \{s_{fw}(z'_{ik}, w_{jr})\}. \end{aligned} \tag{11}$$

Finally, we have the local video-to-caption distribution A_{LVC} and the local caption-to-video distribution A_{LCV}:

$$\begin{aligned} A_{LVC} &= \sigma \begin{pmatrix} s'(v_1,t_1) & ... & s'(v_1,t_B) \\ ... & ... & ... \\ s'(v_B,t_1) & ... & s'(v_B,t_B) \end{pmatrix}, \\ A_{LCV} &= \sigma \begin{pmatrix} s'(t_1,v_1) & ... & s'(t_1,v_B) \\ ... & ... & ... \\ s'(t_B,v_1) & ... & s'(t_B,v_B) \end{pmatrix}, \end{aligned} \tag{12}$$

and the local caption-video distribution alignment loss is defined as:

$$L_L = D_{KL}(A^S_{LCV}, A^T_{LCV}) + D_{KL}(A^S_{LVC}, A^T_{LVC}). \tag{13}$$

Combining L_G and L_V, the cross-modality KD loss is:

$$L_{CM} = L_G + L_L. \tag{14}$$

Our experiment in Section 4.2 shows that only when joint trained with SAB, our local video-caption distribution alignment shows its advantage, which also verifies the effectiveness of SAB. Finally, the total loss for training our model is:

$$\begin{aligned} L = L_{task} + \alpha \cdot L_{TKD} + \beta \cdot L_{SKD} \\ + \gamma \cdot L_{CM} + \delta \cdot L_{SAB}, \end{aligned} \tag{15}$$

where L_{task} is the task-specifical loss, and α, β, γ and δ are loss balance weights.

4. Experiments

We conduct comprehensive experiments on three benchmarks for video-text retrieval (video-to-text ($v2t$) and text-to-video ($t2v$)): MSR-VTT [38], MSVD [4] and LSMDC [32]. The metrics Recall at rank K ($R@K$) are used for evaluation.

4.1. Comparison with State-of-the-Arts

MSR-VTT. In Table 1, we compare the proposed model CLIPPING with eight state-of-the-art methods on MSR-VTT: TACo [39], VideoClip [37], Frozen [3], VIOLET [12], OA-Trans [35], BridgeFormer [13], ALL-in-one [34] and MDMMT [11]. It can be seen that CLIPPING significantly surpasses those large-scale video-text/image-text pre-training models for video-text retrieval. For example, our model exceeds VideoClip and Frozen by absolute 9.8% $t2vR@1$ and 8.2% $t2vR@1$, respectively. Note that the model MDMMT uses CLIP as its backbone, while our CLIPPING uses CLIP as the teacher. For a more efficient and edge deployable video-language retrieval model, we further compress CLIPPING into two models: $\text{CLIPPING}_{\text{w/o T}}$ and CLIPPING$*$. $\text{CLIPPING}_{\text{w/o T}}$

Model	PT Datasets	Params	$R@1$
TACo	HT100M	212M	28.4
VideoClip	HT100M	130M	30.9
Frozen	C3M,W2M,COCO	232M	32.5
ALL-in-one-S	W2M,HT100M	33M	33.5
VIOLET	C3M,W2M	198M	34.5
OA-Trans	C3M,W2M	232M	35.8
BridgeFormer	C3M,W2M	160M	37.6
ALL-in-one-B	W2M,HT100M	110M	37.9
MDMMT	C400M,AudioSet	226M	38.9
CLIPPING$*_{\text{w/o T}}$	-	**8.7M**	37.5
CLIPPING$_{\text{w/o T}}$	-	21.3M	38.6
CLIPPING$*$	-	46.5M	39.8
CLIPPING	-	55.0M	40.6
CLIPPING	IN21K	55.0M	**40.7**

Table 1. Comparison with state-of-the-art models on MSR-VTT ($1k$ split) for text-to-video retrieval. "PT Datasets": datasets used for pre-training the vision encoder. "HT100M": HowTo100M dataset [28]. "C400M": CLIP-400M dataset [30]. "IN21K": ImageNet21K dataset [9]. "W2M": WebVid-2M dataset [3]. "C3M": CC3M dataset [33]. All the methods in Table are the comparable methods with parameters less than 240M and flops less than 360G.

does not use the temporal Transformer and CLIPPING$*$ compresses the text encoder through TinyBERT [17]. It can be seen from Table 1 that the CLIPPING$*$ based models also outperform the small model ALL-in-one-S even though they are smaller. In Fig. 5, we show the performances and Flops of these models. Among previous models, MDMMT has the best performance and ALL-in-one-S is the fastest. CLIPPING not only obtains relative 4.6% performance gain over MDMMT but also is faster than All-in-one-S. Table 2 provides the details of Flops and parameters for each module of our method.

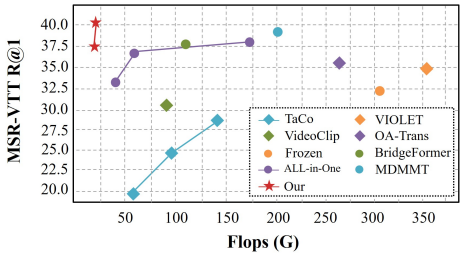

Figure 5. Flops and Performances.

Other Benchmarks. We also compare our CLIPPING with the state-of-the-art Frozen [3], MDMMT [11], NoiseEst [2] and SupportSet [29] on the LSMDC and MSVD datasets. In Table 3, CLIPPING again obtains the best performance with fewest parameters and Flops, which achieves 88.1%–92.9% of the performance of its teacher on the LSMDC and MSVD datasets, showing consistent improvements and generalization across different datasets.

4.2. Ablation Study

Key Components. We provide detailed ablation study to validate each key component of our proposed method,

Modules	Parameters	Flops
Vision encoder (MobileViT-v2)	4.5M	16.8G
Temporal Transformer	37.8M	1.2G
Language encoder ($CLIP_{bert}$)	12.6M	0.15G
Language encoder (TinyBERT)	4.2M	0.05G

Table 2. The Flops and parameters of each module in our method. We adopt 12 frames for each video.

Model	PT Datasets	Vision Encoders	$R@1$
LSMDC dataset			
NoiseEst	HT100M	Resnet152,ResNext-101	6.4
SupportSet	HT100M	ResNet-152,R(2+1)D-34	IN21K
Frozen	C3M,W2M	Space-Time	15.0
MDMMT	C400M	CLIP$_{vision}$	18.8
CLIPPING	-	MobileViT-v2	**19.9**
MSVD dataset			
NoiseEst	HT100M	Resnet152,ResNext-101	20.3
SupportSet	HT100M	ResNet-152,R(2+1)D-34	28.4
Frozen	C3M,W2M	Space-Time	33.7
CLIPPING	-	MobileViT-v2	**42.9**

Table 3. Comparison with the state-of-the-art on the LSMDC and MSVD datasets.

Vision Encoder	KD Types	$t2vR@1$	$v2tR@1$
CLIP$_{vision}$	-	44.5	42.2
MobileViTv2	-	25.7	24.5
MobileViTv2	T	28.8	27.3
MobileViTv2	T,S	33.0	32.8
MobileViTv2	T,S,SAB	37.6	36.2
MobileViTv2	T,S,SAB,CM_G	39.6	39.1
MobileViTv2	T,S,SAB,CM_G,CM_L	**40.7**	**40.2**

Table 4. Ablation study of different KD components of CLIPPING on the $1k$ validation set of MSR-VTT. The first row is the results of the teacher model (Clip4clip). T, S, SAB, CM_G and CM_L denote temporal KD, spatial KD, SAB KD, global cross-modality KD and local cross-modality KD, respectively.

KD Types	$t2vR@1$	$v2tR@1$
T, S	33.0	32.8
T, S, $OL2OL$	34.6	33.4
T, S, TAB	35.1	34.4
T, S, $SAB_{w/o\ masking}$	37.1	35.7
T, S, $SAB_{w/\ masking}$	**37.6**	**36.2**

Table 5. Ablation study of different KD types on MSR-VTT ($1k$ split). All the models are trained for 36 epochs with the same setting (see the supplementary materials for details).

on MSR-VTT. From Table 4, we can see that without any KD, it gets extreme low accuracies. When simply adding the temporal KD and spatial KD, $t2vR@1$ and $v2tR@1$ increase significantly. When we further add SAB KD, and then global and local cross-modality KD, CLIPPING's performance rises gradually. This study shows that all these key components of CLIPPING are effective. Compared with the teacher, the full CLIPPING achieves about 91.5% and 95.3% of the performance of its teacher on $t2vR@1$ and $v2tR@1$, respectively, with the vision encoder of 19.5x

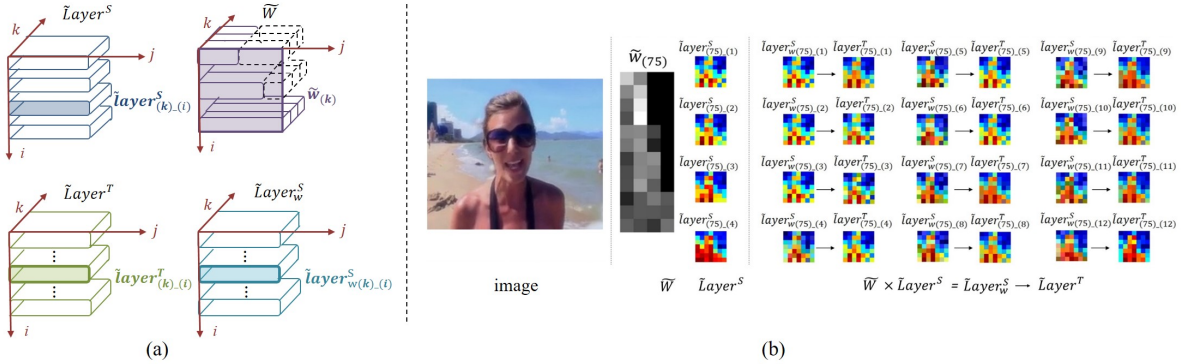

Figure 6. (a) Feature tensors of the student and the teacher. (b) An example to demonstrate that the teacher's features are the linear combinations of the student features. More examples are provided in the supplementary materials.

Figure 7. Visualization of the frame-word alignments. (a) CLIP; (b) CLIPPING$^{-}{}_{\text{TAB}}$; (c) CLIPPING.

smaller.

KD Types. In Table 5, we compare the proposed SAB KD with the previous OL2OL KD and TAB KD. For SAB and TAB, the same layers of the student and the teacher are used for KD, the details of which are give in the supplementary materials. From Table 5, we can see that TAB performers better than OL2OL, and our SAB outperforms TAB. In Section 3.3.2, we use masking to speed up the training. In this study, we also compare "with masking" and "without masking". The last two rows of Table 5 verify that the masking is beneficial.

SAB Property. The student's layers in our SAB KD can be explained as the bases of the feature space, and each layer of the teacher is a linear combination of the bases (Section 3.2). In Fig. 6, we show that this property holds. In SAB KD, the student has 4 layers and the teacher has 12 layers. We randomly pick 3 image examples, and for each example, we select the features of one random token (in the k dimension in Fig. 6(a)). After training, the linear combinations $\tilde{W} \times \tilde{Layer}^S = \tilde{Layer}^S_w$ show feature patterns very similar to the teacher's features ($\tilde{Layer}^T$). This property verifies that the teacher's knowledge is fully absorbed by the student.

In Fig. 7, we visualize the frame-word alignments with and without SAB KD. Fig. 7(a) is the result of the CLIP in CLIP4clip. Fig. 7(b) is the result of CLIPPING$^{-}{}_{\text{TAB}}$, which is our model without SAB KD but with the previous TAB KD. It can be seen that the frame-word alignment with SAB KD (Fig. 7(c)) shows clear and correct word-level attentions (e.g., "naked", "child", "runs" and "through") like CLIP. But with TAB KD, the most important words (e.g., "naked" and "child") cannot be noticed; instead, some insignificant words ([CLS] and "a") are activated. This study shows that the knowledge of pre-trained CLIP can be more effectively transferred to the student at the fine-tuning stage with our SAB.

5. Conclusion

In this paper, we propose a novel and efficient knowledge distillation method that is specially designed for small vision-language models. It includes temporal KD, spatial KD, SAB KD and cross-modality KD. Especially, the SAB KD has the property of the student's layers being the bases of the feature space. After training, the teacher's features are the linear combinations of the bases, indicating that the student has fully absorbed the knowledge of the teacher. CLIPPING significantly outperforms the state-of-the-art and is comparable or even superior to many large pre-training models. In the feature, we will apply CLIPPING to other vision-language models for compression.

Acknowledgements

We gratefully acknowledge the support of MindSpore [1], CANN (Compute Architecture for Neural Networks) and Ascend AI Processor used for this research.

References

[1] https://www.mindspore.cn/. 8

[2] Elad Amrani, Rami Ben-Ari, Daniel Rotman, and Alex Bronstein. Noise estimation using density estimation for self-supervised multimodal learning. In *AAAI*, 2021. 7

[3] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval. In *ICCV*, 2021. 2, 6, 7

[4] David Chen and William Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, 2011. 6

[5] Defang Chen, JianPing Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. Distillation with semantic calibration. In *AAAI*, 2021. 2, 3, 5

[6] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling Knowledge via Knowledge Review. In *CVPR*, 2021. 2, 3

[7] Xianing Chen, Qiong Cao, Yujie Zhong, Jing Zhang, Shenghua Gao, and Dacheng Tao. DearKD: Data-efficient early knowledge distillation for vision transformers. In *ICLR*, 2022. 2

[8] Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobile-Former: Bridging MobileNet and Transformer. In *CVPR*, 2022. 1

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 7

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[11] Maksim Dzabraev, Maksim Kalashnikov, Stepan Komkov, and Aleksandr Petiushko. MDMMT: Multidomain Multimodal Transformer for Video Retrieval. In *CVPR*, 2021. 2, 6, 7

[12] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. VIOLET: End-to-End Video-Language Transformers with Masked Visual-token Modeling. In *arXiv:2111.1268*, 2021. 1, 6

[13] Yuying Ge, Yixiao Ge, Xihui Liu, Dian Li, Ying Shan, Xiaohu Qie, and Ping Luo. Bridgeformer: Bridging video-text retrieval with multiple choice questions. In *CVPR*, 2022. 6

[14] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. CMT: Convolutional Neural Networks Meet Vision Transformers. In *CVPR*, 2022. 1

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. In *arXiv preprint arXiv:1503.02531*, 2015. 1, 2

[16] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 1

[17] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. TinyBERT: Distilling BERT for Natural Language Understanding. In *EMNLP*, 2020. 1, 2, 5, 7

[18] Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *EMNLP*, 2016. 1

[19] Pavan Kumar, Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. An Improved One millisecond Mobile Backbone. In *arXiv preprint arXiv:2206.04040*, 2022. 1, 2

[20] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *arXiv preprint arXiv:2201.12086*, 2022. 1

[21] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NIPS*, 2021. 1

[22] Yanyu Li, Geng Yuan, Yang Wen, Eric Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. EfficientFormer: Vision Transformers at MobileNet Speed. In *arXiv:2206.01191*, 2022. 1

[23] Sihao Lin, Hongwei Xie, Bing Wang, Kaicheng Yu, Xiaojun Chang, Xiaodan Liang, and Gang Wang. Knowledge Distillation via the Target-aware Transformer. In *CVPR*, 2022. 2, 3

[24] Yifan Liu, Chunhua Shen, Changqian Yu, and Jingdong Wang. Efficient Semantic Video Segmentation with Per-Frame Inference. In *ECCV*, 2020. 4

[25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *ICCV*, 2021. 2

[26] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval. *arXiv preprint arXiv:2104.08860*, 2021. 1, 2

[27] Sachin Mehta and Mohammad Rastegari. Separable Self-attention for Mobile Vision Transformers. In *ICLR*, 2022. 1, 2, 4

[28] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*, 2019. 7

[29] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander Hauptmann, João Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning. In *ICLR*, 2021. 7

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, and Jack Clark. Learning transferable visual models from natural language supervision. *ICML*, 2021. 1, 2, 7

[31] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? In *NIPS*, 2021. 2, 3, 5

[32] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for Movie Description. In *CVPR*, 2015. 6

[33] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 7

[34] Alex Jinpeng Wang, Yixiao Ge, Rui Yan, Yuying Ge, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. All in one: Exploring unified video-language pre-training. In *arXiv preprint arXiv:2203.07303*, 2022. 2, 6

[35] Jinpeng Wang, Yixiao Ge, Guanyu Cai, Rui Yan, Xudong Lin, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Object-aware video-language pre-training for retrieval. In *CVPR*, 2022. 6

[36] Zekun Wang, Wenhui Wang, Haichao Zhu, Ming Liu, Bing Qin, and Furu Wei. Distilled dual-encoder model for vision-language understanding. In *arXiv preprint arXiv:2112.08723*, 2021. 2

[37] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *arXiv preprint arXiv:2104.08860*, 2021. 2, 6

[38] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. 2016. 6

[39] Jianwei Yang, Yonatan Bisk, and Jianfeng Gao. Taco: Token-aware cascade contrastive learning for video-text alignment. In *ICCV*, 2021. 6

[40] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *CVPR*, 2022. 1

[41] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis E.H. Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-Token ViT: Training Vision Transformers From Scratch on ImageNet. In *ICCV*, 2021. 5