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

Supplementary Material

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

1. Implementation Details

1.1. CLIPPING Initialization

We initialize the vison encoder MobileViT-v2 by pre-training it on the imageNet21k dataset. The initial parameters of the text encoder are a copy of the CLIP's text encoder, and the temporal Transformer is initialized in the same way as in CLIP4clip. The rest of the parameters, e.g., the linear projections in SAB KD, are initialized randomly from the Gaussian distribution $\mathcal{N}(0,1)$.

1.2. CLIPPING Training Settings

The text encoder is fine-tuned with a small learning rate ($1e-7$) from the beginning. The learning rates of the other parts are initialized by $1e-5$ and decay according to the cosine schedule. The whole KD is optimized by Adam with a batch size of 64 for 36 epochs. From the 1^{st} epoch to the 5^{th} epoch, the weights for SAB KD and CM KD are set to 0 ($\delta = 0$ and $\gamma = 0$), and then they are added during the 6^{th} epoch to the 36^{th} epoch. During the last 30 epochs, the learning rates are re-initialized by $1e-5$ and re-decay for all the parts except for the text encoder. When SAB KD and CM KD are added, we set $\delta = 1$ and $\gamma = 0.125$. For the whole training period, we set the balance weights α and β to 1. In our SAB KD, we choose 4 MobileViT-v2 (student) layers ($layer_2$, $layer_3$, $layer_4$ and $layer_5$) and 12 CLIP (teacher) layers (all the 12 Transformer layers in ViT-B-32). Note that the previous TAB KD [1] in Table 4 is also trained with the same selected layers. In our experiments, the masks are calculated through Eq. 7 with $n_0 = 2$, $m_0 = 4$, $n_1 = 3$ and $m_1 = 9$. A softmax function is added to normalize the similarity matrix along the first dimension both in the training and inference.

1.3. Details about KL Divergence.

In our experiments, all the input and target for KL Divergence loss (D_{KL}) are float tensors, and the loss expects an input in the log-space and a normalized target to avoid underflow issues. Take the Eq. 5 for example, both $Z_i'^S$ and $Z_i'^T$ are 768×12 tensors, and we calculate the loss as $D_{KL}(log_{softmax}(Z_i'^S), softmax(Z_i'^T))$.

1.4. CLIPPING Algorithm

```
Algorithm CLIPPING
Input: The training dataset $D = \{(v_j, t_j)\}_{j=1}^{N}$; the pre-trained teacher model with parameters $\theta^t$; the student model with parameters $\theta^s$; $i$ denotes the training epoch. $i = 1$ at the beginning.
Output: A trained student model.
while $i > 36$ do
    Sample a mini-batch $B$ from $D$.
    Forward $B$ into $\theta^t$ and $\theta^s$ to obtain the features $Layer^S$, $Z_i^S$, $Z_i'^S$, $Layer^T$, $Z_i^T$ and $Z_i'^T$.
    Reshape $Layer^S$ and $Layer^T$ according to Fig. 3 to obtain $\tilde{Layer}^S$ and $\tilde{Layer}^T$.
    Calculate the similarity matrix $\tilde{W} = \sigma(\tilde{Layer}^T \times (\tilde{Layer}^S)^\top \circ Mask)$, where $Mask$ is calculated by Eq. 7.
    Align the video-caption distributions as Eq. 9 and Eq. 12.
    Update the parameters $\theta^s$ by backward propagating the gradients of the loss in Eq. 15.
    $i = i + 1$.
end while
```

2. More Results

2.1. Different Students

Besides MobileViT-v2, we also use other models as the student in CLIPPING, which are EfficientNet-b0 [5] and EfficientFormer-L1 [4]. Table 6 shows that CLIPPING with each of these small models as the student outperformers all the prepared models in Table 1 of the paper, indicating that it is a general KD method.

Vison Encoder	$t2vR@1$	$v2tR@1$
EfficientNet-b0	39.3(88.3%)	38.9(92.2%)
EfficientFormer-L1	40.5(91.0%)	39.8(94.3%)
MobileVITv2	40.7(91.5%)	40.2(95.3%)

Table 6. CLIPPING with different students.

2.2. Masking

In Fig. 8, we show the effect of the masking. It can be seen that CLIPPING with the masking reaches the highest accuracy at the 36th epoch, while CLIPPING without the masking needs to be trained with more epochs for better accuracy. It verifies that the masking is able to speed up the training procedure.

2.3. CLIPPING for Other Tasks

We also apply our CLIIPING method for other tasks, including the image classification and video recognition. For the image classification task, we compare our SAB KD with TAB KD for EfficientFormer on the ImageNet dataset. From Table 7, we can see that our SAB outperforms TAB. For video recognition, we first train the CLIP-based model and frozen it as teacher. Then we distil a small MobileViT-v2-based model by our CLIPPING (for cross-modal KD, we use the text prompts, such as "this is label, a video of action" and "human action of label", for video classification). Table 8 shows the video recognition results on Kinetic 400. Compared with the state-of-the-art methods, our method shows the highest accuracies with lowest flops.

Methods	EfficientFormer [4]	TAB	Our
Accuracy	79.2	80.0 ± 0.2	81.1 ± 0.2

Table 7. Results on ImageNet.

Methods	Flops	Acc
X3D-M [2]	19.4	76.0
MoViNet [3]	56.9	78.2
X3D-XL [2]	150	79.1
Our	16.8	80.0

Table 8. Comparison with the SOTA on the Kinetic400 dataset.

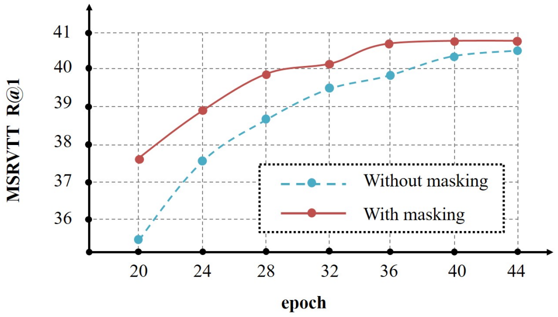

Figure 8. Convergence comparison with and without the masking.

3. More Visualizations

3.1. SAB Property

We give two more examples to show the SAB property (Fig. 9). Both of them show that the student's feature patterns of linear combinations are very similar to the teacher's features, which verifies that the teacher's knowledge is fully absorbed by the student.

3.2. TAB KD vs. SAB KD

To verify the advantage of our SAB KD, we show the linear combinations' results of SAB KD and TAB KD in Fig. 10. For SAB KD, we directly use the results in Fig. 9. For TAB KD, from the KD equation in Fig. 1(b) (similar to Eq. 2), we obtain $L_i^T, i = 1, 2, ..., M$, represented by $L_j^S, i = 1, 2, ..., N$ (similar to Eq. 3). Then we visualize the features $L_i^T, i = 1, 2, ..., M$ and $L_j^S, i = 1, 2, ..., N$ as for SAB KD in Fig. 10(b). It can be seen that TAB KD is not equivalent to our SAB KD and does not have the property of the student as the base of the teacher.

References

[1] Defang Chen, JianPing Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. Distillation with semantic calibration. In *AAAI*, 2021. 1, 3

[2] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, 2020. 2

[3] Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong. Movinets:

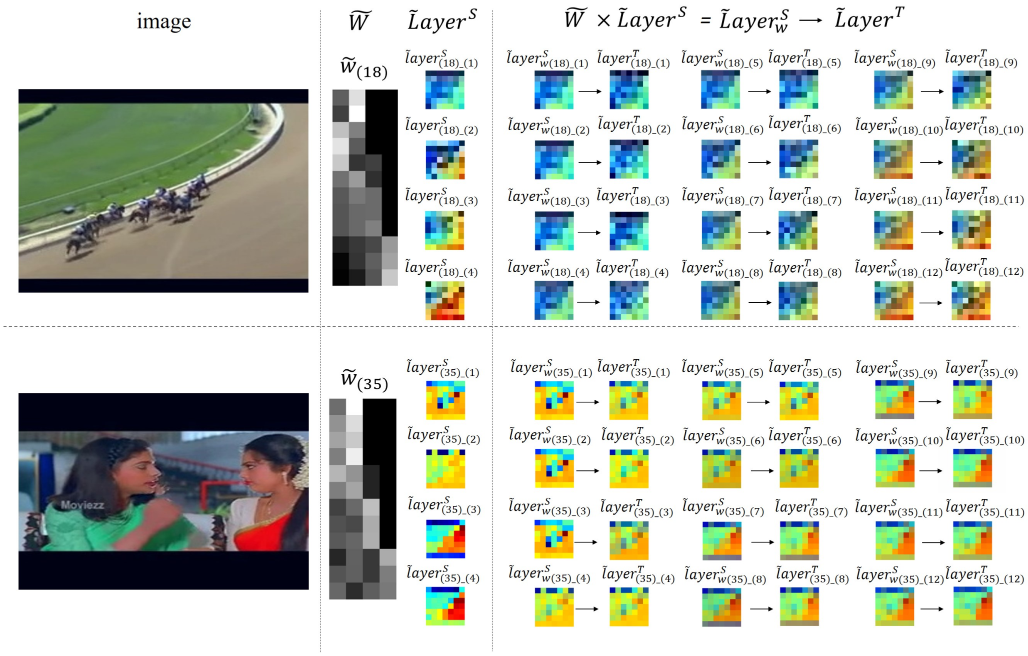
image
$\widetilde{W}$
$\tilde{L}ayer^S$
$\widetilde{W} \times \tilde{L}ayer^S = \tilde{L}ayer^S_w \longrightarrow \tilde{L}ayer^T$
$\widetilde{w}_{(18)}$
$\tilde{l}ayer^S_{(18)\_(1)}$ $\tilde{l}ayer^S_{w(18)\_(1)}$ $\tilde{l}ayer^T_{(18)\_(1)}$ $\tilde{l}ayer^S_{w(18)\_(5)}$ $\tilde{l}ayer^T_{(18)\_(5)}$ $\tilde{l}ayer^S_{w(18)\_(9)}$ $\tilde{l}ayer^T_{(18)\_(9)}$
$\tilde{l}ayer^S_{(18)\_(2)}$ $\tilde{l}ayer^S_{w(18)\_(2)}$ $\tilde{l}ayer^T_{(18)\_(2)}$ $\tilde{l}ayer^S_{w(18)\_(6)}$ $\tilde{l}ayer^T_{(18)\_(6)}$ $\tilde{l}ayer^S_{w(18)\_(10)}$ $\tilde{l}ayer^T_{(18)\_(10)}$
$\tilde{l}ayer^S_{(18)\_(3)}$ $\tilde{l}ayer^S_{w(18)\_(3)}$ $\tilde{l}ayer^T_{(18)\_(3)}$ $\tilde{l}ayer^S_{w(18)\_(7)}$ $\tilde{l}ayer^T_{(18)\_(7)}$ $\tilde{l}ayer^S_{w(18)\_(11)}$ $\tilde{l}ayer^T_{(18)\_(11)}$
$\tilde{l}ayer^S_{(18)\_(4)}$ $\tilde{l}ayer^S_{w(18)\_(4)}$ $\tilde{l}ayer^T_{(18)\_(4)}$ $\tilde{l}ayer^S_{w(18)\_(8)}$ $\tilde{l}ayer^T_{(18)\_(8)}$ $\tilde{l}ayer^S_{w(18)\_(12)}$ $\tilde{l}ayer^T_{(18)\_(12)}$
$\widetilde{w}_{(35)}$
Moviezz
$\tilde{l}ayer^S_{(35)\_(1)}$ $\tilde{l}ayer^S_{w(35)\_(1)}$ $\tilde{l}ayer^T_{(35)\_(1)}$ $\tilde{l}ayer^S_{w(35)\_(5)}$ $\tilde{l}ayer^T_{(35)\_(5)}$ $\tilde{l}ayer^S_{w(35)\_(9)}$ $\tilde{l}ayer^T_{(35)\_(9)}$
$\tilde{l}ayer^S_{(35)\_(2)}$ $\tilde{l}ayer^S_{w(35)\_(2)}$ $\tilde{l}ayer^T_{(35)\_(2)}$ $\tilde{l}ayer^S_{w(35)\_(6)}$ $\tilde{l}ayer^T_{(35)\_(6)}$ $\tilde{l}ayer^S_{w(35)\_(10)}$ $\tilde{l}ayer^T_{(35)\_(10)}$
$\tilde{l}ayer^S_{(35)\_(3)}$ $\tilde{l}ayer^S_{w(35)\_(3)}$ $\tilde{l}ayer^T_{(35)\_(3)}$ $\tilde{l}ayer^S_{w(35)\_(7)}$ $\tilde{l}ayer^T_{(35)\_(7)}$ $\tilde{l}ayer^S_{w(35)\_(11)}$ $\tilde{l}ayer^T_{(35)\_(11)}$
$\tilde{l}ayer^S_{(35)\_(4)}$ $\tilde{l}ayer^S_{w(35)\_(4)}$ $\tilde{l}ayer^T_{(35)\_(4)}$ $\tilde{l}ayer^S_{w(35)\_(8)}$ $\tilde{l}ayer^T_{(35)\_(8)}$ $\tilde{l}ayer^S_{w(35)\_(12)}$ $\tilde{l}ayer^T_{(35)\_(12)}$

Figure 9. More examples to demonstrate that the teacher's features are the linear combinations of the student features.

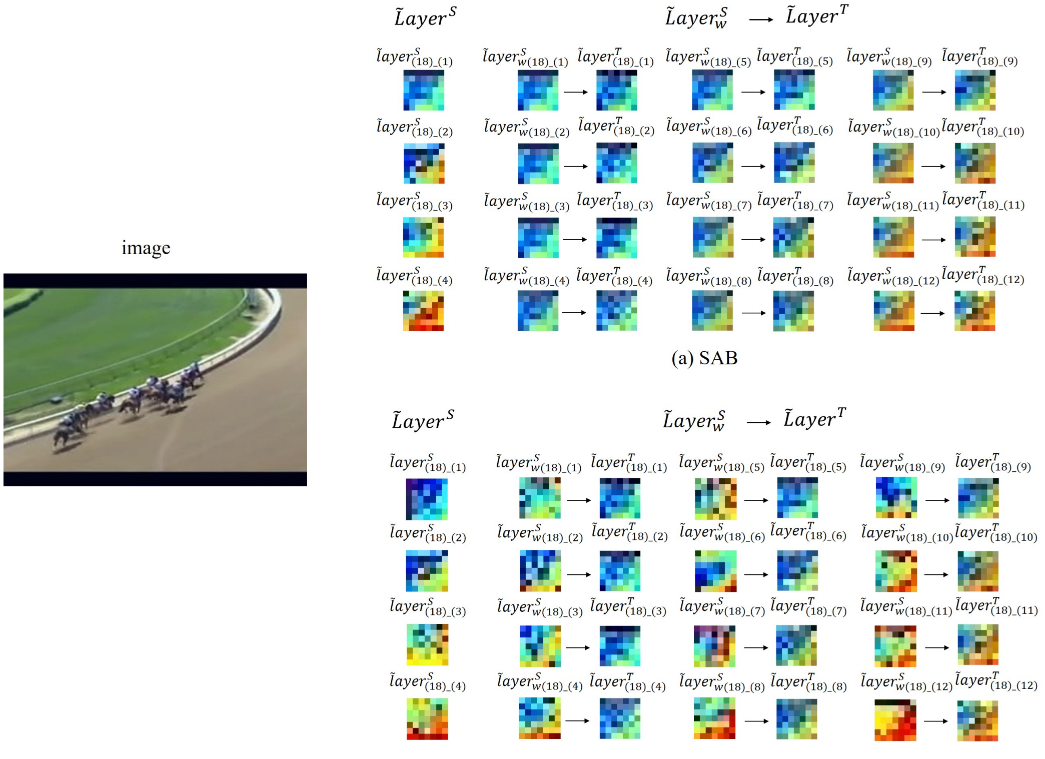
image
$\tilde{L}ayer^S$
$\tilde{L}ayer^S_w \longrightarrow \tilde{L}ayer^T$
$\tilde{l}ayer^S_{(18)\_(1)}$ $\tilde{l}ayer^S_{w(18)\_(1)}$ $\tilde{l}ayer^T_{(18)\_(1)}$ $\tilde{l}ayer^S_{w(18)\_(5)}$ $\tilde{l}ayer^T_{(18)\_(5)}$ $\tilde{l}ayer^S_{w(18)\_(9)}$ $\tilde{l}ayer^T_{(18)\_(9)}$
$\tilde{l}ayer^S_{(18)\_(2)}$ $\tilde{l}ayer^S_{w(18)\_(2)}$ $\tilde{l}ayer^T_{(18)\_(2)}$ $\tilde{l}ayer^S_{w(18)\_(6)}$ $\tilde{l}ayer^T_{(18)\_(6)}$ $\tilde{l}ayer^S_{w(18)\_(10)}$ $\tilde{l}ayer^T_{(18)\_(10)}$
$\tilde{l}ayer^S_{(18)\_(3)}$ $\tilde{l}ayer^S_{w(18)\_(3)}$ $\tilde{l}ayer^T_{(18)\_(3)}$ $\tilde{l}ayer^S_{w(18)\_(7)}$ $\tilde{l}ayer^T_{(18)\_(7)}$ $\tilde{l}ayer^S_{w(18)\_(11)}$ $\tilde{l}ayer^T_{(18)\_(11)}$
$\tilde{l}ayer^S_{(18)\_(4)}$ $\tilde{l}ayer^S_{w(18)\_(4)}$ $\tilde{l}ayer^T_{(18)\_(4)}$ $\tilde{l}ayer^S_{w(18)\_(8)}$ $\tilde{l}ayer^T_{(18)\_(8)}$ $\tilde{l}ayer^S_{w(18)\_(12)}$ $\tilde{l}ayer^T_{(18)\_(12)}$

(a) SAB

(b) TAB

Figure 10. Examples of the linear combinations of the student's features that are trained with SAB KD (ours) and TAB KD [1].

Mobile video networks for efficient video recognition. In *CVPR*, 2021. 2

[4] 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. 2

[5] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 2