Disentangling Spatial and Temporal Learning for Efficient Image-to-Video Transfer Learning

Zhiwu Qing[1] Shiwei Zhang[2*] Ziyuan Huang[3] Yingya Zhang[2]
Changxin Gao[1] Deli Zhao[2] Nong Sang[1*]
[1]Key Laboratory of Image Processing and Intelligent Control
School of Artificial Intelligence and Automation, Huazhong University of Science and Technology
[2]Alibaba Group [3]ARC, National University of Singapore
{qzw, cgao, nsang}@hust.edu.cn {zhangjin.zsw, yingya.zyy}@alibaba-inc.com
ziyuan.huang@u.nus.edu zhaodeli@gmail.com

Abstract

Recently, large-scale pre-trained language-image models like CLIP have shown extraordinary capabilities for understanding spatial contents, but naively transferring such models to video recognition still suffers from unsatisfactory temporal modeling capabilities. Existing methods insert tunable structures into or in parallel with the pre-trained model, which either requires back-propagation through the whole pre-trained model and is thus resource-demanding, or is limited by the temporal reasoning capability of the pre-trained structure. In this work, we present DiST, which disentangles the learning of spatial and temporal aspects of videos. Specifically, DiST uses a dual-encoder structure, where a pre-trained foundation model acts as the spatial encoder, and a lightweight network is introduced as the temporal encoder. An integration branch is inserted between the encoders to fuse spatio-temporal information. The disentangled spatial and temporal learning in DiST is highly efficient because it avoids the back-propagation of massive pre-trained parameters. Meanwhile, we empirically show that disentangled learning with an extra network for integration benefits both spatial and temporal understanding. Extensive experiments on five benchmarks show that DiST delivers better performance than existing state-of-the-art methods by convincing gaps. When pre-training on the large-scale Kinetics-710, we achieve 89.7% on Kinetics-400 with a frozen ViT-L model, which verifies the scalability of DiST. Codes and models can be found in https://github.com/alibaba-mmai-research/DiST.

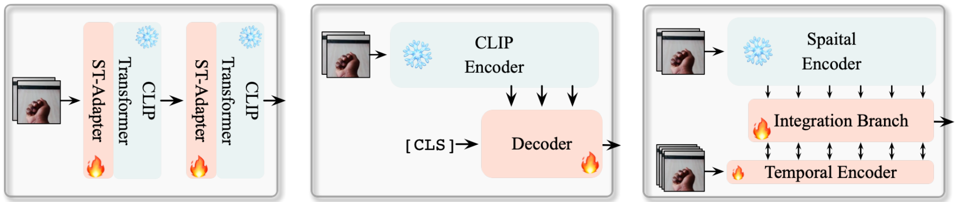
1. Introduction

Video understanding is a fundamental yet challenging research topic in computer vision. Early approaches for this task learn spatio-temporal representations by designing different architectures, such as two-stream models [28, 65], 3D networks [59, 16, 29], Transformers [1, 3, 72], and they have achieved impressive progress on some challenging benchmarks [29, 19]. Recently, a new paradigm that transfers the large-scale language-image pre-training models, *e.g.*, CLIP [52], to video understanding tasks [26, 39, 45, 31] has been drawing lots of attention due to its remarkable spatial modeling potential, and it deserves the enhancement of the potential for spatio-temporal reasoning.

Figure 1: Accuracy vs. per-video GFLOPs on SSV2 [19] and K400 [29] with ViT-B/16 [12]. "EVL" [39]: Efficient Video Learning. "ST-A" [45]: ST-Adapter. "CLIP": Fully fine-tuning the CLIP pre-trained image encoder.

As in Fig. 2 (a), a popular design for efficient transfer learning is to insert tunable structures between pre-trained Transformer blocks [45, 37, 18, 8]. Parameter-efficient as it is, it would require back-propagation through massive parameters that are supposed to be frozen, which is inefficient in training. With a large number of video frames, this inefficiency hinders the scaling of large video-text foundation models under limited GPU memory, as in Fig. 1. To tackle this, the recent work [39] introduces a decoder in parallel with the pre-trained encoder. This indeed increases train-

*Corresponding authors.



(a) ST-Adapter (b) EVL (c) **DiST (Ours)**

Method	BP-Free	Temp.	Spat.
(a) ST-Adapter	✗	Strong	Strong
(b) EVL	✓	Weak	Strong
(c) **DiST**	✓	Strong	Strong

Figure 2: Comparison with existing efficient fine-tuning approaches for video recognition. **(a)** ST-Adapter [45]. **(b)** EVL [39]. **(c)** Our proposed DiST. "BP-Free" indicates "back-propagation-free" for the encoder. "Temp." and "Spat." are "temporal modeling" and "spatial modeling", respectively.

ing efficiency by avoiding back-propagation through pre-trained parameters. However, the major function of the decoder in such an approach is to collect relevant information from the frozen encoder, which makes the output of the decoder highly correlated to the spatial information provided by the pre-trained image Transformer.

In this work, we present DiST, a dual-encoder framework for efficiently transferring the pre-trained image-text foundation models to video-text ones. DiST shares the merits of both frameworks mentioned above by disentangling the spatial and temporal modeling: **(i)** by connecting all the structures in parallel to the frozen model, DiST avoids the back-propagation through the massive parameters in the pre-trained Transformer; **(ii)** by introducing a separated encoder that specifically designed to extract temporal information for the input video, the temporal modeling capability is enhanced. Further, to simultaneously exploit the spatial semantics and the temporal information extracted by the dual-encoder structure, an integration branch is imposed to fuse the features from both spatial and temporal encoders.

We evaluate DiST on three challenging supervised video recognition benchmarks, *i.e.*, Kinetics-400 [29], Something-Something v2 [19], Epic-Kitchens-100 [10], and two zero-shot benchmarks, *i.e.*, UCF101 [55] and HMDB51 [23]. Our DiST achieves state-of-the-art performance on all datasets with convincing gains to existing approaches. Moreover, under limited resources, DiST enables us to pre-train on large-scale video datasets since only the lightweight temporal encoder and integration branch require pre-training. With Kinetics-710 for pre-training, we verify the superior scalability of DiST and achieve better performance than fully fine-tuned ones.

2. Related Works

Visual-language Pre-training. Recently, visual language pre-training [42, 41, 56, 79, 34, 52, 75, 24, 74] has made remarkable progress. One of the most representative works is CLIP [52]. Following that, a series of prompt-based [33, 78, 77, 2, 76] and adapter-based [48, 18, 37, 57] works explored how to efficiently transfer the pre-trained models to image tasks. Meanwhile, transferring language-image pre-trained models to videos [67, 45, 39, 44, 27, 9] has attracted wide attention due to its striking performance. For example, Ni *et al.* [44] proposed adopting a cross-frame attention module and video-specific text prompts for remarkable video "zero-shot" generalization ability. EVL [39] employed frozen CLIP models to extract video features for efficient video learning. Pan *et al.* [45] inserted spatial-temporal adapters (ST-Adapter) into pre-trained transformer blocks to enable space-time modeling capabilities in image models.

EVL [39] is mostly related to ours. It uses a transformer decoder to collect spatial information from frozen features, which is also back-propagation-free for pre-trained parameters. However, our DiST adopts a dual encoder structure to exploit video specific temporal changes, and enjoys both strong space-time modeling and high training efficiency.

Video Recognition. One of the key aspects of video recognition is exploring temporal patterns in videos. For convolutional-based methods [54, 59, 50, 66, 7, 63, 60, 16, 61, 38, 73, 64, 20, 14], which introduce 3D convolutions [59, 7], factorized spatial and temporal convolutions [50, 61, 16], and convolutional modules with temporal modeling capabilities [25, 38, 51, 35, 62, 20]. Due to the limited receptive field of convolutional networks, Transformer-based approaches [1, 3, 13, 36, 53, 32, 40, 47, 43, 4] with global attention have achieved promising performance. For example, ViViT [1] and Transformer [40] achieve space-time modeling by factorized spatial and temporal transformers and window attention, respectively. Apart from designing the model architecture, recently, self-supervised video representation learning [46, 17, 49, 22, 11, 15, 58, 69, 68] has also gained popularity due to its impressive performance.

Our multi-branch design shares similar spirit with SlowFast [16] and Multiview Transformers [72], which both design different views for similar network structures, and all parameters require back-propagation for training. Nevertheless, our work designs an asymmetric network structure with strong temporal modeling capability, and gets rid of back-propagation for massive parameters.

3. Approach

In this work, we seek to empower the large-scale pre-trained language-image models with spatial-temporal mod-

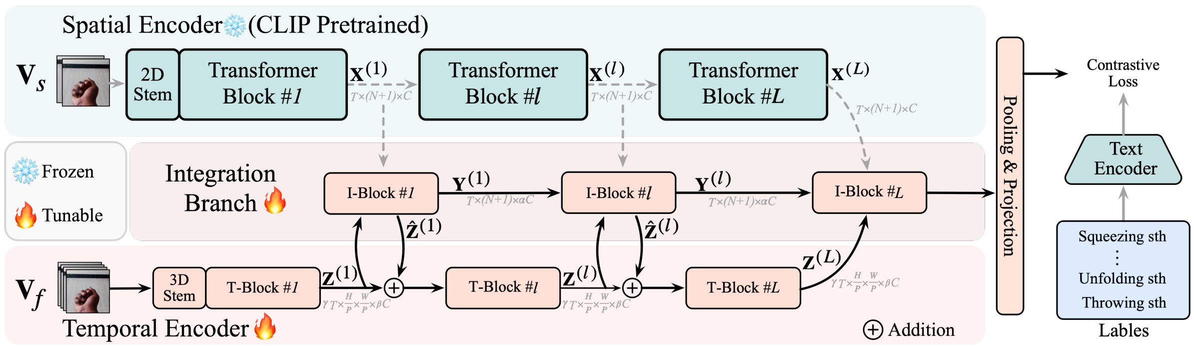


Figure 3: The overall framework of DiST. DiST contains three components, including a spatial encoder, a temporal encoder and an integration branch. The spatial encoder is a CLIP [52] pre-trained Vision Transformer (ViT) [12], which is frozen and back-propagation-free in training. The temporal encoder is composed of a series of lightweight temporal blocks (T-Block), which is responsible for capturing dynamic temporal patters in videos. The integration branch consists of multiple interaction blocks (I-Block), which are designed to integrate the features from the spatial encoder and the temporal encoder.

eling capability in training efficient way. Specifically, as shown in Fig. 3, the proposed DiST comprises three components: the spatial encoder, temporal encoder, and integration branch. The spatial encoder is a heavy CLIP pre-trained Vision Transformer (ViT), which extracts frozen features for sparse frames with powerful spatial semantics. The temporal encoder is a lightweight spatio-temporal network with low-channel capacity, adopting dense frames as input and capturing temporal patterns specific to video understanding. The integration branch links the spatial and temporal encoders by interacting with the disentangled spatial and temporal features. In this section, we first briefly present the formulation of the spatial encoder in Sec. 3.1. Then, the temporal encoder and integration branch are elaborated in Sec. 3.2 and Sec. 3.3, respectively. Finally, the training loss is introduced in Sec. 3.4.

3.1. Spatial Encoder

In DiST, the spatial encoder is an off-the-shelf feature extractor without recording the gradients, resulting in significant efficiency improvements during training. It extracts independent spatial features from several sparse frames. Given a video clip $\mathbf{V}_s \in \mathbb{R}^{T\times H\times W\times 3}$, where T, H and W are the frame number, height, and width, respectively. Following ViT [12], each frame is split into $N = \frac{H}{P} \times \frac{W}{P}$ patches, and the patch size is denoted as $P \times P$. Then, these small patches are projected by a fully connected layer, *i.e.*, the 2D stem in Fig. 3, which generates a sequence of patch embeddings $[\mathbf{x}^{(0)}_{t,1}, \mathbf{x}^{(0)}_{t,2}, \cdots, \mathbf{x}^{(0)}_{t,N}]$, where $t = \{1, \cdots, T\}$ is the frame index. Next, an additional learnable token $\mathbf{x}_{\text{cls}}$ is concatenated for each frame, and the full inputs of Transformer blocks are denoted as:

$$\mathbf{X}^{(0)}_t = [\mathbf{x}^{(0)}_{t,\text{cls}}, \mathbf{x}^{(0)}_{t,1}, \mathbf{x}^{(0)}_{t,2}, \cdots, \mathbf{x}^{(0)}_{t,N}] + \mathbf{e}^{\text{spatial}}, \quad (1)$$

where the $(N+1)$ embeddings are enhanced with the trainable spatial position embedding $\mathbf{e}^{\text{spatial}}$. Assuming that the

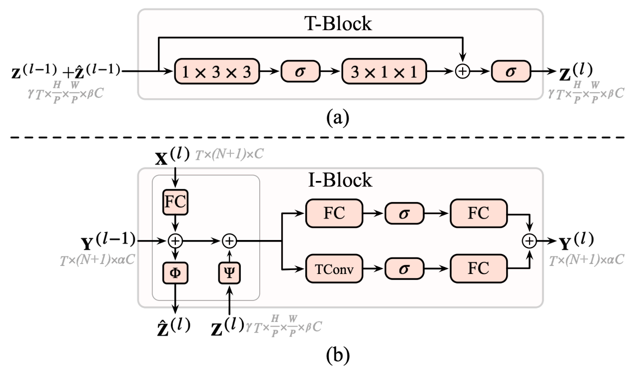


Figure 4: **(a)** shows the structural details of Temporal Block (T-Block). Our default structure is R(2+1)D [61]. **(b)** illustrates the structural details of Interaction Block (I-Block).

spatial encoder has L Transformer blocks, the features of the l_{th} layer for the t_{th} frame can be extracted by:

$$\mathbf{X}^{(l)}_t = \text{Transformer}^{(l)}(\mathbf{X}^{(l-1)}_t) \in \mathbb{R}^{(N+1)\times C}, \quad (2)$$

where $l = \{1, \cdots, L\}$ refers to the layer index and C is the channel dimension. We adopt the notation $\mathbf{X}^{(l)} = [\mathbf{X}^{(l)}_1, \cdots, \mathbf{X}^{(l)}_T] \in \mathbb{R}^{T\times(N+1)\times C}$ to represent the spatial features of T frames in the l_{th} layer.

3.2. Temporal Encoder

With powerful spatial semantics from the CLIP pre-trained spatial encoder, we expect to train a lightweight temporal-specific network to disentangle the fine-grained motion learning for video understanding. Therefore, we design a general temporal encoder, which can utilize the original video frames as input and receive the semantic guidance from the spatial encoder. Without losing generality, we assume that the number of frames in the temporal encoder is γT, $\gamma \in \{1, 2, 4\}$, which means that the temporal input $\mathbf{V}_f \in \mathbb{R}^{\gamma T\times H\times W\times 3}$ can be sampled around the spatial input $\mathbf{V}_s$ by γ times. Next, $\mathbf{V}_f$ is projected by a 3D convolution, *i.e.*, the 3D stem in Fig. 3, for patch embedding. The kernel size and stride of the 3D stem in spatial dimension are both P to align with the spatial size in the spatial encoder for feature integration. Thus the projected temporal features can be formulated as: $\mathbf{Z}^{(0)} = \text{Conv3d}(\mathbf{V}_f) \in \mathbb{R}^{\gamma T \times \frac{H}{P} \times \frac{W}{P} \times \beta C}$. Here, $\beta \in \{\frac{1}{24}, \frac{1}{12}, \frac{1}{6}, \frac{1}{4}\}$, indicates the channel reduction rate of the temporal encoder. Note that we do not perform temporal downsampling to preserve more temporal details. Then, a series of lightweight Temporal Blocks (T-Block) are designed to extract spatio-temporal patterns for these frames, which can be written as:

$$\mathbf{Z}^{(l)} = \text{T-Block}^{(l)}(\mathbf{Z}^{(l-1)} + \hat{\mathbf{Z}}^{(l-1)}) \in \mathbb{R}^{\gamma T \times \frac{H}{P} \times \frac{W}{P} \times \beta C}, \tag{3}$$

where the function T-Block$(\cdot)$ is the smallest unit that can perform spatio-temporal modeling. As shown in Fig. 4 (a), the classic R(2+1)D [61] is adopted for T-Block$(\cdot)$ by default. Meanwhile, other optional designs, such as the convolution-based C3D [59], TAdaConv [20] and the transformer-based joint space-time transformer [58], have also been explored in Tab. 1c. Although the joint transformer generates a large self-attention matrices, it still feasible due to the low channel capacity of the temporal encoder. The $\hat{\mathbf{Z}}^{(l-1)}$ indicates the interaction features from the integration branch. It will be introduced in the next section.

3.3. Integration Branch

The role of the integration branch can be summarized into two aspects: *(i)* receiving and integrating the spatial and temporal features into more discriminative spatio-temporal representations; *(ii)* performing feature interactions between the spatial encoder and temporal encoder. Specifically, it transfers the powerful semantics from the spatial encoder to temporal encoder, thus guiding the random initialized temporal encoder to capture temporal clues.

The integration branch is composed of a series of Interaction Blocks (I-Block). The structural details of one I-Block are shown in Fig. 4. Formally, for $\mathbf{X}^{(l)}$ from the spatial encoder and $\mathbf{Z}^{(l)}$ from the temporal encoder, we first adopt addition to absorb them into the integrated features $\mathbf{Y}^{(l-1)} \in \mathbb{R}^{T \times (N+1) \times \alpha C}$ from the previous layer, and then perform spatio-temporal fusion by a temporal Feed Forward Network. This can be expressed as:

$$\begin{aligned}
&\hat{\mathbf{Y}}^{(l)} = \mathbf{Y}^{(l-1)} + \text{FC}(\mathbf{X}^{(l)}) + \Psi^{(l)}(\mathbf{Z}^{(l)}),\\
&\mathbf{Y}^{(l)} = \text{FC}(\sigma(\text{FC}(\text{LN}(\hat{\mathbf{Y}}^{(l)})))) + \text{FC}(\sigma(\text{TConv}(\text{LN}(\hat{\mathbf{Y}}^{(l)})))).
\end{aligned} \tag{4}$$

Here, $\mathbf{Y}^{(0)}$ is 0. FC$(\mathbf{X}^{(l)})$ reduces the channel dimension from C to αC. $\Psi^{(l)}(\cdot)$ is the lateral interaction from the temporal encoder to the integration branch that will be discussed later. FC$(\cdot)$, LN$(\cdot)$, TConv$(\cdot)$ and $\sigma(\cdot)$ are abbreviations of the linear layer, layer normalization, 1D convolution with the kernel size of $3 \times 1 \times 1$ and activation function, respectively. The 1D convolution introduced here is to further encourage the spatio-temporal blending for the disentangled spatial and temporal features.

Based on the elaborately designed architecture mentioned above, the integration branch is capable of simultaneously receiving the spatial semantics and temporal patterns, and then integrating them into unified spatio-temporal representations for video recognition.

Next, we discuss the interaction details between the integration branch and the temporal encoder. First, the interaction function $\Psi(\cdot)$ is responsible for transmitting the information from the temporal encoder (*i.e.*, $\mathbf{Z}^{(l)}$) to the integration branch. In implementation, to align with the feature size of the integration branch, $\Psi(\cdot)$ needs to downsample the temporal dimension of $\mathbf{Z}^{(l)} \in \mathbb{R}^{\gamma T \times \frac{H}{P} \times \frac{W}{P} \times \beta C}$ from γT to T, then increases the channels from βC to αC, and appends a new class token to align the number of tokens (*i.e.*, $N+1$) in $\mathbf{Y}^{(l-1)}$. Formally, $\Psi(\cdot)$ can be written as:

$$\Psi(\mathbf{Z}) = [\text{Flatten}(\text{DConv}(\mathbf{Z})), \mathbf{z}_{\text{cls}}] \in \mathbb{R}^{T \times (N+1) \times \alpha C}, \tag{5}$$

where DConv$(\cdot)$ is a temporal convolution for temporal downsampling, whose kernel size and stride are both γ in temporal dimension. The input and output channels of DConv$(\cdot)$ are βC and αC. The function Flatten$(\cdot)$ flattens the spatial dimension $\frac{H}{P} \times \frac{W}{P}$ to N. $\mathbf{z}_{\text{cls}} \in \mathbb{R}^{T \times 1 \times \alpha C}$ is a trainable class token. $[\cdot, \cdot]$ indicates the concatenation. In this way, the function $\Psi(\cdot)$ realizes the feature alignment of the temporal branch and the integration branch.

The interaction from the integration branch to the temporal encoder is defined in Eq. 3, *i.e.*, the notation $\hat{\mathbf{Z}}^{(l-1)}$. For simplicity, we use $\hat{\mathbf{Z}}^{(l)}$ instead of $\hat{\mathbf{Z}}^{(l-1)}$ for discussion. Therefore, $\hat{\mathbf{Z}}^{(l)}$ can be formulated as:

$$\hat{\mathbf{Z}}^{(l)} = \Phi(\mathbf{Y}^{(l-1)} + \text{FC}(\mathbf{X}^{(l)})). \tag{6}$$

Here, $\mathbf{Y}^{(l-1)}$ is the integrated feature from the previous layer and $\mathbf{X}^{(l)}$ is the spatial feature from the current layer. For $\hat{\mathbf{Z}}^{(0)}$, we set it to 0. This design can ensure that the spatio-temporal semantic guidance can be timely injected into the temporal encoder.

For the function $\Phi(\cdot)$, which is responsible for the compatibility of the integration features ($\mathbb{R}^{T \times (N+1) \times \alpha C}$) with the temporal features ($\mathbb{R}^{\gamma T \times \frac{H}{P} \times \frac{W}{P} \times \beta C}$). Therefore, we first remove the spatial class token, and reduce the channels from αC to βC by a linear layer. Then, to upsample the temporal dimension from T to γT, we adopt the nearest interpolation by default. Finally, the N tokens of each frame are reshaped to $\frac{H}{P} \times \frac{W}{P}$ to align with the temporal features in Eq. 3.

Unless particularly emphasized, the above-discussed downsampling and upsampling methods are the default implementation for $\Psi(\cdot)$ and $\Phi(\cdot)$, respectively. There are also other alternative implementations, such as the temporal convolution in function $\Psi(\cdot)$ can be replaced with a combination of a pooling layer and a linear layer, and the nearest

Temp.Encoder	Integ.Branch	SSV2	K400
EVL [39]		61.0	82.9
✗	✗	55.0	79.9
✓	✗	63.2	81.8
✗	✓	65.9	83.0
✓	✓	**68.7**	**83.6**

(a) "Temp." is the abbreviation of "temporal". "Integ." is the abbreviation of "Integration".

Integ.→Temp.	Temp.→Integ.	SSV2	K400
✗	✗	67.5	83.0
✓	✗	67.9	83.4
✗	✓	67.7	83.1
✓	✓	**68.7**	**83.6**

(b) "Integ.→Temp." indicates the interactions from the integration branch to the temporal encoder, and vice versa.

case	SSV2	K400	GFLOPs
TAda [20]	67.8	82.4	162.0
C3D [59]	67.8	82.6	168.2
J. Trans. [58]	67.6	83.5	165.0
R(2+1)D [61]	**68.7**	**83.6**	**163.1**

(c) Optional designs of T-Block. "J. Trans." is the space-time joint transformer.

Spat.	Temp.	γ	SSV2	K400	GFLOPs
	8f	1	67.9	83.4	**158.7**
8f	16f	2	68.7	**83.6**	163.1
	32f	4	**69.1**	83.3	171.6
	64f	8	68.5	83.6	188.8

(d) Varying values of γ, *i.e*., the number of frames in the temporal encoder.

Dim	α	SSV2	K400	GFLOPs
96	1/8	62.6	79.0	**149.4**
192	1/4	66.7	82.0	152.8
384	1/2	**68.7**	**83.6**	163.1
768	1	68.7	83.3	196.1

(e) Varying values of α, *i.e*., the channel capacity of the integration branch.

Dim	β	SSV2	K400	GFLOPs
64	1/12	67.9	83.2	**159.7**
96	1/8	68.7	**83.6**	163.1
128	1/6	68.6	83.3	167.4
192	1/4	**68.9**	83.4	178.7

(f) Varying values of β, *i.e*., the channel capacity of the temporal encoder.

Table 1: Ablations on **Something-Something V2** and **Kinetics-400**. Our spatial encoder is a 8-frame vanilla ViT-B/16 pre-trained by CLIP [52] with a channel width of 768. The TSN [65] uniform sampling is performed on both datasets. The inference protocol of all models and datasets are 3 clips $\times$ 1 center crop.

interpolation in $\Phi(\cdot)$ can be replaced with trilinear interpolation or deconvolution. These optional designs will be further explored in experiments, *i.e*., Sec. 4.

3.4. Training Loss

The integration of the disentangled spatial and temporal information yields semantically rich spatio-temporal representations, which can lead to more promising video recognition performances. Next, following CLIP [52], we first perform adaptive pooling to obtain a video-level class token $\mathbf{y}_{\text{cls}}$ for the representation of $\mathbf{Y}^{(L)}$. Then, the text features of the correct category labels are taken as positives, and contrastive loss is employed to train both the temporal encoder and integration branch. The formulation can be written as:

$$\begin{aligned} \mathbf{y}_{\text{cls}} &= \text{Proj}(\text{AdaPooling}(\mathbf{Y}^{(L)})), \\ \mathcal{L}_{\text{CL}} &= -\log \frac{\exp(\text{sim}(\mathbf{y}_{\text{cls}}, \mathbf{u}_i)/\tau)}{\sum_{k=1}^{M} \exp(\text{sim}(\mathbf{y}_{\text{cls}}, \mathbf{u}_k)/\tau)}, \end{aligned} \tag{7}$$

where $\text{Proj}(\cdot)$ indicates the projection to the classification space. $\text{sim}(\cdot,\cdot)$ is the normalized cosine similarity. $\mathbf{u}_i$ is a text feature for the i_{th} label. Here, we assume the correct label of $\mathbf{y}_{\text{cls}}$ is i. τ refers to the temperature parameter. In this manner, the proposed structure can project videos into a text space, which not only enables the video recognition but also retains the zero-shot ability for videos.

4. Experiments

4.1. Implementation

Datasets. We evaluate our proposed DiST on five widely used benchmarks, *i.e*., Kinetics-400 (K400) [29], Something-Something V2 (SSV2) [19], Epic-Kitchens-100 (EK100) [10], HMDB51 [23], and UCF101 [55]. K400 is a large scale action recognition dataset spanning 400 different human actions. SSv2 is a commonly used temporally-heavy dataset. EK100 is a egocentric recorded interaction between persons and objects in the kitchen. Each video is labeled with a verb and a noun. UCF101 and HMDB51 are two relatively small action recognition datasets, which are employed for zero-shot evaluation following [44].

Architecture. Following previous work [39], we use the CLIP [52] pre-trained ViT-B/16, ViT-L/14 and ViT-L/14-336p as our spatial encoder. Unless otherwise specified, we mark the default settings in the temporal encoder and the integration branch in gray in Sec. 4.2.

Training settings. All training and testing settings are provided in Appendix.

4.2. Ablation Studies

The role of the temporal encoder and integration branch. In Tab. 1a, we attempt to remove the temporal encoder and integration from DiST to observe their effects. Compared with the spatial encoder only (the 1st line), imposing temporal encoder and the integration branch can both significantly boost performance, for example, the improvements on SSV2 reach 10.9% and 8.2%, respectively. DiST without integration branch cannot interact and integrate the independent spatial and temporal information, resulting in poorer performance. However, with the integration branch, incorporating the temporal encoder to learn more video-specific features further yields a gain of 2.8%, which proves the necessity of the disentangled temporal encoder.

The feature interactions between the temporal encoder

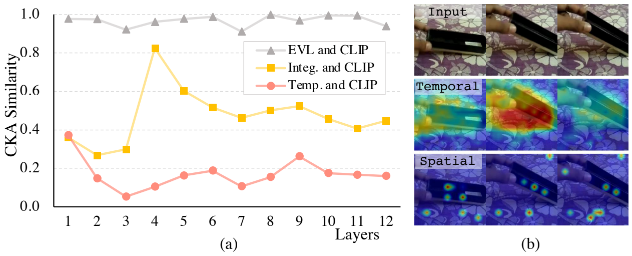


Figure 5: **(a)** We evaluate three different pairs of feature correlations by CKA similarities [30]: *(i)* EVL [39] features and CLIP features; *(ii)* our integrated features and CLIP features; *(iii)* our temporal features CLIP features. **(b)** Visualization of the magnitude of features. Red indicates a large magnitude of feature activation values, while blue indicates a small magnitude.

and integration branch are explored in Tab. 1b. One can observe that both directions of information transmission can improve performances, and the combination of the two can boost accuracy more significantly, *e.g.*, the improvement can reach 1.2% on SSV2. This demonstrates that the spatio-temporal blending in the integration branch and the spatial semantic guidance in temporal encoder are both essential.

Optional designs of T-Block. The T-Block is intended to empower the lightweight temporal encoder with temporal modeling capabilities. Here, we attempt three different modules in Tab. 1c, *i.e.*, the convolution-based TAdaConv [20], C3D [59] and R(2+1)D [61], the transformer-based joint spatial-temporal Transformer [58]. The R(2+1)D outperforms the other approaches by about 1% with less computation on SSV2. We speculate that the lighter R(2+1)D may be easier to optimize.

The parameter analysis of γ, α and β. *(i)* Our temporal encoder can receive flexible frames as input. γ determines the number of frames input to the temporal encoder. Tab. 1d shows that more frames can introduce richer temporal clues and consistently boost the accuracies, especially for the temporally-heavy dataset (*i.e.*, SSV2). However, for computation efficiency, we set γ to 2 by default, which can produce 0.8% gain on SSV2 compared with $\gamma = 1$. *(ii)* α determines the channel width of the integration branch. As in Tab. 1e, when channel width is 96 ($\alpha = 1/8$), compared with 384 ($\alpha = 1/2$), the performance degradation is 6.1% (68.7% vs. 62.6%) on SSV2. This is because the integration branch requires a larger channel width to accommodate the rich semantics in spatio-temporal fusion. Nevertheless, the channel width of 768 ($\alpha = 1$) can not further improve accuracy, which means that 384 is sufficient. *(iii)* Tab. 1f explores the impact of the channel dimension of the temporal encoder. Since the spatial encoder provides powerful spatial semantics, the temporal encoder only needs to capture specific motions in videos. When the channel dimension is 96 ($\beta = 1/8$), it can achieve satisfactory performance, the gain

Downsampling	Top-1
Avg Pooling	68.3
Max Pooling	68.4
DConv	**68.7**

(a)

Upsampling	Top-1
DeConv	67.8
Trilinear	68.3
Nearest	**68.7**

(b)

Table 2: Alternative designs for feature interactions on SSV2. **(a)** Replacing the downsampling function in $\Psi(\cdot)$ with different pooling functions. **(b)** Replacing the nearest interpolation in $\Phi(\cdot)$ with different upsampling functions. "DeConv" is deconvolution.

Method	Training			Inference
	Memory	Step	Epoch	Throughput
ST-Adapter [45]	39.10G	0.95s	38	**50**
EVL [39]	15.05G	0.71s	45	38
DiST	**12.68G**	**0.52s**	**36**	48

Table 3: Training and inference costs on SSV2. "Step": the training step time with batch size of 32 on A100-80G. "Throughput": inference throughput (Videos/s).

is 0.8% compared with smaller 64 dimensions. Meanwhile, more channels are also saturated, which implies designing a lightweight temporal encoder is reasonable.

Has the temporal encoder learned video-specific representations? To demonstrate this, we first utilized CKA similarity [30] to analyze the feature correlation between various video features and the CLIP pre-trained image features. As shown in Fig. 5 (a), we can see that the video features learned by EVL [39] are highly correlated with the image features generated by CLIP. This explains the reason why EVL has weak temporal modeling ability. However, the correlations of our integrated feature, temporal feature are gradually weakened, which fully demonstrates that decoupling spatio-temporal learning indeed enables the temporal encoder to capture temporal patterns largely complementary to spatial features. In Fig.5 (b), we further visualize the activation amplitude of the features from the temporal encoder and the spatial encoder. As can be seen, the temporal is more sensitive to the inter-frame motion, thus facilitating the learning of video-specific features.

Optional designs for feature interactions. In Tab. 2, we evaluate the downsampling ways in function $\Psi(\cdot)$ and upsampling ways in function $\Phi(\cdot)$ for feature interactions. First, for downsampling, the learnable temporal convolution (*i.e.*, DConv) slightly outperformed the pooling methods by around 0.3%. Intriguingly, for the upsampling methods, the performance with the learnable deconvolution is worse. We speculate that the deconvolution and trilinear incorporates adjacent frames, resulting in spatial semantic shifts. We thus employ DConv and nearest as our default.

Training and inference consumption. In Tab. 3, we compare the training and inference time with existing efficient fine-tuning approaches under the same hardware. Firstly,

Method	Pre-train	Architecture	Input Size	FLOPs×Cr.×Cl. (T)	Param (M)	Frozen	Top-1	Top-5
SlowFast [16]	ImageNet-21K	R101+NL	16×224^2	$0.1 \times 3 \times 1$	60	✗	63.1	87.6
ViViT FE [1]	IN21K+K400	ViT-L	16×224^2	$1.0 \times 3 \times 4$	612	✗	65.4	89.8
MTV-B(320p) [72]	IN21K+K400	-	32×224^2	$0.9 \times 3 \times 4$	310	✗	68.5	90.4
MViT [13]	Kinetics-600	MViT-B-24	32×224^2	$0.2 \times 3 \times 1$	53	✗	68.7	91.5
Video Swin [40]	IN21K+K400	Swin-B	32×224^2	$0.3 \times 3 \times 1$	60	✗	69.6	92.7
TAdaConvNeXtV2 [21]	IN1K+K400	ConvNeXt-S	32×224^2	$0.2 \times 3 \times 2$	82	✗	70.0	92.0
EVL❄ [39]	CLIP-400M	ViT-B	32×224^2	$0.68 \times 1 \times 3$	175	✓	62.4	-
ST-Adapter❄ [45]	CLIP-400M	ViT-B	32×224^2	$0.61 \times 1 \times 3$	93	✓	69.5	**92.6**
DiST$_{\gamma=2}$❄	CLIP-400M	ViT-B	8×224^2	$0.16 \times 1 \times 3$	105	✓	68.7	91.1
DiST$_{\gamma=2}$❄	CLIP-400M	ViT-B	16×224^2	$0.32 \times 1 \times 3$	105	✓	70.2	92.0
DiST$_{\gamma=2}$❄	CLIP-400M	ViT-B	32×224^2	$0.65 \times 1 \times 3$	105	✓	**70.9**	92.1
UnifromerV2 [31]	CLIP-400M	ViT-L	32×224^2	$1.73 \times 1 \times 3$	574	✗	73.0	94.5
TAdaFormer [21]	CLIP-400M	ViT-L	32×224^2	$1.70 \times 2 \times 3$	364	✗	73.6	-
EVL❄ [39]	CLIP-400M	ViT-L	32×224^2	$3.21 \times 1 \times 3$	654	✓	66.7	-
EVL❄ [39]	CLIP-400M	ViT-L	32×336^2	$8.08 \times 1 \times 3$	654	✓	68.0	-
ST-Adapter❄ [45]	CLIP-400M	ViT-L	32×224^2	$2.75 \times 1 \times 3$	347	✓	72.3	93.9
DiST$_{\gamma=2}$❄	CLIP-400M	ViT-L	8×224^2	$0.71 \times 1 \times 3$	336	✓	70.8	92.3
DiST$_{\gamma=2}$❄	CLIP-400M	ViT-L	16×224^2	$1.42 \times 1 \times 3$	336	✓	72.5	93.0
DiST$_{\gamma=2}$❄	CLIP-400M	ViT-L	32×224^2	$2.83 \times 1 \times 3$	336	✓	**73.1**	**93.2**

Table 4: Comparison with the state-of-the-art methods on Something-Something V2. “Cr.” and “Cl.” are the abbreviation for “spatial crops” and “temporal clips”. “Frozen” indicates freezing the CLIP pre-trained parameters.

Method	Model	Frames	HMDB51	UCF101
ActionCLIP [67]	B/16	32×1×1	40.8±5.4	58.3±3.4
X-CLIP [44]	B/16	32×1×1	44.6±5.2	72.0±2.3
DiST$_{\gamma=2}$❄	B/16	32×1×1	55.4±1.2	72.3±0.6
DiST$_{\gamma=2}$❄	L/14	32×1×1	**57.5**±1.6	**74.9**±0.8

(a)

Method	Model	Frames	Verb	Noun	Action
EVL❄ [39]	B/16	8×3×1	62.7	51.0	37.7
ST-Adapter❄ [67]	B/16	8×3×1	67.6	55.0	-
DiST$_{\gamma=2}$❄	B/16	8×3×1	69.5	58.1	45.8
DiST$_{\gamma=2}$❄	L/14	8×3×1	**70.7**	**61.6**	**48.9**

(b)

Table 5: Comparison with the state-of-the-art CLIP-based methods on three datasets. “❄”: frozen backbone. **(a)** Zero-shot accuracy on HMDB51 [23] and UCF101 [55] across three splits. **(b)** Results on the Epic-Kitchens-100 [10] validation set.

in training, as a back-propagation-free approach, the GPU memory consumption of our DiST is merely 32% (*i.e.*, 12.68G *v.s.* 39.10G) of ST-Adapter [45]. Benefiting from the lightweight design of the temporal encoder, the training step time is only 73% of EVL [39], which is also based on the back-propagation-free backbone. Moreover, the training epochs required by DiST is also less than that of EVL [39] and ST-Adapter [45], since the complete reliance on image-specific features for spatio-temporal learning may potentially pose additional challenges. Secondly, in inference, we test the throughput for the above methods with batch size of 32. Since ST-Adapter [45] has no additional branches, its throughput is slightly higher than our DiST. However, compared with the similar EVL, our throughput is increased by 1.26 × (*i.e.*, from 38 Videos/s to 48 Videos/s).

4.3. Comparison with State-of-the-art

Zero-shot experiments. Due to the retention of the frozen text branch, our approach is still able to conduct zero-shot tasks. We employ the 32-frame Kinetics-400 fine-tuned models in Tab. 6 for evaluation. As in Tab. 5a, with the same pre-trained model (*i.e.*, ViT-B/16), our method remarkably outperforms existing fully fine-tuned X-CLIP [44] by 10.8% on HMDB51 and is more stable across different splits. The relatively minor improvement on UCF101 is attributed to the spatially-focused dataset with limited temporal clues available for utilization. Furthermore, DiST with larger models can achieve consistent performance gains which can be attributed to the excellent architectural scalability of DiST.

Egocentric action recognition. Tab. 5b presents fair comparisons between our DiST and existing Frozen-CLIP approaches. It is evident that DiST consistently demonstrates a convincing performance advantage. With the same ViT-B/16 as spatial encoder, DiST achieves accuracy improvements of over ST-Adapter [45] by 1.9% and 3.1% on verbs and nouns, respectively. This is attributed to decoupling temporal encoder to learn representations that complement

Method	Pre-train	Architecture	Input Size	TFLOPs×Cr.×Cl.	Param (M)	Frozen	Top-1	Top-5
SlowFast [16]	-	R101+NL	16×224^2	$0.4 \times 3 \times 10$	60	✗	79.8	93.9
TimeSformer [3]	ImageNet-21K	ViT-L	96×224^2	$8.4 \times 3 \times 1$	430	✗	80.7	94.7
MViT [13]	-	MViT-B	64×224^2	$0.5 \times 1 \times 5$	37	✗	81.2	95.1
ViViT FE [1]	ImageNet-21K	ViT-L	128×224^2	$4.0 \times 3 \times 1$	N/A	✗	81.7	93.8
Video Swin [40]	ImageNet-21K	Swin-L	32×224^2	$0.6 \times 3 \times 4$	197	✗	83.1	95.9
TAdaConvNeXtV2 [21]	ImageNet-21K	ConvNeXt-B	32×224^2	$0.3 \times 3 \times 4$	146	✗	83.7	-
ST-Adapter❄ [45]	CLIP-400M	ViT-B	32×224^2	$0.61 \times 1 \times 3$	93	✓	82.7	96.2
EVL❄ [39]	CLIP-400M	ViT-B	32×224^2	$0.59 \times 1 \times 3$	115	✓	84.2	-
DiST$_{\gamma=2}$❄	CLIP-400M	ViT-B	8×224^2	$0.16 \times 1 \times 3$	112	✓	83.6	96.3
DiST$_{\gamma=2}$❄	CLIP-400M	ViT-B	16×224^2	$0.32 \times 1 \times 3$	112	✓	84.4	96.7
DiST$_{\gamma=2}$❄	CLIP-400M	ViT-B	32×224^2	$0.65 \times 1 \times 3$	112	✓	85.0	97.0
DiST$_{\gamma=2}$❄	CLIP-400M+K710	ViT-B	32×224^2	$0.65 \times 1 \times 3$	112	✓	**86.8**	**97.5**
UnifromerV2 [31]	CLIP-400M+K710	ViT-L	32×224^2	$2.66 \times 2 \times 3$	354	✗	89.3	98.2
TAdaFormer [21]	CLIP-400M+K710	ViT-L	32×224^2	$1.41 \times 4 \times 3$	364	✗	89.5	-
ST-Adapter❄ [45]	CLIP-400M	ViT-L	32×224^2	$2.75 \times 1 \times 3$	347	✓	87.2	97.6
EVL❄ [39]	CLIP-400M	ViT-L	32×224^2	$2.70 \times 1 \times 3$	363	✓	87.3	-
DiST$_{\gamma=2}$❄	CLIP-400M	ViT-L	8×224^2	$0.71 \times 1 \times 3$	343	✓	86.9	97.6
DiST$_{\gamma=2}$❄	CLIP-400M	ViT-L	16×224^2	$1.42 \times 1 \times 3$	343	✓	87.6	97.8
DiST$_{\gamma=2}$❄	CLIP-400M	ViT-L	32×224^2	$2.83 \times 1 \times 3$	343	✓	88.0	97.9
DiST$_{\gamma=2}$❄	CLIP-400M+K710	ViT-L	32×224^2	$2.83 \times 1 \times 3$	343	✓	**89.5**	**98.4**
X-CLIP [44]	CLIP-400M	ViT-L	16×336^2	$3.09 \times 3 \times 4$	354	✗	87.7	97.4
BIKE [71]	CLIP-400M	ViT-L	32×336^2	$3.73 \times 3 \times 4$	230	✗	88.6	98.3
EVL❄ [39]	CLIP-400M	ViT-L	32×336^2	$6.07 \times 1 \times 3$	363	✓	87.7	-
Text4Vis❄ [70]	CLIP-400M	ViT-L	32×336^2	$3.83 \times 1 \times 3$	231	✓	87.8	97.6
UnifromerV2❄ [31]	CLIP-400M+K710	ViT-L	32×336^2	$6.27 \times 1 \times 3$	354	✓	88.8	98.1
DiST$_{\gamma=2}$❄	CLIP-400M	ViT-L	32×336^2	$6.64 \times 1 \times 3$	343	✓	88.5	98.2
DiST$_{\gamma=2}$❄	CLIP-400M+K710	ViT-L	32×336^2	$6.64 \times 1 \times 3$	343	✓	**89.7**	**98.5**

Table 6: Comparison with state-of-the-arts on Kinetics-400.

frozen spatial features. Moreover, DiST continues to provide sustained benefits even on larger models.

Video recognition on SSV2 and K400. First, for the temporally-heavy dataset, *i.e*., SSV2 in Tab. 4, DiST outperforms other CLIP-based efficient fine-tuning approach by a notable margin. For example, compared with EVL [39] that also uses the frozen CLIP features, DiST surpasses it by 8.5% with a 32-frame ViT-B. Compared with fully fine-tuned UniformerV2 [31], our efficient DiST still achieve comparable accuracy. Second, on the spatially-heavy K400[29], DiST is still highly competitive. Compared with EVL with better performance, DiST can always achieve improvements around 0.8% regardless of the pre-training models. With these observations, we can summarize that DiST enjoys the dual advantages of spatial modeling and temporal modeling. Besides, following UniformerV2, we pre-train the lightweight temporal encoder and integration branch on a large-scale video dataset, *i.e*., Kinetics-710 [29, 5, 6], and the performances are further improved. When inputting 32 frames with 336×336 size, our approach exceeds UniformerV2 by 0.9% using ViT-L model, which can demonstrate the strong data scalability of DiST.

5. Conclusion

In this work, we propose DiST, an image-to-video transfer learning framework that enjoys both training efficiency and powerful temporal modeling capabilities. It is a dual-encoder structure, which includes a frozen but heavy spatial encoder and a lightweight learnable temporal encoder. Then, an integration branch fuses the spatial and temporal information into the unified spatio-temporal representations for video understanding. Extensive experiments verify the scalability of DiST in both model size and data scale. We hope that our DiST can provide some inspiration for researchers who are interested in large-scale video models.

6. Acknowledgement

This work is supported by the National Natural Science Foundation of China under grant U22B2053 and 62176097, and by Alibaba DAMO Academy through Alibaba Research Intern Program.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, pages 6836–6846, 2021. 1, 2, 7, 8

[2] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Visual prompting: Modifying pixel space to adapt pre-trained models. *arXiv preprint arXiv:2203.17274*, 2022. 2

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021. 1, 2, 8

[4] Adrian Bulat, Juan Manuel Perez Rua, Swathikiran Sudhakaran, Brais Martinez, and Georgios Tzimiropoulos. Space-time mixing attention for video transformer. *NeurIPS*, 34:19594–19607, 2021. 2

[5] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018. 8

[6] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019. 8

[7] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017. 2

[8] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *arXiv preprint arXiv:2205.13535*, 2022. 1

[9] Xing Cheng, Hezheng Lin, Xiangyu Wu, Fan Yang, and Dong Shen. Improving video-text retrieval by multi-stream corpus alignment and dual softmax loss. *arXiv preprint arXiv:2109.04290*, 2021. 2

[10] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020. 2, 5, 7

[11] Ali Diba, Vivek Sharma, Reza Safdari, Dariush Lotfi, Saquib Sarfraz, Rainer Stiefelhagen, and Luc Van Gool. Vi2clr: Video and image for visual contrastive learning of representation. In *ICCV*, pages 1502–1512, 2021. 2

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 1, 3

[13] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *ICCV*, pages 6824–6835, 2021. 2, 7, 8

[14] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, pages 203–213, 2020. 2

[15] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. *NeurIPS*, 2022. 2

[16] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, pages 6202–6211, 2019. 1, 2, 7, 8

[17] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. In *CVPR*, pages 3299–3309, 2021. 2

[18] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021. 1, 2

[19] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *ICCV*, pages 5842–5850, 2017. 1, 2, 5

[20] Ziyuan Huang, Shiwei Zhang, Liang Pan, Zhiwu Qing, Mingqian Tang, Ziwei Liu, and Marcelo H Ang Jr. Tada! temporally-adaptive convolutions for video understanding. In *ICLR*, 2022. 2, 4, 5, 6

[21] Ziyuan Huang, Shiwei Zhang, Liang Pan, Zhiwu Qing, Yingya Zhang, Ziwei Liu, and Marcelo H Ang Jr. Temporally-adaptive models for efficient video understanding. *arXiv preprint arXiv:2308.05787*, 2023. 7, 8

[22] Simon Jenni and Hailin Jin. Time-equivariant contrastive video representation learning. In *ICCV*, pages 9970–9980, 2021. 2

[23] H Jhuang, H Garrote, E Poggio, T Serre, and T Hmdb. A large video database for human motion recognition. In *ICCV*, volume 4, page 6, 2011. 2, 5, 7

[24] 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*, pages 4904–4916. PMLR, 2021. 2

[25] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *ICCV*, pages 2000–2009, 2019. 2

[26] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *ECCV*, pages 105–124. Springer, 2022. 1

[27] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *ECCV*, pages 105–124. Springer, 2022. 2

[28] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014. 1

[29] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1, 2, 5, 8

[30] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *ICML*, volume 97, pages 3519–3529. PMLR, 09–15 Jun 2019. 6

[31] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Limin Wang, and Yu Qiao. Uniformerv2: Spatiotemporal learning by arming image vits with video uniformer. *arXiv preprint arXiv:2211.09552*, 2022. 1, 7, 8

[32] Kunchang Li, Yali Wang, Junhao Zhang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unifying convolution and self-attention for visual recognition. *TPAMI*, 2023. 2

[33] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 2

[34] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. *arXiv preprint arXiv:2212.00794*, 2022. 2

[35] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *CVPR*, pages 909–918, 2020. 2

[36] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *CVPR*, pages 4804–4814, 2022. 2

[37] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. *arXiv preprint arXiv:2210.08823*, 2022. 1, 2

[38] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, pages 7083–7093, 2019. 2

[39] Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard de Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen clip models are efficient video learners. In *ECCV*, pages 388–404. Springer, 2022. 1, 2, 5, 6, 7, 8

[40] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *CVPR*, pages 3202–3211, 2022. 2, 7, 8

[41] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, pages 9879–9889, 2020. 2

[42] 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*, pages 2630–2640, 2019. 2

[43] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *ICCV*, pages 3163–3172, 2021. 2

[44] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. Expanding language-image pretrained models for general video recognition. In *ECCV*, pages 1–18. Springer, 2022. 2, 5, 7, 8

[45] Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. Parameter-efficient image-to-video transfer learning. *arXiv e-prints*, pages arXiv–2206, 2022. 1, 2, 6, 7, 8

[46] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. In *CVPR*, pages 11205–11214, 2021. 2

[47] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and João F Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. *NeurIPS*, 34:12493–12506, 2021. 2

[48] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*, 2020. 2

[49] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *CVPR*, pages 6964–6974, 2021. 2

[50] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, pages 5533–5541, 2017. 2

[51] Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, Xinmei Tian, and Tao Mei. Learning spatio-temporal representation with local and global diffusion. In *CVPR*, pages 12056–12065, 2019. 2

[52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 1, 2, 3, 5

[53] Michael Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: Adaptive space-time tokenization for videos. *NeurIPS*, 34:12786–12797, 2021. 2

[54] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *NeurIPS*, 27, 2014. 2

[55] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2, 5, 7

[56] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, pages 7464–7473, 2019. 2

[57] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *CVPR*, pages 5227–5237, 2022. 2

[58] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *NeurIPS*, 2022. 2, 4, 5, 6

[59] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015. 1, 2, 4, 5, 6

[60] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *ICCV*, pages 5552–5561, 2019. 2

[61] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, pages 6450–6459, 2018. 2, 3, 4, 5, 6

[62] Heng Wang, Du Tran, Lorenzo Torresani, and Matt Feiszli. Video modeling with correlation networks. In *CVPR*, pages 352–361, 2020. 2

[63] Limin Wang, Wei Li, Wen Li, and Luc Van Gool. Appearance-and-relation networks for video classification. In *CVPR*, pages 1430–1439, 2018. 2

[64] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. Tdn: Temporal difference networks for efficient action recognition. In *CVPR*, pages 1895–1904, 2021. 2

[65] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36. Springer, 2016. 1, 5

[66] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *TPAMI*, 41(11):2740–2755, 2018. 2

[67] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*, 2021. 2, 7

[68] Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiyang Dai, Mengchen Liu, Yu-Gang Jiang, Luowei Zhou, and Lu Yuan. Bevt: Bert pretraining of video transformers. In *CVPR*, pages 14733–14743, 2022. 2

[69] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *CVPR*, pages 14668–14678, 2022. 2

[70] Wenhao Wu, Zhun Sun, and Wanli Ouyang. Revisiting classifier: Transferring vision-language models for video recognition. In *AAAI*, volume 37, pages 2847–2855, 2023. 8

[71] Wenhao Wu, Xiaohan Wang, Haipeng Luo, Jingdong Wang, Yi Yang, and Wanli Ouyang. Bidirectional cross-modal knowledge exploration for video recognition with pre-trained vision-language models. In *CVPR*, pages 6620–6630, 2023. 8

[72] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *CVPR*, pages 3333–3343, 2022. 1, 2, 7

[73] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *CVPR*, pages 591–600, 2020. 2

[74] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. 2

[75] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 2

[76] Renrui Zhang, Longtian Qiu, Wei Zhang, and Ziyao Zeng. Vt-clip: Enhancing vision-language models with visual-guided texts. *arXiv preprint arXiv:2112.02399*, 2021. 2

[77] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, pages 16816–16825, 2022. 2

[78] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 130(9):2337–2348, 2022. 2

[79] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *CVPR*, pages 8746–8755, 2020. 2