Revisiting Classifier: Transferring Vision-Language Models for Video Recognition

Wenhao Wu[1], Zhun Sun[2], Wanli Ouyang[3*]

[1]The University of Sydney, NSW, Australia
[2]Baidu Inc., Beijing, China
[3]Shanghai Artificial Intelligence Laboratory, Shanghai, China
whwu.ucas@gmail.com, sunzhun@baidu.com, wanli.ouyang@sydney.edu.au

Abstract

Transferring knowledge from task-agnostic pre-trained deep models for downstream tasks is an important topic in computer vision research. Along with the growth of computational capacity, we now have open-source vision-language pre-trained models in large scales of the model architecture and amount of data. In this study, we focus on transferring knowledge for video classification tasks. Conventional methods randomly initialize the linear classifier head for vision classification, but they leave the usage of the text encoder for downstream visual recognition tasks undiscovered. In this paper, we revise the role of the linear classifier and replace the classifier with different knowledge from the pre-trained model. We utilize the well-pre-trained language model to generate a good semantic target for efficient transferring learning. The empirical study shows that our method improves both the performance and the training speed of video classification, with a negligible change in the model. Our simple yet effective tuning paradigm achieves state-of-the-art performance and efficient training on various video recognition scenarios, *i.e.*, zero-shot, few-shot, and general recognition. In particular, our paradigm achieves the state-of-the-art accuracy of 87.8% on Kinetics-400, and also surpasses previous methods by 20∼50% absolute top-1 accuracy under zero-shot, few-shot settings on five video datasets. Code and models are available at https://github.com/whwu95/Text4Vis.

1 Introduction

Pre-training a task-agnostic model using large-scale general datasets and then transferring its learning feature representations to downstream tasks is a paradigm in many computer vision applications. While in the last decade, the convolutional-based models that are optimized on the ImageNet (Deng et al. 2009) dataset with a supervised style dominated this field. Owing to the dramatically increasing computational capacity, now we can train models that have several magnitude more model parameters and FLOPs on various image and even video datasets in either supervised (Sun et al. 2017) or self-supervised (He et al. 2020; Huang et al. 2021; Fang et al. 2022) style. Recently, contrastive-based vision-language pre-training (Radford et al. 2021) manifest their superior capabilities in improving downstream tasks performance such as classification (Radford et al. 2021), captioning (Mokady, Hertz, and Bermano 2021), image generation (Ramesh et al. 2021), to name a few. These models are powerful for two reasons: i) the employed large-scale weakly-related datasets provide rich semantics and diverse representations of concepts; ii) the representation vectors of images and texts are roughly aligned in the semantic embedding space. However, the most common approach to using these models is fine-tuning the visual encoder on specific tasks. Although the rich semantics and diverse representations of concepts benefit the downstream tasks, the usage of the textual encoder is still left overlooked.

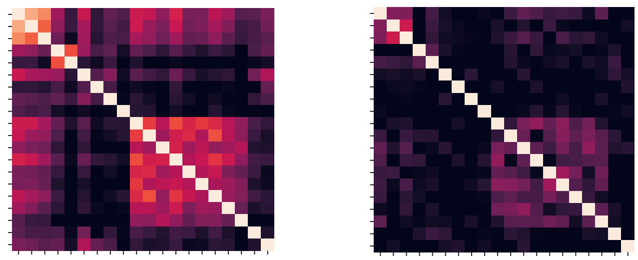

Figure 1: Inter-class correlation maps of "embeddings of class labels" for 20 categories on Kinetics-400. *Left*: The extracted textual vectors of class labels. *Right*: The "embeddings" from learned classifier. The color thresholds are adjusted for a better view. Please zoom in for the best view.

In this study, we aim to improve the transferability of such vision-language pre-training models for downstream classification tasks, with the help of their textual encoders. Our motivation comes from the semantic similarity among the ground-truth labels. To demonstrate this, we employ the Kinetics video recognition dataset (Kay et al. 2017) for the analysis. We extract the embedded textual vectors of class labels using the textual encoder of CLIP. We then calculate the correlation between the embedded textual vectors. The plot is shown on the left of Figure 1. Not surprisingly, the extracted textual vectors of class labels exhibit certain inter-class correlations since part of them include the same verbs in their labels, *e.g.*, *playing <something>*. Meanwhile, the labels with different verbs show a negligible inter-class cor-

[*]Corresponding author.

relation, *e.g.*, *drinking* and *driving*.

Next, we examine the final projection head of a vanilla video recognition framework. We conduct the visual-only fine-tuning progress with the visual encoder that is also released by CLIP (Radford et al. 2021). The detailed configurations are provided in Section 4.3. The projection head is a matrix of $d \times c$ to compute the pre-softmax values (or logits) from the d-dimensional feature vectors for the c classes. Non-rigorously, we can consider the d-dimensional row vectors as the embeddings of the class labels, allowing us to explore the inter-class correlation between these learned "embeddings", as shown on the right side of Figure 1. Interestingly, these learned "embeddings" also reveal certain correlations after the training, despite being initialized randomly and optimized without knowing any textual information [1].

Therefore, we suppose that the semantic information contained in the samples does correlate with inter-classes. Following this motivation, we replace the projection matrix with several variants: i) The projection matrix whose row vectors are randomly sampled (trivial correlation); ii) The projection matrix whose row vectors are orthogonal to each other (non-correlated); iii) The projection matrix that is initialized using the visual statistic knowledge to provide maximized the correlation between labels (see Section 2.2); iv) The projection matrix with fixed embedded textual vectors provides the "proper" correlation. In the empirical studies, we find that textual knowledge significantly improves the transferability of pre-trained models, regarding both the classification accuracy and the convergence speed. Our main contributions are summarized as follows:

- We build a new recognition paradigm to improve the transferability using visual knowledge and textual knowledge from the well-pre-trained vision-language model.
- We conduct extensive experiments on popular video datasets (*i.e.*, Kinetics-400 & 600, UCF-101, HMDB-51 and ActivityNet) to demonstrate the transferability of our solution in many types of transfer learning, *i.e.*, zero-shot / few-shot / general video recognition. Our approach democratizes the training on video datasets and achieves state-of-the-art performance on various video recognition settings, *e.g.*, 87.8% top-1 accuracy on Kinetics-400, and outperforms previous methods by 20$\sim$50% absolute top-1 accuracy under zero-shot, few-shot settings.

2 Methodology

Denotations. In the paper, we use bold letters to denote `Vector`, and capital italic letters to denote `Tensor` or `Matrix`, *e.g.*, we employ $\mathbf{z} \in \mathbb{R}^d$ to denote the feature vector extracted from a pre-trained model of dimension d, we employ $W \in \mathbb{R}^{d \times c}$ to denote the projection matrix for the $c-$class linear classifier. Without ambiguity, we also use capital italic letters to denote the modality in subscripts, especially we employ V and T to denote the *Visual* modality and *Textual* modality, respectively. We further employ lowercase italic letters to denote functions or neural networks. For instance, we employ $g_V(\cdot, \Theta_V)$ and $g_T(\cdot, \Theta_T)$ to denote the visual and textual encoder, respectively. Besides, we employ calligraphic letters, *e.g.*, $\mathcal{D}$, to denote sets of elements.

2.1 Revisiting of Previous Tuning Paradigms

Standard Vision Transferring Paradigm. As shown in Figure 2(a), we start with the most ordinary scenario, where a visual encoder model g_V is optimized using a large-scale dataset $\mathcal{D}$ that contains visual samples with or without ground-truth labels. On our labeled downstream dataset $\tilde{\mathcal{D}} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \ldots\}$, our empirical learning target can be written as

$$g_V^*, W^* = \operatorname*{argmin}_{\Theta_V, W} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\tilde{\mathcal{D}}}\big[H(\boldsymbol{y}|\sigma(W \cdot g_V(\boldsymbol{x})))\big], \quad (1)$$

where $H(\hat{p}|p)$ stands for the `CrossEntropy` between the predicted distribution p and the ground-truth distribution $\hat{p}$, σ denotes the `softmax` operation, $W \in \mathbb{R}^{c \times d}$ denotes the linear projection matrix for classification. The formulation in Eq. 1 is a standard visual feature transferring paradigm, where the visual encoder g_V and the projection matrix (classifier) W are learned simultaneously.

Vision-Language Learning Paradigm. As shown in Figure 2(b), we then review the contrastive learning paradigm of the vision-language models. This paradigm has been widely used for vision-language pre-training *i.e.*, CLIP (Radford et al. 2021), and also been extend to video-text fine-tuning, *i.e.*, ActionCLIP (Wang, Xing, and Liu 2021), CLIP4Clip (Luo et al. 2022). Given a weakly related vision-language pair (*e.g.*, image-text, video-text) dataset $\mathcal{D} = \{(\boldsymbol{x}_{V,1}, \boldsymbol{x}_{T,1}), (\boldsymbol{x}_{V,2}, \boldsymbol{x}_{T,2})...\}$. With slight abuse of the notations, we employ the $\boldsymbol{x}_V, \boldsymbol{x}_T$ to denote a mini-batch of size b, then we minimize the following target,

$$g_V^*, g_T^* = \operatorname*{argmin}_{\Theta_V, \Theta_T} \mathbb{E}_{\boldsymbol{x}_V,\boldsymbol{x}_T\sim\tilde{\mathcal{D}}}\big[H(\mathcal{Q}|\sigma(g_V(\boldsymbol{x}_V)^{\mathrm{T}} \cdot g_T(\boldsymbol{x}_T)))\big], \quad (2)$$

where $\mathcal{Q}$ is the set that contains b one-hot labels of size c, with their $1, 2, \ldots, b$-th element being 1 ($b < c$, denoting the positive vision-language pairs. Here we clarify that, the definition in Eq. 2 is not the rigorous form of the Noise-Contrastive Estimation (NCE) loss proposed in (Van den Oord, Li, and Vinyals 2018). Instead, we employ the cross entropy version implementation in (Radford et al. 2021; Chen, Xie, and He 2021). This implementation depicts a connection between the standard feature transferring paradigm and ours. In which the $g_T(\boldsymbol{x}_T)$ can be considered as the projection matrix that map the visual feature $g_V(\boldsymbol{x}_V)$ to the given label set $\mathcal{Q}$.

2.2 Our Proposed Paradigm

As discussed in Section 1, we replace the learnable randomly initialized linear projection matrix W with pre-defined matrix $\tilde{W}$. Similarly, the training target can be written as

$$g_V^* = \operatorname*{argmin}_{\Theta_V} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\tilde{\mathcal{D}}}\big[H(\boldsymbol{y}|\sigma(\tilde{W} \cdot g_V(\boldsymbol{x})))\big]. \quad (3)$$

Note that $\tilde{W}$ is not in the optimization targets, since we freeze it from updating during the fine-tuning of the downstream tasks. We do this for two reasons: Firstly, it could

[1] That is, optimized with cross-entropy loss with one-hot labels

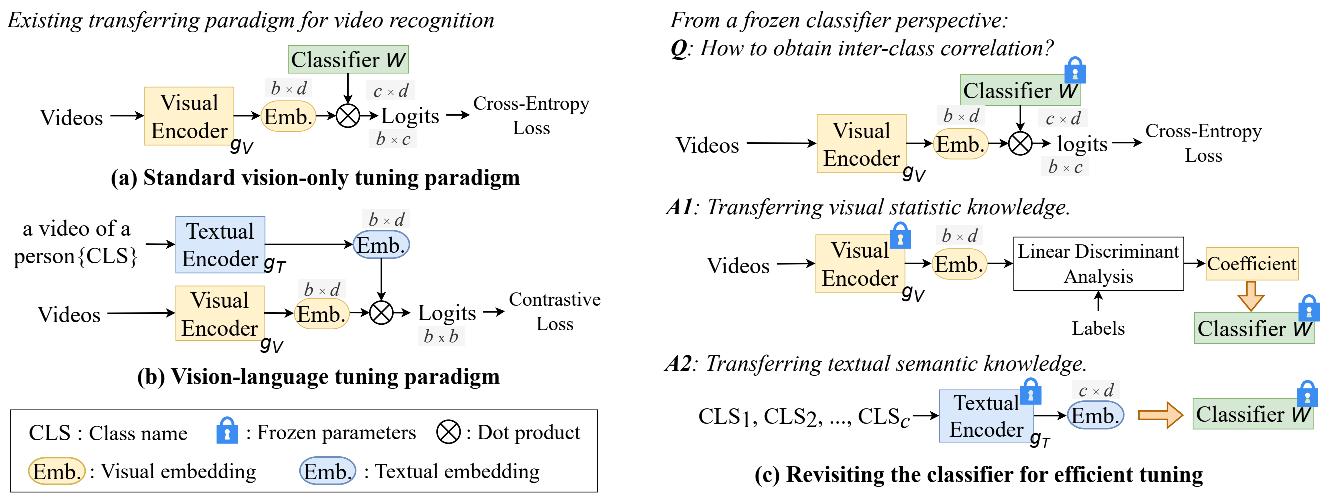

Figure 2: Illustration of transferring vision-language pre-trained models for video recognition. (a) The widely-used standard vision-only tuning paradigm with cross-entropy loss. (b) The vision-language tuning paradigm with contrastive loss. (c) Revisiting the role of the classifier to transfer knowledge from vision-language pre-trained models (*e.g.*, CLIP).

preserve the textual knowledge from being disturbed by the randomness brought by the mini-batch. For instance, when some classes are missing, their embedded feature vector might be broken by the other classes; Secondly, we want to provide a fair comparison between different initialization of $\tilde{W}$. Now we consider how to initialize $\tilde{W}$. To examine how the correlation between the semantic information contained in the samples helps, we investigate the following four types of initialization, which represent different degrees of inter-class correlation.

Randomized Matrix. For the most simple randomized matrix case, we set each row of the $\tilde{W}$ with a random Gaussian vector of zero mean and standard deviation, that is

$$\tilde{W} \sim \mathcal{N}(\mathbf{0}, I_d), \tag{4}$$

where I_d denotes the identity matrix of dimension $d \times d$. Arithmetically, a trivial "correlation" would appear between the row of the $\tilde{W}$, since the sampling size is significantly small to be biased. Evidently, the trivial "correlation" cannot indicate the real correspondence between the classes due to its stochasticity. Therefore we expect the model to have inferior performance since it needs to avoid these incorrect correlations when learning the visual feature representation.

Randomized Orthogonal Matrix. We follow the approach of the randomized matrix. We then remove the correlation by ensuring the row vectors are orthogonal. This is achieved by QR decomposition. Concretely, since $d > c$, we first generate a random matrix of size $d \times d$ and select the first c rows as our projection matrix. Formally, we have,

$$\begin{aligned} \tilde{W}_j &\sim \mathrm{QR}(U)_j, j = 1, 2, \ldots, c, \\ U_i &\sim \mathcal{N}(\mathbf{0}, I_d), i = 1, 2, \ldots, d, \end{aligned} \tag{5}$$

where U is the intermediate randomized matrix, $\mathrm{QR}(U)$ is the row orthogonal matrix obtained through the QR decomposition. Similar to the randomized matrix, we also expect this initialization to have inferior performance. Given the fact that the one-hot label vectors are also orthogonal to each other, it will not be helpful to project the visual feature vectors with an orthogonal matrix, which increases the difficulty of learning meaningful visual features.

Linear Discriminant Projection. We consider another way of initializing the projection matrix. We employ the multi-class Fisher's linear discriminant analysis (LDA) to learn a linear classifier, then employ the weight matrix of the classifier as our initialization of the projection matrix. Specifically, we use the pre-trained visual encoder to extract visual embeddings of samples in the train split, then perform LDA on the pre-extracted visual embeddings of the training set to generate the LDA coefficient. Finally, we use the LDA coefficient to initialize $\tilde{W}$ and freeze it for fine-tuning the visual encoder on the dataset. We compute the LDA projection following previous work (Li, Zhu, and Ogihara 2006). Intuitively, the LDA simultaneously maximizes the inter-class covariance and minimizes intra-class covariance. We, therefore, term this as the maximal correlation initialization using the visual statistic knowledge. As an essential classifier, this type of initialization delivers reasonable performance, but it is largely dependent on the data employed to compute the projection matrix. When the data is limited, the estimated correlation will be biased. On the other hand, in our proposed paradigm, the pre-trained textual encoder provides unbiased correlations for fine-tuning.

Textual Embedding Vectors. We finally describe the paradigm to transfer textual semantic knowledge from a pre-trained textual encoder. Briefly, the projection weight $\tilde{W}$ is composed of the embedded textual feature vectors of the labels. Given a set of tokenized class labels $\mathcal{L} = \{\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_c\}$, we have

$$\tilde{W}_i \sim g_T(\boldsymbol{l}_i), i = 1, 2, \ldots, c, \tag{6}$$

where $\tilde{W}_i$ the i-th row vector in matrix $\tilde{W}$. And $\tilde{W}_i$ is initialized using the textual encoder output of the textual label of the i-th class. In the experimental analysis, we investigate two types of textual feature encoders: i) The encoder that is trained with a visual encoder in the contrastive style, *i.e*., CLIP; ii) The encoder that is trained solely using only textual samples on tasks such as masked language modeling, *i.e*., DistilBERT (Sanh et al. 2019).

3 Related Works

Visual Recognition. Convolutional networks have long been the standard for backbone architectures in image recognition (Krizhevsky, Sutskever, and Hinton 2012; He et al. 2016; Simonyan and Zisserman 2014; Ioffe and Szegedy 2015) and video recognition (Carreira and Zisserman 2017; Qiu, Yao, and Mei 2017; Xie et al. 2018; Tran et al. 2018). Inspired by the Transformer (Vaswani et al. 2017) scaling successes in Natural Language Processing, Vision Transformer (ViT) (Dosovitskiy et al. 2020) applies a standard Transformer directly to images, which delivers impressive performance on image recognition. Since then, ViT (Dosovitskiy et al. 2020) has led a new trend in image recognition backbone architectures, shifting from CNNs to Transformers. To improve performance, follow-up studies, *e.g*., DeiT (Han et al. 2021), Swin (Liu et al. 2021), have been developed. Also, many works has begun to adopt transformers in video recognition, such as TimeSFormer (Bertasius, Wang, and Torresani 2021), ViViT (Arnab et al. 2021), VideoSwin (Liu et al. 2022), and MViT (Fan et al. 2021).

Image-Language Pre-training. Recently, CLIP (Radford et al. 2021) provides good practice in learning the coordinated vision-language pretraining models using the image-text InfoNCE contrastive loss (Van den Oord, Li, and Vinyals 2018). Based on CLIP, several variants (Jia et al. 2021; Li et al. 2022; Yuan et al. 2021; Yu et al. 2022) have been proposed by combining more types of learning tasks such as image-text matching and masked image/language modeling. These contrastively learned models have two deserved properties for downstream tasks: the abundant visual feature representations and the aligned textual feature representations. Yet another study (Yang et al. 2022) merged the downstream classification task into the pretraining progress, which demonstrates a decent improvement of accuracy over the standard cross-entropy loss.

Transferring CLIP Models for Video-Text Learning. Recently, many video-text retrieval methods (Wang, Zhu, and Yang 2021; Zhao et al. 2022; Luo et al. 2022; Wu et al. 2023a) have benefited from vision-language pre-training as well. Moreover, several recent works (Wang, Xing, and Liu 2021; Ju et al. 2022; Wu et al. 2023b) extend the CLIP (Radford et al. 2021) to train a downstream video-text matching model with contrastive loss, then perform video recognition using the similarity between learned video and text embeddings during inference. Instead of these contrastive-based methods, we investigate the correlations of the linear classifier for efficient feature transferring in the standard visual recognition paradigm. Then we directly transfers visual and textual knowledge for video recognition. In comparison to contrastive-based methods, we demonstrate the superiority of our method in efficient training in Table 8. We hope that the simple and effective paradigm can serve as a new baseline for future work.

4 Experiments: Video Recognition

4.1 Setups

To evaluate our method for video recognition, we conduct experiments on five popular datasets, *i.e*., Kinetics-400 (Kay et al. 2017), Kinetics-600 (Carreira et al. 2018), UCF-101 (Soomro, Zamir, and Shah 2012), HMDB-51 (Kuehne et al. 2011) and ActivityNet-v1.3 (Caba Heilbron et al. 2015). *See Supplementary for statistics of these datasets*.

Training & Inference. The video recognition task takes a video as input, and then fed it into a learned encoder to estimate the action category of the video. Given a video, we first uniformly sample T (*e.g*., 8, 16, 32) frames over the entire video. Then we utilize ResNet (He et al. 2016) or ViT (Dosovitskiy et al. 2020) as the video encoders. The classifier in our paradigm is intialized from the textual embedding of the class names and then frozen (fixed), leaving only the parameters in the video encoder to be learned. To trade off accuracy and speed, we consider two inference strategies: (1) *Single View*: We use only 1 clip per video and the center crop for efficient evaluation, (*e.g*., as in Section 4.3). (2) *Multiple Views*: This is a widely used setting in previous works (Feichtenhofer et al. 2019; Carreira and Zisserman 2017) to sample multiple clips per video with several spatial crops in order to get higher accuracy. For comparison with SOTAs, we use four clips with three crops ("4×3 Views") in Table 1. *See Supplementary for training hyperparameters*.

4.2 Main Results

Comparison to State-of-the-Arts. In Table 1, on the challenging **Kinetics-400** dataset, we compare to state-of-the-arts that are pre-trained on large-scale datasets such as ImageNet-21K (Deng et al. 2009), IG-65M (Ghadiyaram, Tran, and Mahajan 2019), JFT-300M (Sun et al. 2017), FLD-900M (Yuan et al. 2021) and JFT-3B (Zhai et al. 2022). Up to now, none of the three largest datasets (*i.e*., JFT-300M, FLD-900M, JFT-3B) is open-sourced and also does not provide pre-trained models. Thus, we use the CLIP (Radford et al. 2021) checkpoints, which are publicly available[2] and have been trained on 400 million web image-text pairs (namely WIT-400M). We can observe that our model outperforms all JFT-pretrained methods in terms of Top-1 and Top-5 accuracy. We achieve an accuracy of 87.8% , which improves even further by 1.3% over Florence (Yuan et al. 2021), although their model and data scale are both 2× larger than ours. Besides, our model is even better than CoVeR (Zhang et al. 2021), and their data scale is 7.5× larger.

To verify the generalization ability of our method, we further evaluate the performance of our method on the well-known untrimmed video benchmark, **ActivityNet-v1.3**. We finetuned the Kinetics-400 pre-trained models with 16 frames on the Activitynet-v1.3 dataset and report the top-1 accuracy and mean average precision (mAP) following the

[2]https://github.com/openai/CLIP/blob/main/clip/clip.py

Method	Input	Pre-train	Top-1	Top-5	FLOPs×Views	Param
NL I3D-101 (Wang et al. 2018)	128×224^2	IN-1K	77.7	93.3	359×10×3	61.8
MVFNet$_{En}$ (Wu et al. 2021a)	24×224^2	IN-1K	79.1	93.8	188×10×3	-
SlowFast NL101 (Feichtenhofer et al. 2019)	16×224^2	Scratch	79.8	93.9	234×10×3	59.9
X3D-XXL (Feichtenhofer 2020)	16×440^2	Scratch	80.4	94.6	144×10×3	20.3
Methods with large-scale pre-training						
TimeSformer-L (Bertasius, Wang, and Torresani 2021)	96×224^2	IN-21K	80.7	94.7	2380×1×3	121.4
ViViT-L/16×2 (Arnab et al. 2021)	32×320^2	IN-21K	81.3	94.7	3992×4×3	310.8
VideoSwin-L (Liu et al. 2022)	32×384^2	IN-21K	84.9	96.7	2107×10×5	200.0
ip-CSN-152 (Tran et al. 2019)	32×224^2	IG-65M	82.5	95.3	109×10×3	32.8
ViViT-L/16×2 (Arnab et al. 2021)	32×320^2	JFT-300M	83.5	95.5	3992×4×3	310.8
TokLearner-L/10 (Ryoo et al. 2021)	32×224^2	JFT-300M	85.4	96.3	4076×4×3	450
MTV-H (Yan et al. 2022)	32×224^2	JFT-300M	85.8	96.6	3706×4×3	-
CoVeR (Zhang et al. 2021)	16×448^2	JFT-300M	86.3	-	-×1×3	-
Florence (Yuan et al. 2021)	32×384^2	FLD-900M	86.5	97.3	-×4×3	647
CoVeR (Zhang et al. 2021)	16×448^2	JFT-3B	87.2	-	-×1×3	-
VideoPrompt (Ju et al. 2022)	16×224^2	WIT-400M	76.9	93.5	-	-
ActionCLIP (Wang, Xing, and Liu 2021)	32×224^2	WIT-400M	83.8	96.2	563×10×3	141.7
Ours ViT-L/14	32×224^2	WIT-400M	87.1	97.4	1662×4×3	230.7
Ours ViT-L/14	32×336^2	WIT-400M	**87.8**	**97.6**	3829×1×3	230.7

Table 1: Comparisons with SOTAs on Kinetics-400. "Views" indicates # temporal clip × # spatial crop. The magnitudes are Giga (10^9) and Mega (10^6) for FLOPs and Param. "IN" denotes ImageNet.

Method	Top-1	mAP
ListenToLook (Gao et al. 2020)	-	89.9
MARL (Wu et al. 2019)	85.7	90.1
DSANet (Wu et al. 2021b)	-	90.5
TSQNet (Xia et al. 2022a)	88.7	93.7
NSNet (Xia et al. 2022b)	90.2	94.3
Ours ViT-L	**92.9**	**96.5**
Ours ViT-L (336↑)	**93.3**	**96.9**

Table 2: Comparisons with SOTAs on ActivityNet.

Method	shot	HMDB	UCF	ANet	K400
VideoSwin (Liu et al. 2022)	2	20.9	53.3	-	-
VideoPrompt (Ju et al. 2022)	5	56.6	79.5	-	58.5
X-Florence (Ni et al. 2022)	2	51.6	84.0	-	-
Ours ViT-L	0	53.8	71.9	75.6	61.0
	1	**72.7**	**96.4**	**89.0**	**75.8**
	2	**73.5**	**96.6**	**90.3**	**78.2**
	All	80.1	96.9	91.1	84.7

Table 3: Comparisons with SOTAs on few-shot recognition.

official evaluation metrics. As shown in Table 2, our method outperforms recent SOTAs with a clear margin. To the best of our knowledge, our method achieves the best performance (96.9%) on ActivityNet. We also evaluate our method on the **UCF-101** and **HMDB-51** datasets to demonstrate its capacity to generalize to smaller data. We achieve the mean class accuracy of 98.2% on UCF and 81.3% on HMDB, respectively. *Please see supplementary for more comparisons on UCF-101 and HMDB-51.*

Few-Shot Video Recognition. Video recognition using only a few samples is known as few-shot video recognition. We study a more challenging K-shot C-way situation instead of the conventional 5-shot 5-way configuration. We scale the task up to categorize **all** categories in the dataset with just K samples per category for training. The lower and upper bound of this situation are denoted by the term "Zero-shot" and "All-shot" respectively. Table 3 reports the Top-1 accuracy for the four datasets. In this extreme scenario of few data, we use CLIP-pretrained ViT-L/14 with 8 frames and TAP for few-shot video recognition. In these extremely data-poor situations (*e.g.*, even with just one shot), we can see that our method offers amazing transferability to diverse domain data. Our approach, in contrast, demonstrates robustness by outperforming SOTAs by a large margin. For instance, when comparing accuracy on HMDB-51 with 2-shot, our method outperforms Swin, X-Florence by **+52.6%** and **+21.9%** respectively. *See Supplementary for training details.*

Zero-Shot Video Recognition. Furthermore, we conduct experiments in the open-set setting. We use our Kinetics-400 pre-trained models (*i.e.*, ViT-L with 8 frames) to perform the zero-shot evaluation on four other video datasets. On UCF-101, HMDB-51 and ActivityNet, there are two major evaluation protocols following (Brattoli et al. 2020): half classes evaluation and full classes evaluation. *Please see Supplementary for the details of two evaluation protocols and the Kinetics-600 evaluation.* We present comprehensive comparisons on four datasets in Table 4, our method shows a strong cross-dataset generalization ability. Our method shows a large improvement upon previous zero-shot recog-

Method	UCF* / UCF	HMDB* / HMDB	ANet*/ ANet	Kinetics-600
GA (Mishra et al. 2018)	17.3±1.1 / -	19.3±2.1 / -	-	-
TS-GCN (Gao, Zhang, and Xu 2019)	34.2±3.1 / -	23.2±3.0 / -	-	-
E2E (Brattoli et al. 2020)	44.1 / 35.3	29.8 / 24.8	26.6 / 20.0	-
DASZL (Kim et al. 2021)	48.9±5.8 / -	- / -	-	-
ER (Chen and Huang 2021)	51.8±2.9 / -	35.3±4.6 / -	-	42.1±1.4
ResT (Lin et al. 2022)	58.7±3.3 / 46.7	41.1±3.7 / 34.4	32.5 / 26.3	-
Ours	**85.8±3.3 / 79.6**	**58.1±5.7 / 49.8**	**84.6±1.4 / 77.4**	**68.9±1.0**

Table 4: Comparisons with SOTAs on zero-shot video recognition. We directly evaluate our method without any additional training on cross-dataset video recognition. ANet is in short for ActivityNet. * means half classes evaluation.

nition methods (**+27.1%** on UCF-101, **+17.0%** on HMDB-51, **+52.1%** on ActivityNet, **+26.8%** on Kinetics-600).

4.3 Ablations on Kinetics

In this section, we conduct extensive ablation experiments on the Kinetics-400 dataset. Unless specified otherwise, we use ViT-B/16 with 8 frames as the video backbone and a single view for testing. The default settings are marked in gray . *See Supplementary for more ablations.*

Different Initializations to the Offline Classifier. We set different initializations described in Section 2.2 to the offline classifier $W \in \mathbb{R}^{d\times c}$ and then train our visual encoder on Kinetics-400. Table 5 lists their comparisons. We show that feeding the offline classifier a random d-by-c matrix with a normal distribution reduces performance significantly. Then we assign the orthogonal matrix to the classifier, and see that removing the inter-class correlation of the classifier will result in inferior performance. Furthermore, we term the linear discriminate projection as the maximal correlation initialization. To do so, we first sample 60 videos from each class in the training set and utilize the pre-trained visual encoder to extract visual embeddings from these 24,000 videos. Finally, we learn the linear classifier by performing linear discriminant analysis on these visual embeddings and their labels. We can see the LDA projection achieves a strong baseline.

Finally, we study the textual embeddings from different textual encoders. We choose DistilBERT (Sanh et al. 2019) and CLIP (Radford et al. 2021) as the textual encoder to pre-extract the text embeddings of c categories. We observe that DistilBERT performs the same performance as CLIP's textual encoder. This may be because both DistillBERT and CLIP are pre-trained with large-scale data, so they both have strong language modeling capabilities and can generate good semantic targets. Although the good semantic targets generated by DistillBERT are not aligned with the visual features of CLIP, it is easy to fit them with trainable visual encoders. We also observe that the loss of DistillBERT will be higher than CLIP in the early stage, but it will quickly decrease to the same level. *More visualizations of these classifiers are in Supplementary.*

Comparison with Vision-Only Tuning Paradigm. As a comparison with our method, we train the unimodality video model, which consists of the same visual encoder and a learnable classifier with random initialization. To produce video embedding, we just apply temporal average pooling (TAP) to frame embeddings. As shown in Table 6, our *Vision-Text* method leads to obvious improvement with the same training recipe, especially in the data-poor situation.

Offline classifier from	Top 1
Random normal matrix	59.3
Random orthogonal matrix	59.4
Linear discriminant projection	80.8
DistilBERT	81.4
Textual encoder of CLIP	**81.5**

Table 5: Exploration of different frozen classifiers.

	Zero-shot	2-shot	Full-shot
Vision-Only	0.2	21.6	75.3
Vision-Text	**54.2**	**65.3**	**80.1**

Table 6: Comparisons with vision-only framework.

Temporal Modeling. Here we explore more temporal modelings for ViT and ResNet: (1) **TAP**: Temporal average pooling is the most straightforward temporal modeling. (2) **T1D**: The channel-wise temporal 1D convolutions, is a common strategy (Wu et al. 2021a; Wang et al. 2021; Liu et al. 2020), to perform efficient temporal interaction in the latter stages (*i.e.*, res_{4-5}) of ResNet. (3) **T-Trans**: The embeddings of frames are fed to a multi-layer (*e.g.*, 6-layer) temporal transformer encoder. (4) **TokenT1D**: We use T1D to model temporal relations for [class] token features that are aggregated from local features via attention in the vision transformer. We perform the TokenT1D in multiple positions of a vision transformer. Results are shown in Table 7. On both backbones, TAP provides simple baselines and T-Trans exhibits the best top-1 accuracy. Both of them maintain the original frame-level representations and then perform temporal modeling. An interesting thing we observed is that T1D does not seem to work in this scenario. The reason lies in that T1D may have the potential to break the learned strong representations provided by CLIP. TokenT1D is another internal-backbone temporal modeling, and it does

Backbone	Modeling	Top-1	Top-5
ResNet-50	TAP	71.2	90.4
	T1D	67.2	88.5
	T-Trans	**74.3**	**91.7**
VIT-B/16	TAP	80.1	95.0
	TokenT1D	80.4	95.0
	T-Trans	**81.5**	**95.5**

Table 7: Temporal modeling for video encoders.

not yield a performance drop, and even slightly improves the TAP baseline. We believe this is because TokenT1D is only imposed on the global [class] token instead of patch tokens, resulting in minimal modifications on pre-trained features.

Ours *v.s.* Contrastive-Based Paradigm. We make a comparison with the Contrastive-based tuning method *i.e*., ActionClip (Wang, Xing, and Liu 2021) mentioned in Section 3. This paradigm treats the recognition task as a video-text matching problem with contrastive loss, thus requiring a batch gathering to collect embeddings of all batches across all GPUs and calculate cosine similarity for a given batch across all other batches. In Table 8, we compare it with the Contrastive-based paradigm and observe that it does not work well without batch gathering. This is due to contrastive learning favors a large batch size (*e.g.,* CLIP used 256 GPUs with a batch size of 128 per GPU to maintain a large 32768×32768 similarity matrix). Besides, involving batch gather will multiply the training time. Also, in this case, the pre-trained textual encoder still needs to be updated, which requires larger GPU memory. However, our paradigm employs pre-extracted text embeddings as our classifier, so the only learned part is the visual encoder. Results show that our method achieves the best accuracy-cost trade-off. Specifically, our method achieves the performance of 81.5% with ViT-B/16, which takes only 10 hours to run the training using 8 GPUs (**2**× **faster** than the matching counterpart). *See Supplementary for details about the batch gathering.*

Paradigm	Batch Gather	Textual Encoder	Top-1	V100-days
Contrastive-Based	✓	online	81.2	6.7 (10*)
	✓	offline	80.7	6.6
	✗	online	77.8	3.5
	✗	offline	76.1	3.3
Ours	✗	offline	**81.5**	**3.3**

Table 8: Ours *vs*. Contrastive-based paradigm with ViT-B/16 on Kinetics-400. The number of V100 days is the number of V100 GPU used for training multiplied by the training time in days. ∗ indicates the official result (Wang, Xing, and Liu 2021) via "Data-parallel training" on 3090 GPUs. For efficient training and fair comparison, we implement all experiments with "Distributed Data-parallel training" in this Table.

Views	Top-1	GFLOPs
Single→Multiple	81.5→**82.9**	**90.3**→90.3×12

Table 9: Two classic evaluation protocols.

Method	Top-1	FLOPs	Params	Throughput
ViViT-L/16-320	81.3	3992G	310.8M	4.2 vid/s*
Ours ViT-B/32	78.5	23.7G	71.6M	322.5 vid/s
Ours ViT-B/16	81.5	90.3G	69.9M	126.5 vid/s
Ours ViT-L/14	85.4	415.4G	230.4M	35.5 vid/s

Table 10: Analysis on throughput. "vid/s" represents the average number of videos per second. The larger "vid/s" represents higher efficiency. * is the official result with TPU-v3.

More Instantiations. Table 10 presents the results of our method using different visual encoders, indicating that deeper backbones can achieve better performance. Table 9 presents the results of our method under two evaluation protocols mentioned in Section 4.1, where the multi-view evaluation protocol results in additional improvements.

Analysis on Efficiency. In Table 10, we present the computational cost and efficiency of our models. We follow the common inference settings by using a single NVIDIA A100 GPU to measure the throughput. We use a batch size of 16 to measure the throughput. Our models achieve the **29**× **faster** throughput and **44**× **fewer** FLOPs compared with the previous transformer-based method ViViT (Arnab et al. 2021) under the same accuracy.

5 Limitation and Conclusion

Limitation: The performance of the proposed paradigm is restricted to how the category labels are represented. For instance, in tasks such as human re-identification, the labels are often set as numerical values such as 0, 1, 2, etc. In this case, we cannot transfer any semantic information from the textual encoders, while transferring visual statistic knowledge (*i.e*., LDA classifier) could be helpful.

Conclusion: We present a new paradigm for improving the transferability of visual recognition that is based on the knowledge from the textual encoder of the well-trained vision-language model. The empirical study shows that our method improves both the performance and the convergence speed of visual classification. The proposed approach has superior performance on both general and zero-shot/few-shot recognition and achieves state-of-the-art performance on video recognition tasks, and democratizes transferring on challenging video datasets, *i.e*., Kinetics-400.

Acknowledgments

This paper is supported by the Australian Research Council Grant DP200103223, Australian Medical Research Future Fund MRFAI000085, CRC-P Smart Material Recovery Facility (SMRF) – Curby Soft Plastics, and CRC-P ARIA - Bionic Visual-Spatial Prosthesis for the Blind.

References

Arnab, A.; Dehghani, M.; Heigold, G.; Sun, C.; Lučić, M.; and Schmid, C. 2021. Vivit: A video vision transformer. In *ICCV*, 6836–6846.

Bertasius, G.; Wang, H.; and Torresani, L. 2021. Is Space-Time Attention All You Need for Video Understanding? In *ICML*, 813–824. PMLR.

Brattoli, B.; Tighe, J.; Zhdanov, F.; Perona, P.; and Chalupka, K. 2020. Rethinking zero-shot video classification: End-to-end training for realistic applications. In *CVPR*, 4613–4623.

Caba Heilbron, F.; Escorcia, V.; Ghanem, B.; and Carlos Niebles, J. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 961–970.

Carreira, J.; Noland, E.; Banki-Horvath, A.; Hillier, C.; and Zisserman, A. 2018. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*.

Carreira, J.; and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*.

Chen, S.; and Huang, D. 2021. Elaborative rehearsal for zero-shot action recognition. In *ICCV*, 13638–13647.

Chen, X.; Xie, S.; and He, K. 2021. An empirical study of training self-supervised vision transformers. In *ICCV*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*, 248–255.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fan, H.; Xiong, B.; Mangalam, K.; Li, Y.; Yan, Z.; Malik, J.; and Feichtenhofer, C. 2021. Multiscale vision transformers. In *ICCV*, 6824–6835.

Fang, B.; Wu, W.; Liu, C.; Zhou, Y.; He, D.; and Wang, W. 2022. Mamico: Macro-to-micro semantic correspondence for self-supervised video representation learning. In *ACM MM*, 1348–1357.

Feichtenhofer, C. 2020. X3D: Expanding Architectures for Efficient Video Recognition. In *CVPR*, 203–213.

Feichtenhofer, C.; Fan, H.; Malik, J.; and He, K. 2019. Slowfast networks for video recognition. In *ICCV*, 6202–6211.

Gao, J.; Zhang, T.; and Xu, C. 2019. I know the relationships: Zero-shot action recognition via two-stream graph convolutional networks and knowledge graphs. In *AAAI*, volume 33, 8303–8311.

Gao, R.; Oh, T.-H.; Grauman, K.; and Torresani, L. 2020. Listen to look: Action recognition by previewing audio. In *CVPR*, 10457–10467.

Ghadiyaram, D.; Tran, D.; and Mahajan, D. 2019. Large-scale weakly-supervised pre-training for video action recognition. In *CVPR*, 12046–12055.

Han, K.; Xiao, A.; Wu, E.; Guo, J.; Xu, C.; and Wang, Y. 2021. Transformer in transformer. *NeurIPS*, 15908–15919.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 9729–9738.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Huang, D.; Wu, W.; Hu, W.; Liu, X.; He, D.; Wu, Z.; Wu, X.; Tan, M.; and Ding, E. 2021. Ascnet: Self-supervised video representation learning with appearance-speed consistency. In *ICCV*, 8096–8105.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 448–456. PMLR.

Jia, C.; Yang, Y.; Xia, Y.; Chen, Y.-T.; Parekh, Z.; Pham, H.; Le, Q.; Sung, Y.-H.; Li, Z.; and Duerig, T. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 4904–4916. PMLR.

Ju, C.; Han, T.; Zheng, K.; Zhang, Y.; and Xie, W. 2022. Prompting visual-language models for efficient video understanding. In *ECCV*, 105–124. Springer.

Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Kim, T. S.; Jones, J.; Peven, M.; Xiao, Z.; Bai, J.; Zhang, Y.; Qiu, W.; Yuille, A.; and Hager, G. D. 2021. Daszl: Dynamic action signatures for zero-shot learning. In *AAAI*, volume 35, 1817–1826.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *NeurIPS*.

Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; and Serre, T. 2011. HMDB: a large video database for human motion recognition. In *ICCV*, 2556–2563.

Li, J.; Li, D.; Xiong, C.; and Hoi, S. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*.

Li, T.; Zhu, S.; and Ogihara, M. 2006. Using discriminant analysis for multi-class classification: an experimental investigation. *Knowledge and information systems*, 10(4).

Lin, C.-C.; Lin, K.; Wang, L.; Liu, Z.; and Li, L. 2022. Cross-modal Representation Learning for Zero-shot Action Recognition. In *CVPR*, 19978–19988.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 10012–10022.

Liu, Z.; Luo, D.; Wang, Y.; Wang, L.; Tai, Y.; Wang, C.; Li, J.; Huang, F.; and Lu, T. 2020. TEINet: Towards an Efficient Architecture for Video Recognition. In *AAAI*, 11669–11676.

Liu, Z.; Ning, J.; Cao, Y.; Wei, Y.; Zhang, Z.; Lin, S.; and Hu, H. 2022. Video swin transformer. In *CVPR*, 3202–3211.

Luo, H.; Ji, L.; Zhong, M.; Chen, Y.; Lei, W.; Duan, N.; and Li, T. 2022. CLIP4Clip: An empirical study of CLIP for end to end video clip retrieval and captioning. *Neurocomputing*, 508: 293–304.

Mishra, A.; Verma, V. K.; Reddy, M. S. K.; Arulkumar, S.; Rai, P.; and Mittal, A. 2018. A generative approach to zero-shot and few-shot action recognition. In *WACV*, 372–380.

Mokady, R.; Hertz, A.; and Bermano, A. H. 2021. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*.

Ni, B.; Peng, H.; Chen, M.; Zhang, S.; Meng, G.; Fu, J.; Xiang, S.; and Ling, H. 2022. Expanding language-image pretrained models for general video recognition. In *ECCV*.

Qiu, Z.; Yao, T.; and Mei, T. 2017. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*, 8748–8763. PMLR.

Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-shot text-to-image generation. In *ICML*, 8821–8831. PMLR.

Ryoo, M. S.; Piergiovanni, A.; Arnab, A.; Dehghani, M.; and Angelova, A. 2021. TokenLearner: What Can 8 Learned Tokens Do for Images and Videos? *arXiv preprint arXiv:2106.11297*.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Sun, C.; Shrivastava, A.; Singh, S.; and Gupta, A. 2017. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 843–852.

Tran, D.; Wang, H.; Torresani, L.; and Feiszli, M. 2019. Video classification with channel-separated convolutional networks. In *ICCV*, 5552–5561.

Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; and Paluri, M. 2018. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *CVPR*, 6450–6459.

Van den Oord, A.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv e-prints*, arXiv–1807.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*, 5998–6008.

Wang, L.; Tong, Z.; Ji, B.; and Wu, G. 2021. TDN: Temporal difference networks for efficient action recognition. In *CVPR*, 1895–1904.

Wang, M.; Xing, J.; and Liu, Y. 2021. Actionclip: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-Local Neural Networks. In *CVPR*, 7794–7803.

Wang, X.; Zhu, L.; and Yang, Y. 2021. T2vlad: global-local sequence alignment for text-video retrieval. In *CVPR*.

Wu, W.; He, D.; Lin, T.; Li, F.; Gan, C.; and Ding, E. 2021a. MVFNet: Multi-View Fusion Network for Efficient Video Recognition. In *AAAI*, volume 35, 2943–2951.

Wu, W.; He, D.; Tan, X.; Chen, S.; and Wen, S. 2019. Multi-Agent Reinforcement Learning Based Frame Sampling for Effective Untrimmed Video Recognition. In *ICCV*, 6222–6231.

Wu, W.; Luo, H.; Fang, B.; Wang, J.; and Ouyang, W. 2023a. Cap4Video: What Can Auxiliary Captions Do for Text-Video Retrieval? In *CVPR*.

Wu, W.; Wang, X.; Luo, H.; Wang, J.; Yang, Y.; and Ouyang, W. 2023b. Bidirectional Cross-Modal Knowledge Exploration for Video Recognition with Pre-trained Vision-Language Models. In *CVPR*.

Wu, W.; Zhao, Y.; Xu, Y.; Tan, X.; He, D.; Zou, Z.; Ye, J.; Li, Y.; Yao, M.; Dong, Z.; et al. 2021b. DSANet: Dynamic Segment Aggregation Network for Video-Level Representation Learning. *ACM MM*, 1903–1911.

Xia, B.; Wang, Z.; Wu, W.; Wang, H.; and Han, J. 2022a. Temporal Saliency Query Network for Efficient Video Recognition. In *ECCV*, 741–759.

Xia, B.; Wu, W.; Wang, H.; Su, R.; He, D.; Yang, H.; Fan, X.; and Ouyang, W. 2022b. NSNet: Non-saliency Suppression Sampler for Efficient Video Recognition. In *ECCV*, 705–723.

Xie, S.; Sun, C.; Huang, J.; Tu, Z.; and Murphy, K. 2018. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 305–321.

Yan, S.; Xiong, X.; Arnab, A.; Lu, Z.; Zhang, M.; Sun, C.; and Schmid, C. 2022. Multiview transformers for video recognition. In *CVPR*, 3333–3343.

Yang, J.; Li, C.; Zhang, P.; Xiao, B.; Liu, C.; Yuan, L.; and Gao, J. 2022. Unified contrastive learning in image-text-label space. In *CVPR*, 19163–19173.

Yu, J.; Wang, Z.; Vasudevan, V.; Yeung, L.; Seyedhosseini, M.; and Wu, Y. 2022. CoCa: Contrastive Captioners are Image-Text Foundation Models. *arXiv preprint arXiv:2205.01917*.

Yuan, L.; Chen, D.; Chen, Y.-L.; Codella, N.; Dai, X.; Gao, J.; Hu, H.; Huang, X.; Li, B.; Li, C.; et al. 2021. Florence: A New Foundation Model for Computer Vision. *arXiv preprint arXiv:2111.11432*.

Zhai, X.; Kolesnikov, A.; Houlsby, N.; and Beyer, L. 2022. Scaling vision transformers. In *CVPR*, 12104–12113.

Zhang, B.; Yu, J.; Fifty, C.; Han, W.; Dai, A. M.; Pang, R.; and Sha, F. 2021. Co-training Transformer with Videos and Images Improves Action Recognition. *arXiv preprint arXiv:2112.07175*.

Zhao, S.; Zhu, L.; Wang, X.; and Yang, Y. 2022. Center-CLIP: Token Clustering for Efficient Text-Video Retrieval. *SIRIR*.