International Journal of Computer Vision (2024) 132:392–409
https://doi.org/10.1007/s11263-023-01876-w

Transferring Vision-Language Models for Visual Recognition: A Classifier Perspective

Wenhao Wu[1] · **Zhun Sun**[2] · **Yuxin Song**[2] · **Jingdong Wang**[2] · **Wanli Ouyang**[3]

Received: 28 February 2023 / Accepted: 7 August 2023 / Published online: 7 September 2023

Abstract
Transferring knowledge from pre-trained deep models for downstream tasks, particularly with limited labeled samples, is a fundamental problem in computer vision research. Recent advances in large-scale, task-agnostic vision-language pre-trained models, which are learned with billions of samples, have shed new light on this problem. In this study, we investigate how to efficiently transfer aligned visual and textual knowledge for downstream visual recognition tasks. We first revisit the role of the linear classifier in the vanilla transfer learning framework, and then propose a new paradigm where the parameters of the classifier are initialized with semantic targets from the textual encoder and remain fixed during optimization. To provide a comparison, we also initialize the classifier with knowledge from various resources. In the empirical study, we demonstrate that our paradigm improves the performance and training speed of transfer learning tasks. With only minor modifications, our approach proves effective across 17 visual datasets that span three different data domains: image, video, and 3D point cloud.

Keywords Visual recognition · Large vision model · Transfer learning

1 Introduction

In the field of optimizing neural network training efficiency, knowledge transfer aims to provide pre-learned information to downstream tasks. For visual recognition tasks, the approach typically involves leveraging feature representations derived from a task-agnostic model optimized with large-scale universal datasets, followed by building a *classifier* on the top of the model. Former studies put more emphasis on learning the base model. Over the last decade, for example, the dominant approach involved training models on the ImageNet (Deng et al., 2009) dataset and subsequently transferring them to downstream tasks. Owing to the dramatically increasing computational capacity, general-proposed pre-trained models with several magnitudes more parameters and FLOPs have been successfully trained in both full-/semi-supervised (Sun et al., 2017) and self-supervised (He et al., 2020, 2022) style. Recently, contrastive vision-language models (Radford et al., 2021; Jia et al., 2021a; Yuan et al., 2021) have garnered increasing interest as pre-training models in transfer learning due to their superior capabilities and effectiveness for visual recognition tasks. These models, which benefit from the knowledge of the language modality, have shown improved performance on various visual tasks, such as zero-shot classification (Radford et al., 2021), captioning (Mokady et al., 2021), and image generation (Ramesh et al., 2021), to name a few.

In this study, we aim to enhance the transferability of vision-language pre-training models for downstream visual recognition tasks by revisiting the knowledge-transferring progress from the perspective of the classifier. Specifically, we examine the properties of the pre-training models, and propose a simple yet effective paradigm to enhance

Communicated by Kaiyang Zhou.

✉ Wenhao Wu
wenhao.wu@sydney.edu.au

Zhun Sun
sunzhun@baidu.com

Yuxin Song
songyuxin02@baidu.com

Jingdong Wang
wangjingdong@baidu.com

Wanli Ouyang
ouyangwanli@pjlab.org.cn

[1] The University of Sydney, Darlington, Australia

[2] Department of Computer Vision Technology, Baidu Inc., Beijing, China

[3] Shanghai Artificial Intelligence Laboratory, Shanghai, China

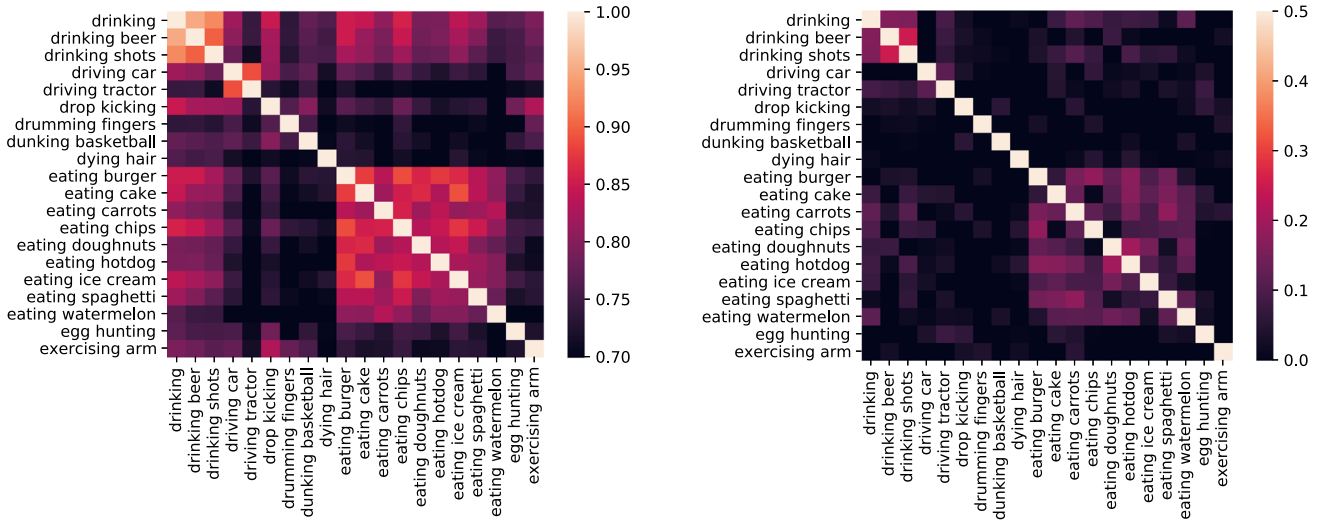

Fig. 1 Inter-class correlation maps of "embeddings of class labels" for 20 categories on Kinetics-400. Left: The extracted textual vectors of class labels, Right: The "embeddings" from learned classifier. The color thresholds are adjusted for a better view. Please zoom in for the best view

their transferability. Our findings demonstrate that these pre-training models hold three essential properties for our paradigm: (i) **Semantic-rich representations**, which are obtained by training the models with extensive weakly-related image-text sample pairs using large neural network architectures. In contrast to supervised-style models learned on standard image-label datasets, the semantic-rich representations are expected to contain more semantics and diverse representations of concepts, which is crucial in the unknown target domain settings. (ii) **Modality alignment**, which aligns the representation vectors from a paired sample's visual and textual modality in semantic embedding space. This property provides an advantage in the initialization when the samples for downstream tasks are limited, *i.e.*, in the zero-/few-shot scenarios, compared to the visual-only classifier fine-tuning approach. (iii) **Intra-modality correlations.** The contrastive training algorithm also provides weak intra-modality correlations. That is, the representation vectors of similar images or texts are close to each other (Radford et al., 2021; Sun, 2022). In contrast to the aforementioned properties, intra-modality correlations from samples' influence are often overlooked. Concisely, a classifier with appropriately correlated targets rather than one-hot labels learns faster and performs better.

To demonstrate the importance of appropriately correlated classifier targets, we conduct a toy experiment to depict the intra-modality correlations in two scenarios. We employ the Kinetics video recognition dataset (Kay et al., 2017) for the analysis (The detailed configurations are provided in Sect. 4.3). In the first scenario, we extract the textual embedding vectors of *the name of class labels* using the textual encoder of CLIP (Radford et al., 2021) and then calculate the correlation among the textual embedding vectors. In the second scenario, we examine the final projection head of a vanilla fine-tuning framework. Precisely, we learn a classifier based on the visual encoder from the same CLIP model. The projection head of the classifier is a matrix of $d \times c$ used to compute the pre-softmax logits, from the d-dimensional feature vectors for the c classes. Therefore, we treat the d-dimensional row vectors as the "embeddings" of the class labels. This non-rigorous setting allows us to explore the intra-modality correlation between these learned "embeddings". The results are plotted in Fig. 1. While we could observe clear correlations among the embeddings of category names since some of them contain the same keywords (*e.g.*, *playing <something>*.) Interestingly, in the second scenario, these learned "embeddings" also reveal a similar correlation map after the training, despite being initialized randomly and optimized without knowing any textual information (That is, optimized with the cross-entropy loss with one-hot labels).

In summary, we take full advantage of the large-scale contrastive image-language pre-trained models and build a novel general paradigm for the transfer learning settings. Our main contributions are as follows:

- We revisit the transfer learning pipeline from the perspective of classifiers and spot that properly correlated targets, and pre-aligned semantic knowledge are crucial for downstream visual recognition tasks.
- We build a new paradigm to transfer textual knowledge for visual recognition using contrastively pre-trained vision-language models. Our paradigm accelerates the transfer learning progress while taking full advantage of the pre-trained models.
- Comprehensive experiments are conducted on **17** visual datasets that span three distinct data domains: image,

video, and 3D point cloud. For video recognition, we evaluate our model on 6 well-known video benchmarks, including single-label and multi-label recognition. while also verifying its effectiveness in zero-shot and few-shot scenarios. For image classification, we perform experiments on 10 different image datasets, and the results demonstrate that our method is an effective few-shot learner. For 3D point cloud recognition, we validate our method on the ModelNet40 dataset, and find that it outperforms the vision-only paradigm by a significant margin.
- We open-source our code and models at https://github.com/whwu95/Text4Vis.

2 Related Works

2.1 Visual Recognition Tasks and Transfer Learning

Visual recognition is one of the most important tasks in the design of machine learning systems. From the perspective of the visual backbone, we could roughly divide the evolution of the system into two eras: i) The Convolutional Neural Network (CNN) based architectures for image (Krizhevsky et al., 2012; He et al., 2016; Simonyan & Zisserman, 2014; Ioffe & Szegedy, 2015) or video recognition (Carreira & Zisserman, 2017; Qiu et al., 2017; Xie et al., 2018; Tran et al., 2018; Wu et al., 2021a, b). ii) The Vision Transformer (ViT) based architectures for image (Dosovitskiy et al., 2020; Han et al., 2021; Liu et al., 2021) or video recognition (Bertasius et al., 2021; Arnab et al., 2021; Liu et al., 2022; Fan et al., 2021). As ViT models are challenging to train from scratch without large-scale datasets, transfer learning techniques have regained popularity.

Transfer learning aims to enhance target learners' performance on target domains by transferring knowledge from related but different source domains (Tan et al., 2018; Ribani & Marengoni, 2019; Zhuang et al., 2020), thereby reducing the requirements of target domain data for learning the target model. A typical transfer learning system is built with a pre-trained model trained with source domain data and a classifier for the target domain data. This study discusses a sub-family of transfer learning systems that utilize large-scale task-agnostic models. Related studies on this sub-family are discussed in Sect. 2.3.

2.2 Image-Language Pre-training

The recent success of Contrastive Language-Image Pre-Training (CLIP) (Radford et al., 2021) has paved the way for coordinated vision-language pre-training models utilizing the image-text InfoNCE contrastive loss (Van den Oord et al., 2018). After that, several works have since been proposed that combine various learning tasks, including image-text matching and masked image/language modeling, such as ALIGN (Jia et al., 2021b), BLIP (Li et al., 2022b), Florence (Yuan et al., 2021), and CoCa (Yu et al., 2022). These contrastively learned models exhibit two essential properties for downstream tasks: rich visual feature representations and aligned textual feature representations. Another recent study (Yang et al., 2022) has incorporated the downstream classification task into the pretraining process, resulting in improved accuracy over the standard cross-entropy loss. These developments demonstrate the potential for coordinated pre-training of vision and language models and open up exciting opportunities for further advances in vision-language understanding.

2.3 Transferring CLIP for Downstream Tasks

The transfer of pre-trained CLIP to downstream tasks is a recent and emerging research direction. Several recent studies (Gao et al., 2021; Zhang et al., 2021b; Zhou et al., 2021, 2022) have investigated the efficient transfer of pre-trained CLIP to downstream image recognition tasks. In addition, CLIP has been leveraged to enhance dense prediction tasks such as object detection (Rao et al., 2022) and segmentation (Lüddecke & Ecker, 2022; Li et al., 2022a). In the video domain, CLIP has also benefited many text-video retrieval methods (Zhao et al., 2022; Luo et al., 2021). For video recognition, ActionClip (Wang et al., 2021b) and VideoPrompt (Ju et al., 2022) extend CLIP (Radford et al., 2021) to train a downstream video-text matching model with contrastive loss and utilize the similarity between learned video and text embeddings during inference. Other methods, such as ST-Adapter (Pan et al., 2022) and EVL (Lin et al., 2022b), use only the visual encoder for unimodality transferring without involving textual knowledge. This study investigates the correlation between the linear classifier and efficient feature transfer in the standard visual recognition paradigm. We propose a direct transfer of visual and textual knowledge for visual recognition, without using contrastive-based methods.

3 Methodology

3.1 Denotations

In this paper, we use bold letters to denote `Vector`, while capital italic letters are used to denote `Tensor` or `Matrix`. For example, we use $\mathbf{z} \in \mathbb{R}^d$ to denote the feature vector extracted from a pre-trained model of dimension d, and $W \in \mathbb{R}^{d\times c}$ 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. Specifically, we use V and T to denote the *Visual* modality and the *Textual* modality, respectively. We also use lowercase italic letters to

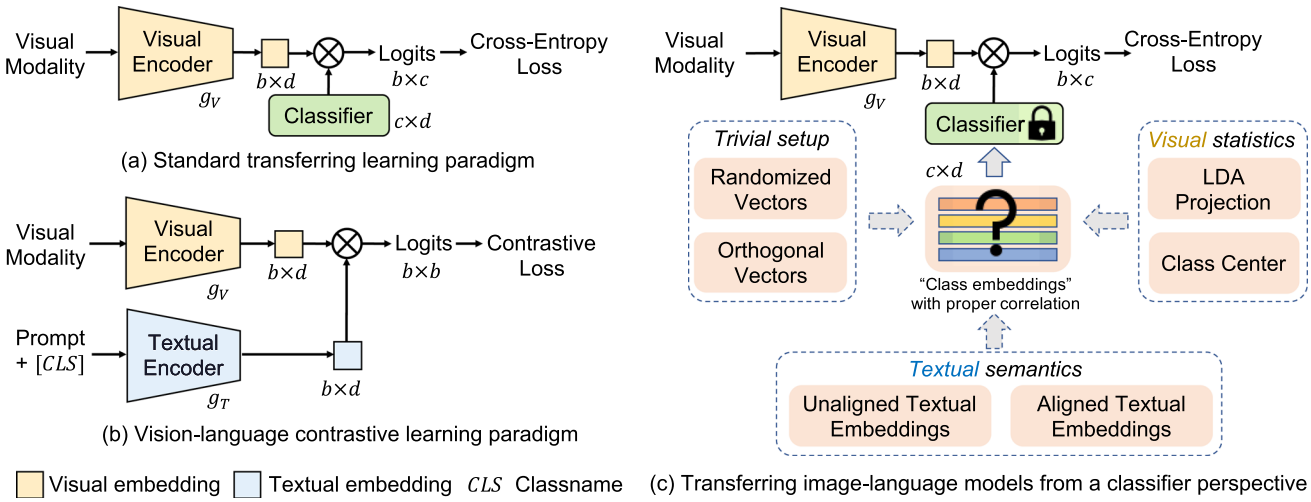

Fig. 2 Illustration of transferring vision-language pre-trained models for visual recognition. **a** The widely-used standard vision-only tuning paradigm with cross-entropy loss. **b** The vision-language contrastive learning paradigm with contrastive loss, *e.g.*, CLIP (Radford et al., 2021), ActionCLIP (Wang et al., 2021b). **c** Revisiting the role of the classifier to transfer knowledge from vision-language pre-trained models. c denotes the number of categories, b is the batch size and d represents the dimension of embeddings

denote functions or neural networks, such as $g_V(\cdot, \Theta_V)$ and $g_T(\cdot, \Theta_T)$, which represent the visual and textual encoders, respectively. Furthermore, we employ calligraphic letters, such as $\mathcal{D}$, to denote sets of elements.

3.2 Revisiting of Existing Learning Paradigms

Standard Transfer Learning Paradigm In Fig. 2a, we depict the conventional scenario, where a visual encoder model g_V is trained on a large-scale dataset $\mathcal{D}$ containing 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 expressed as

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

where $H(\hat{p}|p)$ represents the `CrossEntropy` between the predicted distribution p and the ground-truth distribution $\hat{p}$. The symbol 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 W are learned jointly.

Vision-Language Contrastive Learning Paradigm As shown in Fig. 2b, we then review the contrastive learning paradigm of vision-language models, which has gained widespread use in vision-language pre-training, such as CLIP (Radford et al., 2021), and extended to video-text fine-tuning, *e.g.*, ActionCLIP (Wang et al., 2021b), CLIP4Clip (Luo et al., 2021).

Given a dataset $\mathcal{D} = \{(\boldsymbol{x}_{V,1}, \boldsymbol{x}_{T,1}), (\boldsymbol{x}_{V,2}, \boldsymbol{x}_{T,2}), \cdots\}$, consisting of weakly related vision-language pairs (*e.g.*, image-text, video-text). 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:

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

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 ($b < c$), representing the positive vision-language pairs. We note that the definition in Eq. 2 is not the rigorous form of the Noise-Contrastive Estimation (NCE) loss proposed in Van den Oord et al. (2018). Instead, we employ the cross-entropy version implementation used in Radford et al. (2021); Chen et al. (2021). The contrastive learning paradigm first projects the visual feature $g_V(\boldsymbol{x}_V)$ with a projection matrix $g_T(\boldsymbol{x}_T)$, then follows the standard transfer learning paradigm to match the similarity matrix with the diagonal label set $\mathcal{Q}$.

3.3 Our Proposed Paradigm

As depicted in Fig. 2, we propose a more generalized paradigm by replacing the learnable, randomly initialized linear projection matrix W with a pre-defined matrix $\tilde{W}$, building upon the classifier perspective. Following Sect. 3.2, the training target can be formulated as:

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

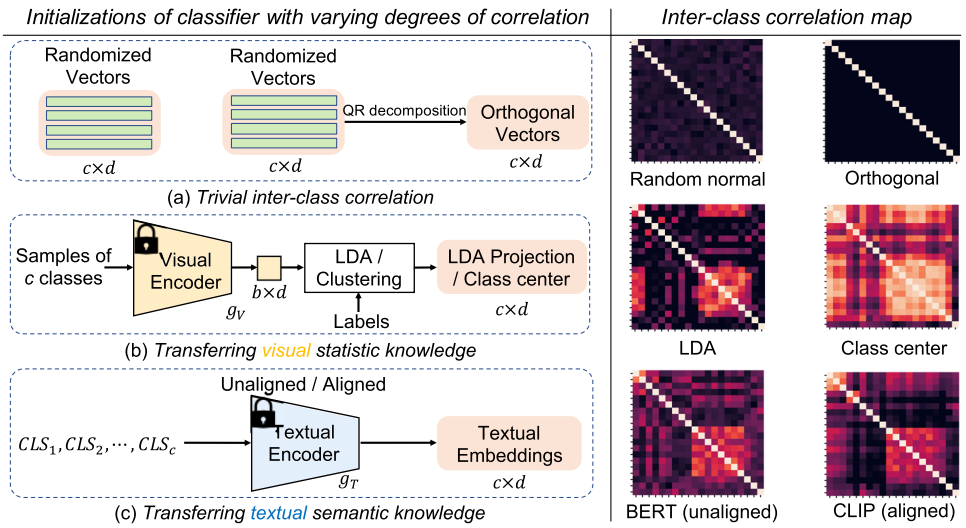

Fig. 3 Illustration of 6 types of projection matrix initialization which develop different levels of correlation between the target embedding vectors. On the Left: **a** trivial or no correlation between the target vector; **b** correlation calculated from the visual statistic; **c** correlation calculated from the textual semantic knowledge. On the Right: Inter-class correlation map obtained from the six types of initialization. Impressively, correlation maps yield a similar appearance from transferring visual statistics and textual semantic knowledge. See Fig. 1 for more details

In the following subsections, we investigate different initialization methods for $\tilde{W}$.

3.4 Discussion on Initialization

To investigate the extent to which the correlation between semantic information contained in the samples is helpful, we examine several types of initialization, which represent different degrees of intra-modality (or inter-class from the perspective of classifier) correlation, as illustrated in Fig. 3.

3.4.1 Trivial Inter-class Correlation

Randomized Matrix We start with the simplest initialization method, which involves setting each row of $\tilde{W}$ to a random Gaussian vector with zero mean and standard deviation. This can be denoted as follows:

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

where I_d denotes the identity matrix of dimension $d \times d$. While this method generates trivial correlations between the rows of $\tilde{W}$ due to its stochasticity, these correlations cannot reflect the actual correspondence between the visual classes. 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 Next, we consider the case where correlations are removed from the projection matrix. We follow the approach of the randomized matrix and then remove the correlation by ensuring that 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, and $\mathrm{QR}(U)$ is the row orthogonal matrix obtained through the QR decomposition. Similar to the randomized matrix, we expect this initialization to have inferior performance. Since 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 may increase the difficulty of learning meaningful visual features.

3.4.2 Correlation from Visual Statistic Knowledge

Class Center Projection To utilize the visual encoder's statistical knowledge, we randomly select a small subset of labeled samples from the training dataset. For our experiments on the Kinetics-400 dataset, we sample 60 videos from each class, which is approximately 10% of the training data. Next, we compute the mean value of each class's visual embeddings extracted from the visual encoder. These mean vectors are treated as the centers for each class and are used to initialize the classifier's parameters. The class center initialization provides a basic approximation of the visual knowledge obtained from the pre-trained model. However, its effectiveness largely depends on the data used to compute the projection matrix, and when the data is limited, the estimated correlation among visual embeddings may be biased.

Linear Discriminant Projection We propose another approach to initializing the projection matrix using visual statistics. We use multi-class Fisher's linear discriminant analysis (LDA) to learn a linear classifier and employ the weight matrix of the classifier as our initialization for the projection matrix.

Specifically, we first use the same visual embeddings as the previous approach to computing the LDA coefficient, following previous work (Li et al., 2006). Then, we use the LDA coefficient to initialize $\tilde{W}$ and freeze it for fine-tuning the visual encoder on the dataset. Intuitively, the LDA simultaneously maximizes the inter-class covariance and minimizes intra-class covariance. Therefore, we term this as the maximal correlation initialization using visual statistic knowledge. However, the linear discriminant projection also suffers from biased data sampling progress.

3.4.3 Correlation from Textual Semantic Knowledge

Textual Embedding Vectors We now describe how we transfer textual semantic knowledge from a pre-trained textual encoder to initialize the projection weight $\tilde{W}$. Given a set of tokenized class labels $\mathcal{L} = \boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_c$, we initialize the i-th row vector in $\tilde{W}$ as follows:

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

where g_T is a function that maps a textual input to an embedded feature vector using a pre-trained textual encoder. In our experiments, we investigate two types of textual feature encoders: i) The encoder that is trained solely using textual samples on tasks such as masked language modeling, *i.e.*, DistilBERT (Sanh et al., 2019); ii) The encoder that is trained with a visual encoder in the contrastive style, *i.e.*, CLIP (Radford et al., 2021). Using the textual embeddings to initialize $\tilde{W}$ allows us to roughly pre-align the visual and textual embeddings in the same embedding space.

3.5 Discussion on Parameter Frozen

It is worth mentioning that, in our paradigms, $\tilde{W}$ is not in the optimization targets. This means we freeze it from updating during the fine-tuning of the downstream tasks. We have the following reasons for this: firstly, since the textual knowledge is extracted by the textual encoder, freezing this part could significantly decrease the computational resources required for fine-tuning. As we showed in Sect. 4, freezing the parameters of $\tilde{W}$ leads to a decrease in the training period. Secondly, freezing the parameter helps to reduce biases brought by the limited semantic knowledge of class names. By keeping the feature embeddings distributed as they were learned on large-scale datasets, we improve the diversity of the representations and the learning stability. Finally, this configuration also compares former studies that employ textual information for vision transfer learning.

4 Experiments: Video Recognition

In this section, we transfer the image-language pre-trained model to the video modality, *i.e.*, the video recognition task. To evaluate the effectiveness of the transferred model, we conduct experiments on **six** well-known video datasets, which include both trimmed and untrimmed video data. Specifically, the datasets are Kinetics-400 & 600 (Kay et al., 2017; Carreira et al., 2018), UCF-101 (Soomro et al., 2012), HMDB-51 (Kuehne et al., 2011), ActivityNet-v1.3 (Caba Heilbron et al., 2015), and Charades (Sigurdsson et al., 2016). These datasets are selected to represent a wide range of video recognition tasks, and are commonly used as benchmarks in this field.

We evaluate the transferred model in three distinct scenarios: zero-shot, few-shot, and regular video recognition. In the zero-shot scenario, the model has not trained on the target dataset but is evaluated on it, allowing us to assess its ability to generalize to new data. In the few-shot scenario, the model is trained on a small subset of the target dataset and evaluated on the validation set, enabling us to explore its capacity to learn from limited labeled data. In the typical recognition scenario, the model is trained on the entire target dataset and evaluated on the validation set, allowing us to measure its performance in a standard supervised learning configuration. By evaluating the model in these three scenarios, we aim to provide a comprehensive assessment of its performance under different conditions.

4.1 Training

The video recognition task takes a video as input, then feeds 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 initialized from the textual embedding of the class names and then frozen (fixed), leaving only the parameters in the video encoder to be learned.

Default Training Recipe Table 1 presents our training details for regular video recognition. We share the same recipe on all the video datasets, *i.e.*, Kinetics-400, ActivityNet, HMDB-51, UCF-101, and Charades.

Few-Shot Video Recognition All training strategies employed in the training process are consistent with those presented in Table 1, with only one modification: the number of epochs was increased to 100.

Zero-Shot Video Recognition We use the Kinetics-400 pre-trained models to directly perform cross-dataset zero-shot video recognition **without any additional training** on other datasets, *i.e.*, ActivityNet, HMDB-51, UCF-101 and Kinetics-600.

Table 1 Default training details for video recognition

Setting	Value
Training hyper-parameter	
Batch size	256
Vocabulary size	49408
Training epochs	30
Optimizer	AdamW
Learning rate (base, minimal)	(5e-5, 5e-6), cosine
Weight decay	0.2
Linear warm-up epochs	5
Adam β_1,β_2	0.9, 0.999
Augmentation	
Resize	RandomSizedCrop
Crop size	224 (Default)
Random flip	0.5
Random grayscale	0.2
RandAugment	$N = 2, M = 9$

4.2 Inference

To trade off accuracy and speed, we consider two inference strategies: (1) *Single View*: This strategy involves using only a single clip per video and the center crop for efficient evaluation, as shown in Table 4.3. (2) *Multiple Views*: This strategy, which is widely used in previous works, involves sampling multiple clips per video with several spatial crops to improve accuracy. For comparison with state-of-the-art approaches, we use four clips with three crops ("4×3 Views").

4.3 Ablation Studies

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 ***bold italics***.

Different Initializations to the Offline Classifier We first examine how the initializations affect the learning of classifiers. Then, we prepare our controlled environment using a classifier with parameters $W \in \mathbb{R}^{d\times c}$, which is built on the average of pooled temporal feature representations of all the frames. According to Sect. 3.3, we evaluate the performance of six types of initializations on both the few-shot and full-shot settings. For reference, we also provide the results using the standard vision-only fine-tuning (*i.e.*, online) classifier with trainable weights.

Table 2 lists the results. Feeding the offline classifier a random d-by-c matrix with a normal distribution leads to significantly reduced performance. Furthermore, removing the classifier's intra-modality correlation also results in inferior performance. From this family of initialization, we understand the necessity of a proper correlation in the classifier targets. Next, we observe that providing correlation information using a small labeled sub-set from the visual side leads to improved performance, with the classifier no longer guessing the results in the few-shot scenario and learning reasonably well in the full-shot scenario, compared to the vision-only online classifier. Compared to learnable correlation, pre-extracted proper correlation provides a more explicit target, making it a more efficient approach, especially in the process of transfer learning, particularly in few-shot learning. Notably, the class center initialization performs better than the LDA initialization in the few-shot scenario, demonstrating the CLIP encoder has a naturally well-distributed feature embedding.

Finally, we investigate the effect of the textual semantic family of initialization on the classifier's performance. We observe that the embeddings from the textual encoder of CLIP significantly improve the few-shot and full-shot accuracy. Interestingly, the DistilBERT-based initialization also performs remarkably well despite the semantics not being directly aligned with the visual modality. This result can be explained by the fact that both DistillBERT and CLIP are pre-trained with large-scale data and have strong language modeling capabilities, allowing them to generate good semantic targets. Therefore, we conclude that the visual

Table 2 The effects of different initializations for the frozen (offline) classifiers

Offline classifier	Alignment	Correlation	Few-shot Acc.	Full-shot Acc.
Random normal vectors	✗	Random	0.6	58.7
Random orthogonal vectors	✗	Non-correlation	0.6	57.7
Linear discriminant projection	✗	Visual statistics	25.5	79.6
Class center	✗	Visual statistics	32.3	79.0
DistilBERT	✗	Textual semantic	32.2	77.8
Textual encoder of CLIP	✓	Textual semantic	65.3	**80.1**
Vision-only (online)	✗	Learnt weight	21.6	75.3

Alignment denotes if the cross-modality knowledge is pre-aligned. **Correlation** shows the source of the correlation among the class embeddings

Table 3 Temporal modeling for video encoders

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***

embeddings benefit from the correlation of semantic targets, and the extract alignment further boosts the learning progress, reducing the need for a large number of samples.[1]

We also provide the visualizations of these classifiers in Fig. 3. Apparently, the latter two families of initializations share the same patterns among the correlation maps, which could be easily distinguished from the random and orthogonal ones.

Temporal Modeling In this study, we explore several temporal modeling strategies for both ViT and ResNet, including:

1. **TAP**: Temporal average pooling is a straightforward temporal modeling strategy that provides a simple baseline for comparison.
2. **T1D**: Channel-wise temporal 1D convolutions, which are commonly used in previous works (Wu et al., 2021a; Wang et al., 2021a; Liu et al., 2020), are employed to facilitate efficient temporal interaction in the later stages (res_{4-5}) of ResNet.
3. **T-Trans**: This strategy involves feeding the embeddings of frames to a multi-layer (*e.g.*, 6-layer) temporal transformer encoder.
4. **TokenT1D**: This approach involves using T1D to model temporal relations for the [class] token features that are aggregated from local features via attention in the vision transformer. We apply TokenT1D to multiple positions of a vision transformer to model temporal dependencies among the tokens.

Our experimental results are presented in Table 3. We observed that on both ViT and ResNet backbones, TAP provides a simple baseline for temporal modeling, and T-Trans achieves the best top-1 accuracy. Interestingly, we found that T1D does not appear to be effective in this scenario. This could be due to the potential for T1D to disrupt the strong representations learned by CLIP. In contrast, TokenT1D is another internal-backbone temporal modeling strategy that modifies only the global [class] token features instead of patch features. We observed that TokenT1D does not lead to a performance drop and even slightly improves the TAP baseline. We believe that this is because TokenT1D results in minimal modifications to the pre-trained features, which allows the model to retain the learned representations while incorporating temporal dependencies among the tokens.

Ours *v.s.* Contrastive-Based Paradigm we compare our proposed approach with the contrastive-based tuning method ActionClip (Wang et al., 2021b), which is introduced in Sect. 2.2. This paradigm treats the video recognition task as a video-text matching problem with a contrastive loss, which requires batch gathering to collect embeddings of all batches across all GPUs and calculate cosine similarity for a given batch across all other batches.

To ensure a fair comparison, we follow the official code and configurations from ActionClip (Wang et al., 2021b) in our experiments. In contrast to the contrastive-based paradigm, our recognition paradigm uses the Cross-Entropy loss to train the model, and we employ pre-extracted text embeddings as our classifier. Thus, the only learned part in our paradigm is the visual encoder, whereas the pre-trained textual encoder still needs to be updated in the Contrastive-based paradigm, requiring larger GPU memory. In Table 4, we compare our approach with the contrastive-based paradigm and observe that the latter performs poorly without batch gathering. This is because contrastive learning favors a large batch size, *e.g.*, CLIP (Radford et al., 2021) used 256 GPUs with a batch size of 128 per GPU to maintain a large 32768×32768 similarity matrix. Moreover, involving batch gathering will multiply the training time.

Our results demonstrate that our proposed approach achieves the best accuracy-cost trade-off. Specifically, our method achieves a performance of 81.5% with ViT-B/16, which takes only 10 h to run the training using 8 GPUs and is **2× faster** than the matching counterpart. Our approach is more efficient and effective for video recognition tasks, especially in applications with limited computational resources. *Please refer to Appendix §A.2 for further details on batch gathering.*

Additionally, in order to mitigate the impact of different implementation details, we have incorporated the contrastive-style training loss function based on our code. As observed in Table 5, training with contrastive loss introduces a reduction in training efficiency without significant performance improvement. Moreover, we have further enhanced the performance by incorporating a fixed offline classifier in the contrastive-style approach. This improvement can be attributed to the accelerated convergence achieved by the fixed textual target during training.

Text Input Forms We investigate several text input forms in Table 6, including class names, single hard template, multiple hard templates, and learnable templates. The details are as follows:

[1] We also observe that the loss of DistillBERT is initially higher than that of CLIP but quickly decreases to the same level.

Table 4 Ours *vs.* Contrastive-based paradigm with ViT-B/16 on Kinetics-400

Paradigm	Contrastive	Batch Gather	Textual Encoder	Top-1	V100-days
ActionCLIP (Wang et al., 2021b)	✓	✓	🔓	81.2	6.7 (10*)
		✓	🔒	80.7	6.6
		✗	🔓	77.8	3.5
		✗	🔒	76.1	3.3
Ours	✗	✗	🔒	***81.5***	**3.3**

Table 5 More ablations on contrastive-style paradigm

Contrastive	Offline classifier	Top-1(%)	Training time
✗	✗	81.5	1×
✓	✗	81.4	2×
✓	✓	81.7	2×

Table 6 Study on various text input forms

Text input form	Top-1
Class name	81.4
"A video of a person" + class name	***81.5***
Multiple fixed templates + class name	80.9
Learnable template + class name	81.2

1. **Class name.** To generate textual embeddings, we utilize the category names of the dataset as the text input, such as *"eating hotdog"* or *"driving car"*. The results show that using only the label text can yield good performance.
2. **Single hard template.** We use a hand-crafted template, *"a video of a person {class name}."* as input. This template only slightly improves performance over the label text's baseline.
3. **Multiple hard templates.** CLIP [2] provides 28 templates for Kinetics, including the single template described above. During training, we use these templates as text augmentation by randomly selecting one at each iteration. Then, we evaluate the model using the single template as input. The performance decreases by 0.6% on Kinetics-400, which may be because various prompt templates introduce extra noise during training.
4. **Learnable templates.** We use the automated prompt CoOp (Zhou et al., 2021) to describe a prompt's context using a set of learnable vectors. Specifically, the prompt given to the text encoder is designed with the following form:

$$\boldsymbol{t} = [\mathrm{V}]_1[\mathrm{V}]_2 \dots [\mathrm{V}]_M[\text{class name}], \tag{7}$$

where each $[\mathrm{V}]_m$ ($m \in \{1, \dots, M\}$) is a vector of the same size as word embeddings, and M is the number of context tokens. We set M to 4 in our experiments.

[2] https://github.com/openai/CLIP/blob/main/data/prompts.md.

Table 7 Analysis on throughput

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

"vid/s" denotes the average number of videos processed per second. A higher value of "vid/s" corresponds to greater efficiency. * represents the official result with TPU-v3

Our results suggest that different templates have little impact on our model's performance.

Computational Cost and Efficiency Table 7 presents our models' computational cost and efficiency, measured in terms of throughput using a single NVIDIA A100 GPU with a batch size of 16, which aligns with standard inference settings. Our models exhibit a **29× faster** throughput, and **44× fewer** FLOPs than the previous transformer-based method ViViT (Arnab et al., 2021), while maintaining the same accuracy. These results confirm the high efficiency of our approach.

4.4 Main Results

Regular Video Recognition We evaluate the performance of our model on the Kinetics-400 dataset, a challenging benchmark for regular video recognition. Table 8 provides a comparison of our model with state-of-the-art methods that were pre-trained on large-scale datasets such as ImageNet-21K (Deng et al., 2009), IG-65M (Ghadiyaram et al., 2019), JFT-300M (Sun et al., 2017), FLD-900M (Yuan et al., 2021), and JFT-3B (Zhai et al., 2021). To date, none of the three largest datasets (JFT-300M, FLD-900M, and JFT-3B) are open-sourced, and pre-trained models are not provided. Hence, we utilized the publicly available CLIP (Radford et al., 2021) checkpoints, which have been trained on 400 million web image-text pairs (WIT-400M). Signif-

Table 8 Comparison with previous works on Kinetics-400

Method	Input	Pre-train	Top-1	Top-5	FLOPs×Views	Param
NL I3D-101 (Wang et al., 2018b)	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
MViT-B, 64×3 (Fan et al., 2021)	64×224^2	Scratch	81.2	95.1	455×3×3	36.6
Methods with large-scale pre-training						
TimeSformer-L (Bertasius et al., 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
ViViT-H/16×2 (Arnab et al., 2021)	32×224^2	JFT-300M	84.8	95.8	8316×4×3	647.5
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., 2021a)	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., 2021a)	16×448^2	JFT-3B	87.2	-	–×1×3	–
VideoPrompt ViT-B/16 (Ju et al., 2022)	16×224^2	WIT-400M	76.9	93.5	–	–
ActionCLIP ViT-B/16 (Wang et al., 2021b)	32×224^2	WIT-400M	83.8	96.2	563×10×3	141.7
ST-Adapter ViT-L/14 (Pan et al., 2022)	32×224^2	WIT-400M	87.2	97.6	8248	–
EVL ViT-L/14 (Lin et al., 2022b)	32×224^2	WIT-400M	87.3	–	8088	–
EVL ViT-L/14 (Lin et al., 2022b)	32×336^2	WIT-400M	87.7	–	18196	–
Ours ViT-L/14	32×224^2	WIT-400M	87.6	97.8	1662×4×3	230.7
Ours ViT-L/14	32×336^2	WIT-400M	**88.4**	**98.0**	3829×4×3	230.7
Ours ViT-L/14	32×336^2	Merged-2B	**89.4**	**98.1**	3829×4×3	230.7

"Views" indicates # temporal clip × # spatial crop. The magnitudes are Giga (10^9) and Mega (10^6) for FLOPs and Param. "IN" denotes ImageNet

icantly, by utilizing the same CLIP pre-trained backbones, our model demonstrates substantial performance improvements over EVL (Lin et al., 2022b) and ST-Adapter (Pan et al., 2022). Furthermore, our method achieves superior performance compared to methods that were pre-trained with JFT-300 M (Sun et al., 2017) or FLD-900 M (Yuan et al., 2021), while requiring less computational cost or a smaller resolution. Furthermore, with the significant scale-up of the pre-training data to 2 billion samples (namely Merged-2B (Sun et al., 2023), which merges 1.6 billion samples from the LAION-2B (Schuhmann et al., 2022) dataset with 0.4 billion samples from the COYO-700 M (Byeon et al., 2022) dataset), our method achieves an outstanding top accuracy of **89.4%**, solidifying its position as a state-of-the-art approach.

To verify the generalization ability of our method, we further evaluate its performance on the widely-used untrimmed video benchmark, **ActivityNet-v1.3**. Specifically, we finetune the Kinetics-400 pre-trained models, with ViT-L backbone and 16 frames, on the ActivityNet-v1.3 dataset. The top-1 accuracy and mean average precision (mAP) are reported using the official evaluation metrics. As shown in Table 9, our method outperforms recent state-of-the-art models with a clear margin, with an mAP accuracy of 96.9%.

Table 9 Comparisons with previous works on ActivityNet

Method	Top-1	mAP
ListenToLook (Gao et al., 2020)	–	89.9
MARL (Wu et al., 2019b)	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***

We also evaluate our method on the **UCF-101** and **HMDB-51** datasets to demonstrate its capacity to generalize to smaller datasets. We finetune our models on these two datasets using the pre-trained ViT-L model on Kinetics-400 and present the mean class accuracy on split one. We utilize

Table 10 **Mean class accuracy** on UCF-101 and HMDB-51 achieved by different methods which are transferred from their **Kinetics** models with RGB modality

Method	UCF-101	HMDB-51
ARTNet (Wang et al., 2018a)	94.3%	70.9%
I3D (Carreira & Zisserman, 2017)	95.6%	74.8%
R(2+1)D (Tran et al., 2018)	96.8%	74.5%
S3D-G (Xie et al., 2018)	96.8%	75.9%
TSM (Lin et al., 2019)	95.9%	73.5%
STM (Jiang et al., 2019)	96.2%	72.2%
TEINet (Liu et al., 2020)	96.7%	72.1%
MVFNet (Wu et al., 2021a)	96.6%	75.7%
TDN (Wang et al., 2021a)	97.4%	76.4%
Ours ViT-L	**98.1%**	**81.3%**
Ours ViT-L (336↑)	**98.2%**	**81.3%**

16 frames as inputs. As shown in Table 10, our model exhibited strong transferability, achieving a mean class accuracy of 98.2% on UCF-101 and 81.3% on HMDB-51.

Few-Shot Video Recognition In few-shot video recognition, where only a few training samples are available, we investigate a more challenging K-shot C-way situation, instead of the conventional 5-shot 5-way configuration. We aim to categorize **all** categories in the dataset with just K samples per category for training, where the lower and upper bounds are denoted by the terms "Zero-shot" and "All-shot", respectively. Using the CLIP-pretrained ViT-L/14 with 8 frames and TAP for few-shot video recognition, we report the Top-1 accuracy for the four datasets in Table 11. Despite the limited amount of data, our method demonstrates remarkable transferability to diverse domain data. Furthermore, our approach outperforms previous methods significantly, showing robustness in these extremely data-poor situations. For instance, when comparing the accuracy on HMDB-51 with 2-shot, our method outperforms Swin (Liu et al., 2022) and X-Florence (Ni et al., 2022) by **+52.6%** and **+21.9%**, respectively.

Multi-Label Video Recognition We mainly focused on the single-label video recognition scenario in the previous experiments. To further validate the performance of our method, we conducted experiments on multi-label video recognition tasks. The Charades dataset is a multi-label untrimmed video dataset containing long-term activities with multiple actions. For this task, we utilized the Kinetics-400 pre-trained ViT-L backbone for training and evaluated our results using the Mean Average Precision (mAP) metric. As shown in Table 12, our method achieved the highest performance of 46.0 mAP, demonstrating its effectiveness in multi-label video classification.

Table 12 Comparison with previous works on **Multi-Label** video dataset Charades

Method	Frames	mAP
MultiScale TRN (Zhou et al., 2018)	–	25.2%
STM (Jiang et al., 2019)	16	35.3%
Nonlocal (Wang et al., 2018b)	–	37.5%
SlowFast R50 (Feichtenhofer et al., 2019)	8+32	38.0%
SlowFast R101 (Feichtenhofer et al., 2019)	16+64	42.5%
LFB+NL (Wu et al., 2019a)	32	42.5%
X3D-XL (312↑) (Feichtenhofer, 2020)	16	43.4%
ActionCLIP (Wang et al., 2021b)	32	44.3%
Ours	16	**46.0%**

Zero-Shot Video Recognition In addition, we conducted experiments in the open-set setting. We use our Kinetics-400 pre-trained models (*i.e.*, ViT-L with 8 frames) to perform zero-shot evaluations on four other video datasets. For UCF-101, HMDB-51, and ActivityNet, we follow two evaluation protocols from E2E (Brattoli et al., 2020):

1. To make a fair comparison with previous works, we randomly selected half of the test dataset's classes: 50 for UCF-101, 25 for HMDB-51, and 100 for ActivityNet, and evaluated our method on them. We repeated this process ten times and averaged the results for each test dataset. We refer to this setting as UCF*, HMDB*, and ANet*.
2. In the second evaluation protocol, we directly evaluated the full datasets to obtain more realistic accuracy scores.

Table 11 Comparisons with previous works on few-shot action recognition

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 13 Comparison with previous works on zero-shot video recognition

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 et al., 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 & Huang, 2021)	51.8±2.9 / –	35.3±4.6 / –	–	42.1±1.4
ResT (Lin et al., 2022a)	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***

We directly evaluate our method without any additional training on cross-dataset video recognition. ANet is short for ActivityNet. * means half classes evaluation

For Kinetics-600, we chose 220 new categories outside Kinetics-400 for evaluation. We used the three splits provided by Chen and Huang (2021) and sampled 160 categories for evaluation from the 220 categories in Kinetics-600 for each split. We reported the mean accuracy for the three splits. As shown in Table 13, our method demonstrates a strong cross-dataset generalization ability, achieving significant improvements over previous zero-shot video recognition methods (**+27.1%** on UCF-101, **+17.0%** on HMDB-51, **+52.1%** on ActivityNet, **+26.8%** on Kinetics-600).

5 Experiments: Image Recognition

In this work, we also apply our method to image recognition. We conduct a comprehensive evaluation on **10** datasets that represent a diverse set of visual recognition tasks, *i.e.*, ImageNet (Deng et al., 2009), StanfordCars (Krause et al., 2013), Caltech101 (Fei-Fei et al., 2004), OxfordPets (Parkhi et al., 2012), Flowers102 (Nilsback & Zisserman, 2008), Food101 (Bossard et al., 2014), FGVCAircraft (Maji et al., 2013), SUN397 (Xiao et al., 2010), DTD (Cimpoi et al., 2014), EuroSAT (Helber et al., 2019). These tasks include classifying generic objects, scenes, and fine-grained categories and specialized tasks such as texture recognition and satellite imagery analysis.

5.1 Training

For the pre-trained CLIP model, we use the ResNet-50 (He et al., 2016) as the default backbone for the image encoder, and the image backbone is updated during training. We train the model using the AdamW optimizer with an initial learning rate of 5e-6 and a cosine annealing schedule to reduce the learning rate gradually. We also employ a warmup strategy of 5 epochs. The maximum number of training epochs is set to 150.

5.2 Main Results

Results on 10 Image Datasets As illustrated in Fig. 4, the performance of our method, vision-only method, and zero-shot method are evaluated on 10 image datasets, all trained with 16 shots. The results of 10 datasets are arranged from left to right, and the average results of the 10 datasets are presented on the far right. Our findings reveal that CLIP showcases strong zero-shot performance on all 10 datasets. However, the vision-only method exhibits poor performance on all datasets. We posit that this may be attributed to the absence of a suitable classifier target. Consequently, it may be susceptible to biases in small samples, which can disrupt the well-pretrained image encoder. Our approach demonstrates a substantial improvement in recognition accuracy compared to both the vision-only and zero-shot methods on all 10 datasets. Specifically, the average improvement over the 10 datasets is 41% and 18% compared to the vision-only method and the zero-shot method, respectively. This indicates the effectiveness of our method in enhancing few-shot learning performance.

Table 14 presents a further comparison of our method with two other transfer methods, specifically linear probe and CoOp (Zhou et al., 2021), on the widely used ImageNet dataset. We implemented the linear probe method as instructed in the original CLIP paper (Radford et al., 2021). Our findings indicate that the CoOp method contributes to a significant enhancement of the zero-shot model by 4.77%. Notably, our proposed approach surpasses this performance by achieving a further improvement of 8.33% on the zero-shot model, underscoring the effectiveness of our method in incorporating an appropriate classifier target.

6 Experiments: 3D Point Cloud Recognition

We further extend our approach to 3D point cloud recognition and evaluated it on the ModelNet40 dataset (Wu et al., 2015). This dataset comprises 12,311 3D CAD models across

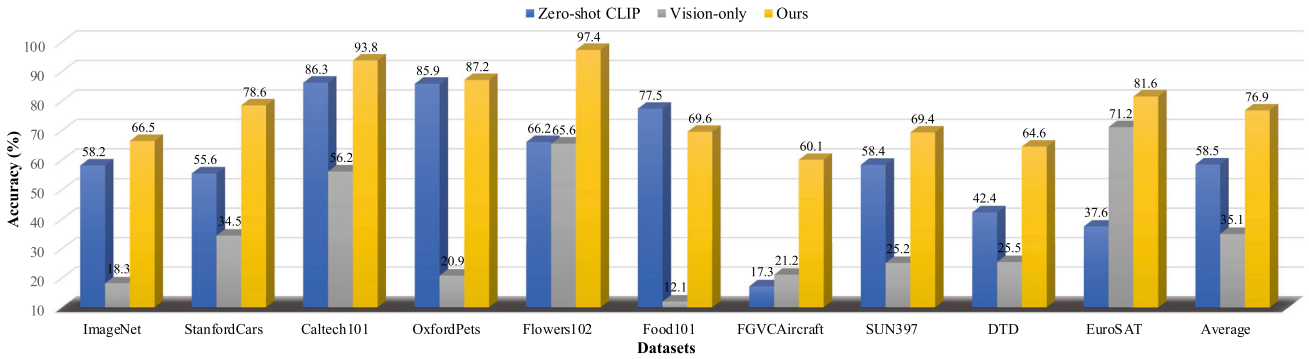

Fig. 4 Comparison of few-shot learning performance on 10 image datasets. Assessment of zero-shot CLIP, vision-only, and the proposed method underlines the significance of incorporating a suitable classifier target to mitigate biases in small samples and achieve high accuracy on a diverse set of image datasets

Table 14 Comparison of our method with other tuning methods on ImageNet (using 16 shots)

	ImageNet	Δ
Zero-shot CLIP	58.18	–
Linear probe	55.87	**-2.31**
CoOp (Zhou et al., 2021)	62.95	**+4.77**
Ours	**66.51**	**+8.33**

Δ denotes the difference with the zero-shot model

40 categories: airplanes, cars, plants, and lamps. The point clouds are normalized to a unit sphere and divided into 9,843 training models and 2,468 testing models. ModelNet40 is a widely used benchmark for point cloud recognition.

6.1 Training

For the visual encoder, we use the ResNet-101 architecture (He et al., 2016) as the default backbone and apply multi-view perspective projection on the input point cloud following SimpleView (Goyal et al., 2021). SimpleView projects the point cloud from six orthogonal views: front, right, back, left, top, and bottom. In addition, we also include the views of the upper/bottom-front/back-left corners based on the observation from Zhang et al. (2022) that the left view is the most informative for few-shot recognition. For each view, a point with a 3D coordinate is projected onto a pixel on the 2D image plane, and its depth value is used as the pixel intensity, which is repeated three times for the RGB channels. Finally, all the resulting images are upsampled to (224, 224) to align with CLIP's settings.

We train the model using the Stochastic Gradient Descent (SGD) optimizer with an initial learning rate of 2e-4 and a cosine annealing schedule to reduce the learning rate gradually. We also employ a warmup strategy of 10 epochs; the maximum number of training epochs is set to 250.

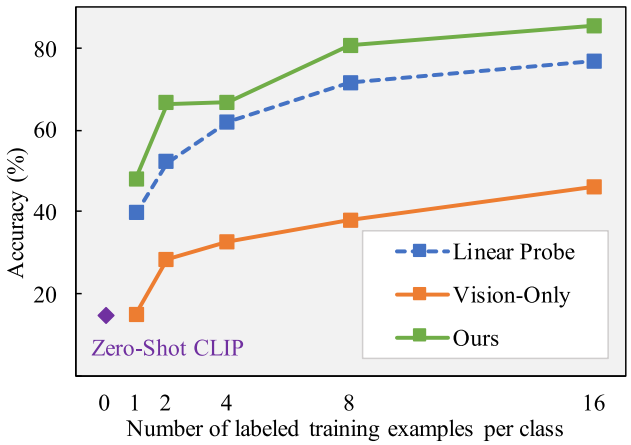

Fig. 5 Results of 3D point cloud recognition on the ModelNet40 dataset. Comparison of different tuning methods in the few-shot scenario

6.2 Main Results

Comparison with the Vision-Only Paradigm in the Few-Shot Scenario We evaluate our method using the few-shot evaluation protocol adopted in CLIP (Radford et al., 2021), which involves training with 1, 2, 4, 8, and 16 shots and deploying models on the full test set. As shown in Fig. 5, we first present our method's zero-shot result (14.9%), which was obtained by directly utilizing the CLIP model to classify each view and averaging the results of the 10 views. We then compare the performance of different tuning models on 3D point cloud recognition. Our results show that all models gradually improve in accuracy as the number of training samples increases. Notably, our method (green curve) outperforms the vision-only method (orange curve) by a large absolute improvement of 30%-40%, which is consistent with findings in image recognition and video recognition, and validates the effectiveness of our approach. Additionally, we find that our method significantly outperforms the linear probe method (blue curve) at all training sample levels, known as a strong few-shot learning baseline. These results confirm

the effectiveness and superiority of our proposed approach, which involves textual knowledge to improve transferability.

7 Conclusion and Limitation

This study presents a new paradigm for enhancing the transferability of visual recognition tasks based on the knowledge from the textual encoder of a well-trained vision-language model. Specifically, we initialize the classifier with semantic targets from the textual encoder and freeze it during optimization. We conduct extensive experiments to examine how the paradigm functions: Firstly, we demonstrate that proper correlation among target initialization is beneficial. Secondly, we show that alignment of visual and textual semantics is key to improving few-shot performance and shortening the learning progress. Finally, we verify the effectiveness of our proposed paradigm on three types of visual recognition tasks (*i.e.*, image, video, and 3D point cloud recognition) across 17 visual datasets.

The study still has some limitations worth diving into in future research. i) The performance of the proposed paradigm is restricted to how the category labels are represented. For instance, in tasks such as human re-identification, where the labels are often 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. ii) The performance of the proposed paradigm relies on the capacity of the vision-language pre-training models. Although we use CLIP as our source model in this study, obtaining models with better performance remains an open problem. iii) The way category names are described also impacts performance. For example, in the action recognition dataset Something-Something, category names such as "Putting something into something" and "Covering something with something" lack a clear target subject. Consequently, leveraging the prior knowledge of pre-aligned vision-language models becomes challenging, resulting in subpar performance.

Acknowledgements Wanli Ouyang was 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.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Appendix

A Additional Details

In this appendix, §A contains additional *details* for: the statistics of video datasets (§A.1), visual encoder architectures (§A.4), Batch Gather (§A.2) and LDA (§A.3).

A.1 Statistics of Video Datasets

Kinetics-400 (Kay et al., 2017) is a large-scale video dataset, which consists of 240k training videos and 20k validation videos in 400 different human action categories. Each video in the dataset is a 10-second clip of an action moment annotated from raw YouTube videos.
Kinetics-600 (Carreira et al., 2018) is an extensions of Kinetics-400. Kinetics-600 consists of around 480k videos from 600 action categories. The 480K videos are divided into 390k, 30k, and 60k for training, validation, and test sets, respectively. In this paper, we use its test set for zero-shot evaluation.
UCF-101 (Soomro et al., 2012) contains 13k videos spanning over 101 human actions.
HMDB-51 (Kuehne et al., 2011) contains approximately 7k videos belonging to 51 action class categories.
ActivityNet-v1.3 (Caba Heilbron et al., 2015) is a large-scale untrimmed video benchmark, contains 19,994 untrimmed videos of 5 to 10 min from 200 activity categories.
Charades (Sigurdsson et al., 2016) is a video dataset designed for action recognition and localization tasks. It contains over 10,000 short video clips of people performing daily activities, and consists of 157 action categories.

A.2 Batch Gather for Distributed InfoNCE

Instead of Data-Parallel Training (DP), which is single-process, multi-thread, and only works on a single machine, Distributed Data-Parallel Training (DDP) is a widely adopted single-program multiple-data training paradigm for single- and multi-machine training. Due to GIL contention across threads, per-iteration replicated model, and additional overhead introduced by scattering inputs and gathering outputs, DP is usually slower than DDP even on a single machine. Hence, we develop the Distributed InfoNCE based on DDP for large batch size and fast training. The core of the Distributed InfoNCE implementation is batch gathering. Say

Algorithm 1 Numpy-like Pseudocode that illustrates the role of Batch Gather in Distributed InfoNCE.

```
# text_encoder: encoder network for text input
# vision_encoder: encoder network for vision input, e.g., images or videos.
# V: minibatch of vision inputs
# T: minibatch of text inputs
# N: the local batch size of each GPU, e.g.,16
# M: the number of GPUs, e.g.,8
# N * M: the global batch size for multi-gpu training, e.g.,128

# extract feature representations of each modality
local_vision_features = vision_encoder(V) # shape: [N, embed_dim]
local_text_features = text_encoder(T) # shape: [N, embed_dim]

# normalization
local_vision_features = l2_normalize(local_vision_features, axis=1)
local_text_features = l2_normalize(local_text_features, axis=1)

# batch_gather is a function gathering and concatenating the tensors across GPUs.
all_vision_features = batch_gather(local_vision_features) # shape: [N * M, embed_dim]
all_text_features = batch_gather(local_text_features) # shape: [N * M, embed_dim]

# scaled pairwise cosine similarities
# shape = [N, N * M]
logits_vision = logit_scale * local_vision_features @ all_text_features.t()
# shape = [N, N * M]
logits_text = logit_scale * local_text_features @ all_vision_features.t()

# The logits are then used as inputs for N*M-way (e.g., 128-way) classification,
# resulting in a loss value corresponding to N inputs in each GPU.
# Then Distributed Data Parallel mechanism takes care of averaging these across GPUs,
# which becomes equivalent to calculating the loss over NMxNM (e.g.,128x128) similarities.
```

Algorithm 2 The code generates the LDA coefficient for Kinetics-400 dataset.

```
import numpy as np
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
input = np.load('feats_labels_400class.npz')   # pre-extracted visual features
feats = input['feats']  # size: [24000, 512]
labels = input['labels']  # size: [24000,]
lda = LDA()
lda.fit(feats, labels)
classifier = lda.coef_ # size: [400, 512]
```

there are M GPUs and each GPU gets N input pairs, we need to calculate the NM×NM similarity matrix across the GPUs for InfoNCE loss. Without batch gathering, each GPU only computes a local N×N matrix, *s.t.* N≪NM, Then the cosine similarity and the InfoNCE loss would be calculated only for the pairs within a single GPU and later their gradients would be averaged and synced. That's obviously not what we want.

The batch gathering for Distributed InfoNCE is presented as follows. When calculating the similarity matrix (and thus the logit scores across text inputs for each image/video), a GPU only needs to hold M vision features, and perform matrix product with NM text features, yielding an M×NM matrix. This computation is distributed (i.e., sharded) across N GPUs, and we have calculated NM×NM similarities across the GPUs in total. The loss we employ is symmetric and the same happens *w.r.t.* text inputs. As shown in Algorithm 1, we also give an example pseudocode to help you understand the statement.

A.3 LDA Classifier

Here we provide the details of LDA classifier. We directly use the official CLIP-pretrained visual encoder to extract video embeddings, and the visual encoder is not finetuned on Kinetics-400. Then we perform LDA on the pre-extracted video embeddings of the training set in Kinetics-400 to initialize W and freeze it for finetuning the visual encoder on the Kinetics-400 dataset.

LDA is commonly used for feature classification or feature dimensionality reduction. However, in this work, we only use LDA for feature classification (in order to get "discriminant coefficients" as the classifier) instead of feature dimensionality reduction. For better understanding, we show the code in Algorithm 2 which generates the LDA coefficient and there is no dimension reduction.

A.4 Visual Encoder Architectures

We provide the full architecture details of the visual encoder and textual encoders in this paper. Table 15 shows the CLIP-ResNet architectures. Table 16 shows the CLIP-ViT architectures.

Table 15 CLIP-ResNet hyperparameters

Model	Embedding dimension	Input resolution	ResNet		Text Transformer		
			blocks	width	layers	width	heads
RN50	1024	224	(3, 4, 6, 3)	2048	12	512	8

Table 16 CLIP-ViT hyperparameters

Model	Embedding dimension	Input resolution	Vision Transformer			Text Transformer		
			layers	width	heads	layers	width	heads
ViT-B/32	512	224	12	768	12	12	512	8
ViT-B/16	512	224	12	768	12	12	512	8
ViT-L/14	768	224	24	1024	16	12	768	12
ViT-L/14-336px	768	336	24	1024	16	12	768	12

References

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

Bertasius, G., Wang, H., & Torresani, L. (2021). Is space-time attention all you need for video understanding? In *ICML, PMLR* (pp. 813–824).

Bossard, L., Guillaumin, M., & Van Gool, L. (2014). Food-101–mining discriminative components with random forests. In *ECCV*.

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

Byeon, M., Park, B., Kim, H., Lee, S., Baek, W., & Kim, S. (2022). Coyo-700m: Image-text pair dataset. https://github.com/kakaobrain/coyo-dataset

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

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

Carreira, J., Noland, E., Banki-Horvath, A., Hillier, C., & Zisserman, A. (2018). A short note about kinetics-600. arXiv preprint arXiv:1808.01340

Chen, S., & Huang, D. (2021). Elaborative rehearsal for zero-shot action recognition. In *ICCV* (pp. 13638–13647).

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

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., & Vedaldi, A. (2014). Describing textures in the wild. In *CVPR*.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *CVPR* (pp. 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., & Feichtenhofer, C. (2021). Multiscale vision transformers. In *ICCV* (pp. 6824–6835).

Fei-Fei, L., Fergus, R., & Perona, P. (2004). Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. In *Computer vision and pattern recognition workshop*.

Feichtenhofer, C. (2020). X3d: Expanding architectures for efficient video recognition. In *CVPR* (pp. 203–213).

Feichtenhofer, C., Fan, H., Malik, J., & He, K. (2019). Slowfast networks for video recognition. In *ICCV* (pp. 6202–6211).

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

Gao, P., Geng, S., Zhang, R., Ma, T., Fang, R., Zhang, Y., Li, H., & Qiao, Y. (2021). Clip-adapter: Better vision-language models with feature adapters. arXiv preprint arXiv:2110.04544

Gao, R., Oh, T. H., Grauman, K., & Torresani, L. (2020). Listen to look: Action recognition by previewing audio. In *CVPR* (pp. 10457–10467).

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

Goyal, A., Law, H., Liu, B., Newell, A., & Deng, J. (2021). Revisiting point cloud shape classification with a simple and effective baseline. In *International conference on machine learning, PMLR* (pp. 3809–3820).

Han, K., Xiao, A., Wu, E., Guo, J., Xu, C.,& Wang, Y. (2021). Transformer in transformer. In *NeurIPS* (pp. 15908–15919).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *CVPR* (pp. 770–778).

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

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 16000–16009).

Helber, P., Bischke, B., Dengel, A., & Borth, D. (2019). Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing., 12*(7), 2217–2226.

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

Jia, C., Yang, Y., Xia, Y., Chen, Y. T., Parekh, Z., Pham, H., Le, Q., Sung, Y. H., Li, Z., & Duerig, T. (2021a). Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning, PMLR* (pp. 4904–4916).

Jia, C., Yang, Y., Xia, Y., Chen, Y. T., Parekh, Z., Pham, H., Le, Q., Sung, Y. H., Li, Z., & Duerig, T. (2021b). Scaling up visual and vision-

language representation learning with noisy text supervision. In *ICML, PMLR* (pp. 4904–4916).
Jiang, B., Wang, M., Gan, W., Wu, W., & Yan, J. (2019). Stm: Spatiotemporal and motion encoding for action recognition. In *ICCV* (pp. 2000–2009).
Ju, C., Han, T., Zheng, K., Zhang, Y., & Xie, W. (2022). Prompting visual-language models for efficient video understanding. In *ECCV* (pp. 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., & Hager, G. D. (2021). Daszl: Dynamic action signatures for zero-shot learning. *AAAI,* (vol. 35, pp. 1817–1826).
Krause, J., Stark, M., Deng, J., & Fei-Fei, L. (2013). 3D object representations for fine-grained categorization. In *4th International IEEE workshop on 3D representation and recognition (3dRR-13)*, Sydney, Australia.
Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *NeurIPS* (pp. 25).
Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., & Serre, T. (2011). Hmdb: A large video database for human motion recognition. In *ICCV* (pp. 2556–2563).
Li, B., Weinberger, K. Q., Belongie, S., Koltun, V., & Ranftl, R. (2022a). Language-driven semantic segmentation. arXiv preprint arXiv:2201.03546
Li, J., Li, D., Xiong, C., & Hoi, S. (2022b). Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. arXiv preprint arXiv:2201.12086
Li, T., Zhu, S., & Ogihara, M. (2006). Using discriminant analysis for multi-class classification: An experimental investigation. *Knowledge and Information Systems, 10*(4), 453–472.
Lin, C. C., Lin, K., Wang, L., Liu, Z., & Li, L. (2022a). Cross-modal representation learning for zero-shot action recognition. In *CVPR* (pp. 19978–19988).
Lin, J., Gan, C., & Han, S. (2019). Tsm: Temporal shift module for efficient video understanding. In *ICCV*.
Lin, Z., Geng, S., Zhang, R., Gao, P., de Melo, G., Wang, X., Dai, J., Qiao, Y., & Li, H. (2022b). Frozen clip models are efficient video learners. In *ECCV* (pp. 388–404), Springer.
Liu, Z., Luo, D., Wang, Y., Wang, L., Tai, Y., Wang, C., Li, J., Huang, F., & Lu, T. (2020). Teinet: Towards an efficient architecture for video recognition. In *AAAI* (pp. 11669–11676).
Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV* (pp. 10012–10022).
Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., & Hu, H. (2022). Video swin transformer. In *CVPR* (pp. 3202–3211).
Lüddecke, T., & Ecker, A. (2022). Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7086–7096).
Luo, H., Ji, L., Zhong, M., Chen, Y., Lei, W., Duan, N., & Li, T. (2021). Clip4clip: An empirical study of clip for end to end video clip retrieval. arXiv preprint arXiv:2104.08860
Maji, S., Rahtu, E., Kannala, J., Blaschko, M., & Vedaldi, A. (2013). Fine-grained visual classification of aircraft. arXiv preprint arXiv:1306.5151
Mishra, A., Verma, V. K., Reddy, M. S. K., Arulkumar, S., Rai, P., & Mittal, A. (2018). A generative approach to zero-shot and few-shot action recognition. In *WACV* (pp. 372–380).
Mokady, R., Hertz, A., & 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., & Ling, H. (2022). Expanding language-image pretrained models for general video recognition. In *ECCV*.
Nilsback, M. E., & Zisserman, A. (2008). Automated flower classification over a large number of classes. In *ICVGIP*.
Van den Oord, A., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. arXiv e-prints (pp. arXiv–1807).
Pan, J., Lin, Z., Zhu, X., Shao, J., & Li, H. (2022). St-adapter: Parameter-efficient image-to-video transfer learning for action recognition. arXiv preprint arXiv:2206.13559
Parkhi, O. M., Vedaldi, A., Zisserman, A., & Jawahar, C. (2012). Cats and dogs. In *CVPR*.
Qiu, Z., Yao, T., & Mei, T. (2017). Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV* (pp. 5533–5541).
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, PMLR* (pp. 8748–8763).
Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., & Sutskever, I. (2021). Zero-shot text-to-image generation. In *ICML, PMLR* (pp. 8821–8831).
Rao, Y., Zhao, W., Chen, G., Tang, Y., Zhu, Z., Huang, G., Zhou, J., & Lu, J. (2022). Denseclip: Language-guided dense prediction with context-aware prompting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 18082–18091).
Ribani, R., & Marengoni, M. (2019). A survey of transfer learning for convolutional neural networks. In *2019 32nd SIBGRAPI conference on graphics, patterns and images tutorials (SIBGRAPI-T)* (pp. 47–57), IEEE.
Ryoo, M. S., Piergiovanni, A., Arnab, A., Dehghani, M., & 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., & Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108
Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. (2022). Laion-5b: An open large-scale dataset for training next generation image-text models. arXiv preprint arXiv:2210.08402
Sigurdsson, G. A., Varol, G., Wang, X., Farhadi, A., Laptev, I., & Gupta, A. (2016). Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Computer vision–ECCV 2016: 14th European conference, Amsterdam, The Netherlands, proceedings, part I 14*, (pp. 510–526), Springer.
Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
Soomro, K., Zamir, A. R., & 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.,& Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV* (pp. 843–852).
Sun, Q., Fang, Y., Wu, L., Wang, X.,& Cao, Y. (2023). Eva-clip: Improved training techniques for clip at scale. arXiv preprint arXiv:2303.15389
Sun, Z. (2022). Design of the topology for contrastive visual-textual alignment. arXiv preprint arXiv:2209.02127
Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). A survey on deep transfer learning. In *Artificial neural networks and machine learning–ICANN 2018: 27th international conference on artificial neural networks, Rhodes, Greece, proceedings, part III 27* (pp. 270–279), Springer.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., & Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In *CVPR* (pp. 6450–6459).

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

Wang, L., Li, W., Li, W., & Van Gool, L. (2018a). Appearance-and-relation networks for video classification. In *CVPR*.

Wang, L., Tong, Z., Ji, B., & Wu, G. (2021a). Tdn: Temporal difference networks for efficient action recognition. In *CVPR* (pp. 1895–1904).

Wang, M., Xing, J., & Liu, Y. (2021b). Actionclip: A new paradigm for video action recognition. arXiv preprint arXiv:2109.08472

Wang, X., Girshick, R., Gupta, A., & He, K. (2018b). Non-local neural networks. In *CVPR* (pp. 7794–7803).

Wu, C. Y., Feichtenhofer, C., Fan, H., He, K., Krahenbuhl, P., & Girshick, R. (2019a). Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 284–293).

Wu, W., He, D., Tan, X., Chen, S., & Wen, S. (2019b). Multi-agent reinforcement learning based frame sampling for effective untrimmed video recognition. In *ICCV* (pp. 6222–6231).

Wu, W., He, D., Lin, T., Li, F., Gan, C., & Ding, E. (2021). Mvfnet: Multi-view fusion network for efficient video recognition. *AAAI* (vol. 35, pp. 2943–2951).

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. In *ACM MM* (pp. 1903–1911).

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., & Xiao, J. (2015). 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1912–1920).

Xia, B., Wang, Z., Wu, W., Wang, H., & Han, J. (2022a). Temporal saliency query network for efficient video recognition. In *ECCV* (pp. 741–759).

Xia, B., Wu, W., Wang, H., Su, R., He, D., Yang, H., Fan, X., & Ouyang, W. (2022b). Nsnet: Non-saliency suppression sampler for efficient video recognition. In *ECCV* (pp. 705–723).

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., & Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*.

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

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

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

Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., & 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., & Beyer, L. (2021). Scaling vision transformers. arXiv preprint arXiv:2106.04560

Zhang, B., Yu, J., Fifty, C., Han, W., Dai, A. M., Pang, R., & Sha, F. (2021a). Co-training transformer with videos and images improves action recognition. arXiv preprint arXiv:2112.07175

Zhang, R., Fang, R., Zhang, W., Gao, P., Li, K., Dai, J., Qiao, Y., & Li, H. (2021b). Tip-adapter: Training-free clip-adapter for better vision-language modeling. arXiv preprint arXiv:2111.03930

Zhang, R., Guo, Z., Zhang, W., Li, K., Miao, X., Cui, B., Qiao, Y., Gao, P., & Li, H. (2022). Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8552–8562).

Zhao, S., Zhu, L., Wang, X., & Yang, Y. (2022). Centerclip: Token clustering for efficient text-video retrieval. In *SIRIR*.

Zhou, B., Andonian, A., Oliva, A., & Torralba, A. (2018). Temporal relational reasoning in videos. In *ECCV*.

Zhou, K., Yang, J., Loy, C. C., & Liu, Z. (2021). Learning to prompt for vision-language models. arXiv preprint arXiv:2109.01134

Zhou, K., Yang, J., Loy, C. C., & Liu, Z. (2022). Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 16816–16825).

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE, 109*(1), 43–76.