Cross-dataset Training of Transformers for Robust Action Recognition

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study on robust feature representations that can generalize on multiple datasets for action recognition using transformers. Although we have witnessed great progress of action recognition in the past decade, it remains challenging yet valuable how to train a single model that can perform well across multiple datasets. Here we propose a novel cross-dataset training paradigm, ***CrossRoad***, with the design of two new loss terms, namely informative loss and projection loss, aiming to learn robust representations for action recognition. We verify the effectiveness of our method on five challenging datasets, Kinetics-400, Kinetics-700, Moments-in-Time, Activitynet and Something-something-v2 datasets. Extensive experimental results show that our method can consistently improve the state-of-the-art performance. We will release our code and models.

1 Introduction

Human vision can recognize video actions efficiently despite the variations of scenes and domains. Convolutional neural networks (CNNs) [37, 38, 6, 33, 14] fully utilize the power of modern computational devices and employ spatial-temporal filters to recognize actions, which outperform traditional models such as oriented filtering in space time (HOG3D) [23]. However, due to the high variations in space-time, the state-of-the-art of action recognition is still far from being satisfactory, compared with the success of 2D CNNs in image recognition [19]. Recently, vision transformers like ViT [10], MViT [12] that are based on the self-attention [40] mechanism are proposed to tackle the problems of image and video recognition. Instead of modeling pixels as CNNs, transformers apply attentions on top of visual tokens. The inductive bias of translation invariance in CNNs makes it require less training data than pure-attention-transformers in general. However, transformer has the advantage that it can better harness the parallel processing units of modern computing devices such as GPUs and TPUs, making it more computationally efficient than CNNs. We have seen a rapid growth in video datasets [21] in recent years, which would make up for the shortcomings of data-hungry transformers. The video data has not only grown in quantity from hundreds to millions of videos [31] but also evolved from simple actions such as handshaking to complicated daily activities from the Kinetics-700 dataset [7]. Meanwhile, transformers combined with low-level convolutional operations have been proposed [12] to further improve the original design.

Due to the data-hungry nature of transformers, most transformer-based models for action recognition requires large-scale pre-training with image datasets such as ImageNet-21K [9] and JFT-3B [44] to achieve good performance. This pre-training and fine-tuning training paradigm is time-consuming and it is not parameter-efficient, meaning that for each action dataset, a new model need to be trained end-to-end. Different from large image datasets such as ImageNet-21K that covers a wide range of object classes, currently the most diverse action dataset, Kinetics-700, only contains 700 classes. Each action dataset may be also limited to a certain topic or camera views. For example, Moments-

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

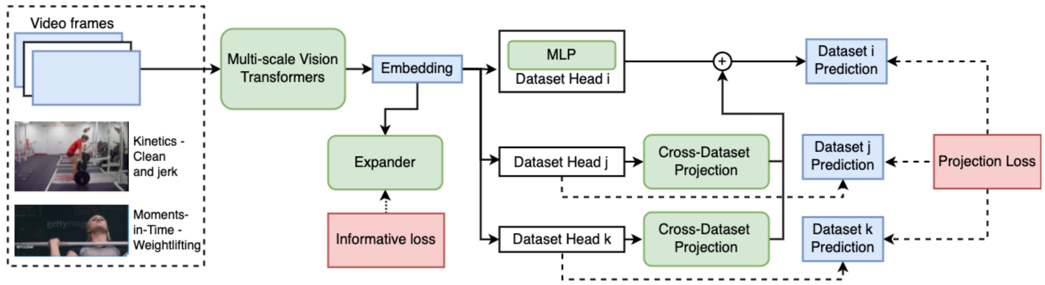

Figure 1: Overview of our cross-dataset training framework. We propose to utilize the intrinsic relations between classes across different action datasets. As we see, the two video examples from Kinetics and Moments-in-Time dataset, respectively, show that samples from these two classes can be used to train both classification heads. The videos from multiple action datasets are input to the MViTv2 (see Section 3.1) backbone and the model is trained jointly. The informative loss is applied to maximize the information content of the embedding from the backbone and projection loss is applied to learn the intrinsic relations (see Section 3.2).

in-Time [31] only contains short actions that happen in 3 seconds and Something-Something-v2 [18] focuses on close-up camera view of person-object interactions. These dataset biases might hinders models trained on a single dataset to generalize and be used in a practical way. These challenges in action datasets make learning a general-purpose action model difficult. An ideal model should be able to cover a wide range of action classes meanwhile keeping the computation cost low. However, simply combining all these datasets to train a joint model does not lead to good performance [27]. In previous work [45], the authors have shown the benefit of training a joint model using multiple action datasets but their method requires large-scale image datasets such as ImageNet-21K [9] and JFT-3B [44], which is not available to the research community.

In this paper, we propose a general training paradigm for **Cross**-dataset training of **Ro**bust **a**ction recognition mo**d**els, ***CrossRoad***. Our method is designed to learn robust and informative feature representations in a principled way, using the informative loss for regularization. We do not assume the availability of large-scale image dataset pre-training (although one can certainly start with). Since there are intrinsic relations between different classes across different action datasets (See Fig. 1 for examples of similar classes from two datasets), we propose a projection loss to mine such relations such that the whole network is trained to avoid over-fitting to certain dataset biases. Finally, all proposed loss terms are weighted using learned parameters, so no hyper-parameter tuning is needed. Our empirical findings as shown in Table 1 indicate that our robust training method can consistently improve model backbone performance across multiple datasets. We show that our model can achieve competitive results compared to state-of-the-art methods, even without large-scale image dataset pre-training, and with lower computational cost.

The main contributions of this paper are three-fold:

- To our knowledge, this is the first work to introduce informative representation regularization into cross-dataset training for action recognition.
- We propose an effective approach to mine intrinsic class relations in cross-dataset training by introducing the projection loss.
- Our method requires negligible computation overhead during training and no additional computation during inference to the backbone network. Extensive experiments on various datasets suggest our method can consistently improve performance.

2 Related Work

CNNs and Vision Transformers. CNNs work as the standard backbones throughout computer vision tasks for image and video. Various effective convolutional neural architectures have been raised to improve the precision and efficiency (*e.g.*, VGG [34], ResNet [19] and DenseNet [20]). Although CNNs are still the primary models for computer vision, the Vision Transformers have already shown their enormous potential. Vision Transformer (ViT [10]) directly applies the architecture of

Transformer on image classification and get encouraging performance. ViT and its variants (*e.g.*, ViViT [2], TimesFormer [4], MViT [12], Swin [29], MTV [41]) achieve outstanding results in both image and video processing in recent years. These transformer-based modeling approaches have driven most of the recent advancements in the action recognition task. We focus on the training paradigm instead and study how training on various datasets can lead to robust general-purpose models.

Action Recognition/Classification. The research of action recognition has advanced with both new datasets and new models. The modern benchmarks for action recognition is the Kinetics dataset [21]. The Kinetics dataset proposes a large benchmark with more categories and more videos (*e.g.*, 400 categories 160,000 clips in [21] and 700 categories in [7]) as a harder benchmark compared to previous datasets like UCF-101 [36]. The Moments-in-Time [31] (MiT) dataset provides a million short video clips that covers 305 action categories. Note that it is impossible for Kinetics and MiT datasets to cover all the possible actions in all possible scales. For example, surveillance actions are missing in the two datasets. Many new approaches [39, 46, 28, 15, 42] have been carried out on these datasets, of which the SlowFast network [15] and MViT [12] obtain promising performance. We can see the trend of action recognition in the last two decades is to collect larger datasets (*e.g.*, Kinetics) and build models with a larger capacity.

Cross-dataset Training. Different datasets are constructed using different data sources (*e.g.*, movies, internet videos, and daily photography), labeling definitions (actions by a single person, actions between persons, and actions by a person with some objects). Thus, dataset bias and domain shift are inevitably involved. The domain shift hampers the generalization of the recognition model and restrict application feasibility. Several works [32, 8, 35] were proposed to tackle this issue. Previous works typically focused on the issue of domain adaption or transfer learning. However, the transferred models still suffer from problem of parameter-inefficiency, meaning that separate models are needed for different datasets. Larger datasets often deliver better results. Combining multiple datasets to boost data size, and improve the final performance [17], and the simultaneous use of multiple datasets is also likely to alleviate the damaging impact of dataset bias. OmniSource [11] utilizes web images as part of the training dataset to expand the diversity of the training data to reduce dataset bias. VATT [1] uses additional multi-modal data for self-supversied pretraining and finetunes on downstream datasets. CoVeR [45] combines image and video training even during the finetuning stage and reports significant performance boost compared to single-dataset training. Within each batch, CoVeR randomly samples from both image and video datasets and the sampling rate is proportional to the size of the datasets. PolyViT [27] further extends to training with image, video and audio datasets. Several sampling procedures including Task-by-Task, Alternating, Uniform task sampling, etc., are proposed to facilitate effective co-training. In this paper, we propose to utilize regularization methods and simple random sampling to fully leverage information across different datasets to produce general-purpose representations, without the use of any image or additional data from other modality.

3 Method

Our method is built upon the backbone of the Improved Multi-scale Vision Transformers (MViTv2) [26, 12]. Note that our approach works with any action recognition backbones. Given videos from multiple datasets during training, the model backbone takes the video frames and produces feature embeddings for each video. The same number of Multi-layer Perceptron (MLP) as the datasets are constructed as model heads to predict action classes for each dataset. To facilitate robust cross-dataset training, we propose two loss terms, namely, *the informative loss and projection loss*. The informative loss aims to maximize the embeddings' representative power. The projection loss, with the help of multiple cross-dataset projection layers, guides the model to learn intrinsic relations between classes of different dataset, hence the model heads can be trained jointly. See Fig. 1 for an overview of our framework. In this section, we first briefly describe the MViTv2 backbone design, and then present our proposed robust cross-dataset training paradigm.

3.1 The MViTv2 Backbone

Our model is based on the improved multi-scale vision transformers (MViTv2) [12, 26], which learns a hierarchy from dense (in space) and simple (in channels) to coarse and complex features. The series

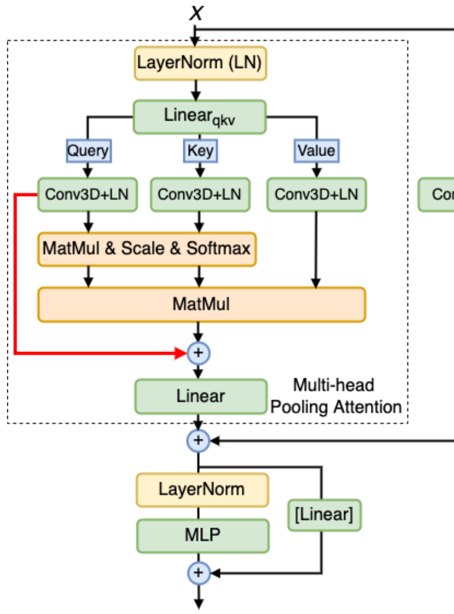

Figure 2: The MViTv2 Block. The residual connection for pooled query tensor (red arrow) and the residual 3D convolution operation outside the Multi-head Pooling Attention block are additions to the MViTv1 [12] design. The linear layer in the residual connection of the MLP block is only needed when the output embedding dimension is different. Compared to the MViTv2 paper [26], we do not use the decomposed relative embedding.

of work of vision transformers [10] (ViTs) follows the basic self-attention architecture [40] originally proposed for machine translation. In contrast to natural language which can be directly tokenized into words, given the input video $\mathbf{V} \in \mathbb{R}^{T \times H \times W \times 3}$, ViTs extract tokens by splitting the video into $N = \lfloor T/t \rfloor \times \lfloor H/h \rfloor \times \lfloor W/w \rfloor$ non-overlapping patches, $\{\mathbf{v_1}, \cdots, \mathbf{v_N} \in \mathbb{R}^{t \times h \times w}\}$. Each patch is then projected into a patch embedding by a 3D convolution operator E. All patch embeddings are then concatenated into a sequence, and separate learnable spatial-temporal positional embeddings $\mathbf{p_s}, \mathbf{p_t}$ are also added to this sequence. The patch embedding process is denoted by:

$$\mathbf{X_0} = [\mathbf{E}\mathbf{v_1} \cdots \mathbf{E}\mathbf{v_N}] + P(\mathbf{p_s}, \mathbf{p_t}) \in \mathbb{R}^{N \times d_p} \tag{1}$$

The P function extends the separate position embedding into the length of the sequence by repeating at the same spatial or temporal location. d_p is the dimension of the patch embedding.

The key component of the MViTv1 model [12] is the Multi Head Pooling Attention (MHPA), which pools the sequence of latent tensors to reduce the spatial or temporal dimension of the feature representations. In MViTv2 [26], a residual connection in MHPA for the pooled query tensor and a decomposed relative position embedding [1] are added. In this paper, we use 3D convolution as the pooling operation. Fig. 2 shows the detailed architecture of the MViTv2 block (our implementation). Each MViTv2 block consists of a multi-head pooling attention layer (MHPA) and a multi-layer perceptron (MLP), and the residual connections are built in each layer. The feature of each MViTv2 block is computed by:

$$\begin{aligned} X_1 &= \mathrm{MHPA}(LN(X)) + Pool(X) \\ Block(X) &= \mathrm{MLP}(LN(X_1)) + X_1 \end{aligned} \tag{2}$$

where X is the input tensor to each block. Multiple MViTv2 blocks are grouped into stages to reduce the spatial dimension while increase the channel dimension. The full backbone architecture is listed in supplementary material.

Classification head For the action recognition problem, the model produces C-class classification logits by first averaging the feature tensor from the last stage along the spatial-temporal dimensions

[1] We did not implement this part as the code was not available at the time of writing.

(we do not use the [CLASS] token in our transformer implementation), denoted as $\mathbf{z} \in \mathbb{R}^d$. A linear classification layer is then applied on the averaged feature tensor to produce the final output, $\mathbf{y} = \mathbf{W_{out}}\mathbf{z} \in \mathbb{R}^C$.

Pre-training and finetuning In the standard training paradigm for action recognition, models are pretrained using image datasets (ImageNet [9] or large-scale datasets like JFT-3B [44]) and then finetune on the target action recognition dataset. For CNN-based backbones, model weight inflation [21] is utilized to adapt the model trained on 2D image data to 3D video input. For transformer-based backbones, as the inputs are tokenized into a sequence, to adapt the model from image pretraining, the positional embeddings are interpolated to account for the additional temporal dimension before the finetuning.

Cross-dataset training paradigm In general, to facilitate cross-dataset training of K datasets, the same number of classification heads are appended to the feature embeddings. The k-th dataset classification output is defined as $\mathbf{Y}_k = h_k(\mathbf{Z}; \mathbf{W}_k) \in \mathbb{R}^{B \times C}$, where h_k could be a linear layer or a MLP and $\mathbf{W}_k$ is the layer parameter.

3.2 *CrossRoad*: Robust Cross-dataset Training

Our training process fully leverages different action recognition datasets by enforcing an **informative loss** to maximize the expressiveness of the feature embedding and a **projection loss** for each dataset that mines the intrinsic relations between classes across other datasets. We then use uncertainty to weight different loss terms without the need for any hyper-parameters.

Informative loss. Inspired by the recently proposed VICReg [3] and Barlow Twins [43] method for self-supervised learning in image recognition, we propose to utilize an informative loss function with two terms, **variance and covariance**, to maximize the expressiveness of each variable of the embedding. This loss is applied to each mini-batch, without the need for batch-wise nor feature-wise normalization. Given the feature embeddings of the mini-batch, $\mathbf{Z} \in \mathbb{R}^{B \times d}$, an expander (implemented as a two-layer MLP) maps the representations into an embedding space for the informative loss to be computed, denoted as $\mathbf{Z}' \in \mathbb{R}^{B \times d}$. The **variance loss** is computed using a hinge function and the standard deviation of each dimension of the embeddings by:

$$\mathcal{L}^v = \frac{1}{d} \sum_{j=1}^{d} \max(0, 1 - \sqrt{\frac{\sum (\mathbf{Z'}_{ij} - \bar{\mathbf{Z}}'_{:j})}{d-1} + \epsilon}) \tag{3}$$

Where : is a tensor slicing operation that extracts all elements from a dimension, and $\bar{\mathbf{Z}}'_{:j}$ is the mean over the mini-batch for j-th dimension. ϵ is a small scalar preventing numerical instabilities. With random sampling videos across multiple datasets for each batch, this criterion encourages the variance of each dimension in the embedding to be close to 1, preventing embedding collapse [43]. The **covariance loss** $c(\mathbf{Z}')$ is defined as:

$$\begin{aligned} C(\mathbf{Z}') &= \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{Z}'_i - \bar{\mathbf{Z}}')(\mathbf{Z}'_i - \bar{\mathbf{Z}}')^T \text{ , where } \bar{\mathbf{Z}}' = \frac{1}{n} \sum_{i=1}^{n} \bar{\mathbf{Z}}'_i \\ \mathcal{L}^c &= \frac{1}{d} \sum_{i \neq j} [C(\mathbf{Z}')]^2_{i,j} \end{aligned} \tag{4}$$

Inspired by VICReg [3] and Barlow Twins [43], we first compute the covariance matrix of the feature embeddings in the batch, $C(\mathbf{Z}')$, and then define the covariance term $\mathcal{L}^c$ as the sum of the squared off-diagonal coefficients of $C(\mathbf{Z}')$, scaled by a factor of $1/d$.

Projection Loss. In previous works [45, 27], the intrinsic relations between classes from across different datasets have been mostly ignored during training. We believe that samples in one dataset can be utilized to train the classification head of other datasets. As shown in Fig. 1, the "Clean and jerk" video sample from Kinetics can be considered as a positive sample for "Weightlifting" in Moments-in-Time as well (but not vice versa). Based on this intuition, we propose to add a directed projection layer for each pair of datasets for the model to learn such intrinsic relations. One can also initialize the projection using prior knowledge but it is out-of-scope for this paper. Given the output from the k-th dataset classification output, the projected classification output is defined as:

$$\mathbf{Y'}_k = \mathbf{Y}_k + \sum_{i \neq k}^{K-1} \mathbf{W}_{ik}^{proj} \mathbf{Y}_i \in \mathbb{R}^{C_k} \tag{5}$$

where C_k is the number of classes for the k-th dataset and $\mathbf{W}_{ik}^{proj}$ is the learned directed class projection weights from i-th to k-th dataset. In this paper we only consider a linear projection function. We then use the ground truth labels of the k-th dataset to compute standarad cross-entropy loss:

$$\mathcal{L}_k = -\sum_{c=1}^{C_k} \hat{\mathbf{Y}}_{k,c} \log(\mathbf{Y'}_{k,c}) \tag{6}$$

where $\hat{\mathbf{Y}}_{k,c}$ is the ground truth label for the c-th class from the k-th dataset.

Training. We jointly optimize the informative loss and the projection loss during cross-dataset training. To avoid tuning loss weights of different terms, we borrow the weighting scheme from multi-task learning [22] and define the overall objective function as:

$$\mathcal{L}(\sigma) = \mathcal{L}^v + \mathcal{L}^c + \sum_{k=1}^{K} \frac{1}{2\sigma_k^2} \mathcal{L}_k + \log \sigma_k \tag{7}$$

where σ is a vector of parameters of size K (the number of datasets) for each projection loss term.

4 Experiments

In this section, to demonstrate the efficacy of our training framework, we experiment on five action recognition datasets, including Kinetics-400 [21], Something-Something-v2 [18], Moments-in-Time [31], Activitynet [5] and Kinetics-700 [7]. The action recognition task is defined to be a classification task given a trimmed video clip. In the experiments, we aim to showcase that our method can achieve significant performance improvement with minimal computation overhead.

4.1 Experimental Setup

Datasets. We evaluate our method on five datasets. Kinetics-400 [21] (K400) consists of about 240K training videos and 20K validation videos in 400 human action classes. The videos are about 10 seconds long. Kinetics-700 [7] (K700) extends the action classes to 700 with 545K training and 35K validation videos. The Something-Something-v2 (SSv2) [18] dataset contains person-object interactions, which emphasizes temporal modeling. SSv2 includes 168K videos for training and 24K videos for evaluation on 174 action classes. The Moments-in-Time (MiT) dataset is one of the largest action dataset with 727K training and 30k validation videos. MiT videos are mostly short 3-second clips. The ActivityNet dataset [5] (ActNet) originally contains untrimmed videos with temporal annotations of 200 action classes. We cut the videos into 10-second long clips and split the dataset into 107K training and 16K testing. Following previous works [15, 45], we follow the standard dataset split and report top-1/top-5 classification accuracy on the test split for all datasets. We conduct two sets of experiments, namely K400 + MiT + SSv2 + ActNet and K700 + MiT + SSv2 + ActNet.

Implementation. Our backbone model utilizes MViTv2 as described in Section 3.1. Our models are trained from scratch with random initialization, without using any pre-training (same as in [15] and different from previous works [45, 27] that require large-scale image dataset pre-training like ImageNet-21K [9] or JFT-3B [44]). We follow standard dataset splits as previous works [26, 15, 41]. See more details in the supplementary material.

Baselines. PolyViT [27] utilizes multi-task learning on image, video and audio datasets to improve vision transformer performance. The backbone they used are based on ViT-ViViT [2]. Similarly, VATT [1] utilizes additional multi-modal data for self-supversied pretraining and finetunes on downstream datasets. The backbone network is based on ViT [10]. CoVER [45] is a recently proposed co-training method that includes training with images and videos simultaneously. Their model backbone is based on TimeSFormer [4]. We also compare our method with other recent models trained using large-scale image datasets. See Table 1 and Table 2 for the full list.

method	Training Data	gFLOPs	K400	MiT	SSv2	ActNet
ViViT [2]	+ IN-21K	3992	81.3 / 94.7	38.5 / 64.1	65.9 / 89.9	-
VidTr [25]	+ IN-21K	392	80.5 / 94.6	-	63.0 / -	-
TimeSFormer [4]	+ IN-21K	2380	80.7 / 94.7	-	62.4 / -	-
X3D-XXL [13]	+ IN-21K	194	80.4 / 94.6	-	-	-
MoViNet [24]	Scratch	386	81.5 / 95.3	40.2 / -	64.1 / 88.8	
MViT-B [12]	Scratch	455	81.2 / 95.1	-	67.7 / 90.9	-
MTV-B (320p) [41]	+ IN-21K	1116	82.4 / 95.2	**41.7** / 69.7	68.5 / 90.4	-
Video Swin [29]	+ IN-21K	2107	84.9 / 96.7	-	69.6 / -	-
MViTv2-L [26]	+ IN-21K	2828	**86.1 / 97.0**	-	**73.3 / 94.1**	-
MViTv2 w/o rel	Scratch	225	80.1 / -	-	-	-
Ours-baseline	Scratch	224	79.8 / 93.9	38.6 / 67.5	67.0 / 90.7	81.5 / 95.1
VATT [1]*	AudioSet + HowTo100M + Downstream	2483	82.1 / 95.5	41.1 / 67.7	-	-
CoVER [45]	IN-21K + K400 + SSv2 +MiT	2380	83.1 / -	41.3 / -	64.2 / -	-
PolyViT [27]	IN-1K + K400 + MiT + [Audio] + [Image]	3992	82.4 / 95.0	38.6 / 65.5	-	-
CrossRoad	K400 + SSv2 +MiT + ActNet	224	81.9 / 95.2	**41.7** / 71.0	68.9 / 91.6	87.4 / 97.3
CrossRoad(312p)	K400 + SSv2 +MiT + ActNet	614	83.2 / 96.4	**43.1 / 71.9**	69.3 / 92.1	88.2 / 97.6

Table 1: Comparison with state-of-the-art on Kinetics-400, Moments-in-Time, Something-something-v2 and ActivityNet. We divide the baselines into two groups based on whether they are parameter-efficient. We report top-1/top-5 accuracy for each dataset. The **bold** numbers and **underlined** are ranked first and second, respectively. "IN+21K" means ImageNet-21K dataset. The FLOPs computation is for a single video clip input. PolyViT [27] is trained jointly with multiple image, audio and video datasets. We list the larger ones. "*" is pretrained on AudioSet [16] and HowTo100M [30] in a self-supervised fashion and then finetuned on each downstream datasets, which results in separate models for each dataset.

4.2 Main Results

We summarize our method performance in Table 1 and Table 2. We train our model jointly on MiT, SSv2, ActNet and two version of the Kinetics datasets.

We first compare our method with the original MViTv2 backbone in Table 1. The "MViTv2 w/o rel" indicates the model without the relative positional embedding in the original paper. As we see, compared to our implementation, the performance difference is minor. The difference could be due to the small difference in the datasets (Kinetics videos are taking down from Youtube from the time of release. See supplementary material for full dataset statistics. We train our baseline model on the training set of each dataset to investigate the baseline performance. As we see, after adding robust joint training proposed in this paper, performance on each dataset has increased by 2.1%, 3.1%, 1.9% and 5.9% on K400, MiT, SSv2, ActivityNet, respectively in terms of top-1 accuracy. Note

method	Training Data	gFLOPs	K700	MiT	SSv2	ActNet
VidTr [25]	+ IN-21K	392	70.8 / -	-	63.0 / -	-
MoViNet [24]	Scratch	386	72.3 / -	-	-	-
MTV-B (320p) [41]	+ IN-21K	1116	75.2 / 91.7	41.7 / 69.7	68.5 / 90.4	-
MViT-v2 [26]	+ IN-21K	2828	**79.4 / 94.9**	-	**73.3 / 94.1**	-
Ours-baseline	Scratch	224	74.1 / 91.9	38.6 / 67.5	67.0 / 90.7	81.5 / 95.1
VATT [1]*	AudioSet + HowTo100M + Downstream	2483	72.7 / -	41.1 / 67.7	-	-
CoVER [45]	IN-21K + K700 + SSv2 +MiT	2380	74.9 / -	41.5 / -	64.7 / -	-
CrossRoad	K700 + SSv2 +MiT + ActNet	224	75.8 / 93.2	<u>42.2</u> / <u>72.3</u>	69.1 / 92.2	88.1 / 97.2
CrossRoad(312p)	K700 + SSv2 +MiT + ActNet	614	<u>76.3</u> / <u>93.5</u>	**43.5 / 73.0**	<u>70.4</u> / <u>93.1</u>	89.1 / 98.1

Table 2: Comparison with state-of-the-art on Kinetics-700, Moments-in-Time, Something-something-v2 and ActivityNet. The **bold** numbers and **<u>underlined</u>** are ranked first and second, respectively. See text and caption in Table 1 for details.

that our method achieves such improvement withtout large-scale image pre-training and additional computational cost.

We then compare our method with state-of-the-art on these datasets. We train a higher resolution model with larger spatial inputs (312p) and achieves better performance compared to recent cross-dataset training methods, CoVER [45] and PolyVit [27], on Kinetics-400, and significantly better on MiT and SSv2, as shown in Table 1. Note that our model does not use any image training datasets, and our model computation cost is only a fraction of the baselines. We also show that our performance boost does not come from the additional training dataset of ActivityNet in Table 3.

Our method also achieves competitive results compared to state-of-the-art models trained with large-scale image dataset (ImageNet-21K [9]). Compared to a recent method, MTV-B [41], our method is able to achieve significantly better top-1 accuracy across Kinetics-400, MiT, SSv2 by 0.8%, 1.4%, 0.8%, respectively, at half of the computation cost and without large-scale pre-training. Note that our model is parameter-efficient, while multiple MTV-B models need to be trained and tested on these datasets separately. Our method can achieve better performance with a deeper base backbone or larger resolution inputs but we have not tested due to limitation of computation resources.

We then compare our method on the Kinetics-700, MiT, SSv2 and ActivityNet training with baselines. Our parameter-efficient model can achieve better performance than MTV-B [41] at one-fifth of the computation cost. With a larger resolution model at 312p, we achieves significantly better performance than the baseline across Kinetics-400, MiT, SSv2 by 2.2%, 4.9%, 3.4%, respectively.

4.3 Ablation Experiments

In this section, we perform ablation studies on the K400 set. To understand how action models can benefit from our training method, we explore the following questions (results are shown in Table 3):

Does our proposed robust loss help? We compare our model training with vanilla cross dataset training, where multiple classification heads are attached to the same backbone and the model is trained simply with cross-entropy loss. The vanilla model is trained from a K400 checkpoint as ours. As shown in Table 3, we try training the vanilla model with both the same training schedule as ours and a 4x longer schedule. As we see, there is a significant gap between the overall performance of the vanilla model and ours, validating the efficacy of our proposed method. Also, longer training schedule

method	Training Data	K400	MiT	SSv2	ActNet
CrossRoad	K400 + SSv2 + MiT + ActNet	81.9 / 95.2	41.7 / 71.0	68.9 / 91.6	87.4 / 97.3
Vanilla (50 ep)	K400 + SSv2 + MiT + ActNet	80.1/ 94.0	33.4 / 60.1	60.8 / 89.0	86.5 / 97.1
Vanilla (200 ep)	K400 + SSv2 + MiT + ActNet	80.6 / 94.7	35.1 / 63.9	56.8 / 85.3	86.3 / 97.2
- Informative Loss	K400 + SSv2 + MiT + ActNet	13.5 / 33.4	7.3 / 19.9	9.7 / 28.5	24.8 / 54.3
- Projection Loss	K400 + SSv2 + MiT + ActNet	80.6 / 94.8	39.9 / 69.2	61.5 / 88.0	86.9 / 97.5
- ActNet	K400 + SSv2 + MiT	81.4 / 95.0	41.3 / 70.5	68.7 / 91.3	-

Table 3: Ablation experiments. We investigate the effectiveness of each component of our method as well as compare to vanilla multi-dataset training method. The numbers are top-1/top-5 accuracy, respectively.

does not lead to better performance on some datasets, including SSv2, suggesting vanilla cross-dataset training is unstable. In terms of performance on ActivityNet, we observe that both training methods achieve good results, which might be because ActivityNet classes are highly overlapped with Kinetics-400 (65 out of 200).

How important is the informative loss? We then experiment with removing the informative loss (Section 3.2) during cross-dataset training. It seems that the feature embedding of the model collapse and the model is not trained at all.

How important is the projection loss? We then experiment with removing the projection heads (Section 3.2) during cross-dataset training. The model is trained with the original cross-entropy loss and the informative loss. As shown in Table 3, the performance on MiT and SSv2 suffers by a large margin, indicating that the projection design helps boost training by better utilizing cross-dataset information.

Does the additional ActivityNet data help? In previous methods like CoVER and PolyViT, the ActivityNet dataset has not been used. In this experiment, we investigate the important of the ActivityNet dataset by removing it from the training set. From Table 3, we can see that the performance across all datasets drop by a small margin, indicating our superior results compared to CoVER (see Table 1 and Table 2) come from the proposed robust training paradigm rather than the additional data.

4.4 Discussion

By cross-dataset training transformers on various datasets, we obtain competitive results on multiple action datasets, without large-scale image datasets pre-training. Our method, ***CrossRoad***, is parameter-efficient and does not require hyper-parameter tuning. Current limitations of our experiments are that we have not tried co-training with image datasets such as ImageNet-21K [9]. Hence we do not know how much performance gain that would entail. We plan to explore this in future work. In addition, we have not tried training larger model with FLOPs on par with state-of-the-art or other backbone architectures (e.g. CNNs) due to limitation of our computational resources. Hence we are not sure how our algorithm would behave with these models. Although our model is trained on multiple datasets, potential dataset biases can still cause negative societal impact in real-world deployment, as the datasets we have do not fully represent all aspects of human actions.

5 Conclusion

In this paper, we present ***CrossRoad***, a robust cross-dataset training approach that maximizes information content of representation and learns intrinsic relations between individual datasets. Our method can train parameter-efficient models that perform well across multiple datasets.

References

[1] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *Advances in Neural Information Processing Systems*, 34, 2021.

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846, 2021.

[3] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding. *arXiv preprint arXiv:2102.05095*, 2(3):4, 2021.

[5] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

[6] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, 2017.

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

[8] Min-Hung Chen, Zsolt Kira, Ghassan AlRegib, Jaekwon Yoo, Ruxin Chen, and Jian Zheng. Temporal attentive alignment for large-scale video domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6321–6330, 2019.

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

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

[11] Haodong Duan, Yue Zhao, Yuanjun Xiong, Wentao Liu, and Dahua Lin. Omni-sourced webly-supervised learning for video recognition. In *European Conference on Computer Vision*, pages 670–688. Springer, 2020.

[12] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021.

[13] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 203–213, 2020.

[14] Christoph Feichtenhofer, Axel Pinz, and Richard P. Wildes. Spatiotemporal residual networks for video action recognition. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *NeurIPS*, 2016.

[15] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019.

[16] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 776–780. IEEE, 2017.

[17] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12046–12055, 2019.

[18] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[20] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

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

[22] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[23] Alexander Kläser, Marcin Marszalek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In Mark Everingham, Chris J. Needham, and Roberto Fraile, editors, *BMVC*, 2008.

[24] Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong. Movinets: Mobile video networks for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16020–16030, 2021.

[25] Xinyu Li, Yanyi Zhang, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. Vidtr: Video transformer without convolutions. *arXiv e-prints*, pages arXiv–2104, 2021.

[26] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Improved multiscale vision transformers for classification and detection. *arXiv preprint arXiv:2112.01526*, 2021.

[27] Valerii Likhosherstov, Anurag Arnab, Krzysztof Choromanski, Mario Lucic, Yi Tay, Adrian Weller, and Mostafa Dehghani. Polyvit: Co-training vision transformers on images, videos and audio. *arXiv preprint arXiv:2111.12993*, 2021.

[28] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, 2019.

[29] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021.

[30] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640, 2019.

[31] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(2):502–508, 2019.

[32] Jonathan Munro and Dima Damen. Multi-modal domain adaptation for fine-grained action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 122–132, 2020.

[33] Gunnar A Sigurdsson, Santosh Divvala, Ali Farhadi, and Abhinav Gupta. Asynchronous temporal fields for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 585–594, 2017.

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[35] Xiaolin Song, Sicheng Zhao, Jingyu Yang, Huanjing Yue, Pengfei Xu, Runbo Hu, and Hua Chai. Spatio-temporal contrastive domain adaptation for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9787–9795, June 2021.

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

[37] Graham W. Taylor, Rob Fergus, Yann LeCun, and Christoph Bregler. Convolutional learning of spatio-temporal features. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *ECCV*, 2010.

[38] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[39] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[41] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. *arXiv preprint arXiv:2201.04288*, 2022.

[42] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *CVPR*, 2020.

[43] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

[44] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. *CoRR*, abs/2106.04560, 2021. URL `https://arxiv.org/abs/2106.04560`.

[45] Bowen Zhang, Jiahui Yu, Christopher Fifty, Wei Han, Andrew M Dai, Ruoming Pang, and Fei Sha. Co-training transformer with videos and images improves action recognition. *arXiv preprint arXiv:2112.07175*, 2021.

[46] Yue Zhao, Yuanjun Xiong, and Dahua Lin. Trajectory convolution for action recognition. In *NeurIPS*, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 4.4
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 4.4
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]

(b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We plan to open-source our code upon paper acceptance. We have provided detailed implementation instructions in the main text and in the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4 and the supplemental material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We follow standard data splits and evaluation protocol as previous works. See Section 4.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See supplemental material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No] We used open-source (Apache-2.0 licensed) MViTv2 code and standard public datasets
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See Section 4.4
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]