LoKi: Low-dimensional KAN for Efficient Fine-tuning Image Models

Xuan Cai, Renjie Pan, Hua Yang*
Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University, China
Shanghai Key Lab of Digital Media Processing and Transmission, China
China MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China

Abstract

'Pre-training + fine-tuning' has been widely used in various downstream tasks. Parameter-efficient fine-tuning (PEFT) has demonstrated higher efficiency and promising performance compared to traditional full-tuning. The widely used adapter-based and prompt-based methods in PEFT can be uniformly represented as adding an MLP structure to the pre-trained model. These methods are prone to over-fitting in downstream tasks, due to the difference in data scale and distribution. To address this issue, we propose a new adapter-based PEFT module, i.e., LoKi, which consists of an encoder, a learnable activation layer, and a decoder. To maintain the simplicity of LoKi, we use single-layer linear networks for the encoder and decoder, and for the learnable activation layer, we use a Kolmogorov-Arnold Network (KAN) with the minimal number of layers (only 2 KAN linear layers). With a bottleneck rate much lower than that of Adapter, LoKi is equipped with fewer parameters (only half of Adapter) and eliminates slow training speed and high memory usage of KAN. We conduct extensive experiments on LoKi under image classification and video action recognition across 9 datasets. LoKi demonstrates highly competitive generalization performance compared to other PEFT methods with fewer tunable parameters, ensuring both effectiveness and efficiency.

1. Introduction

The rise of pre-trained models has transformed the landscape of computer vision [15, 38, 61], with models like Vision Transformers (ViT) [15] achieving state-of-the-art performance across various tasks. The widely used paradigm of Pre-training + Fine-tuning (PT+FT) has become a cornerstone for transferring these fundamental models to various downstream tasks [1, 3, 29, 33, 34, 54, 59, 61, 82], which leverages the strong generalized representations of pre-trained models for generalization. As shown in Fig. 1,

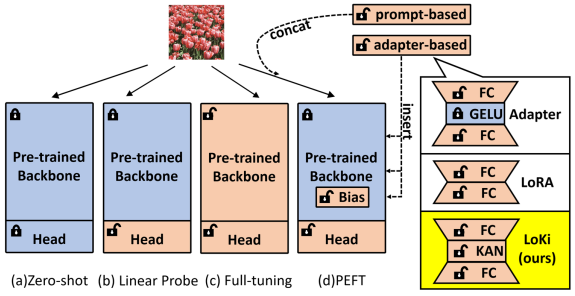

Figure 1. Common fine-tuning methods of pre-trained models.The "lock mark" indicates frozen during training. Compared to Adapter and LoRA, LoKi introduces trainable activation with nonlinear fitting capabilities.

apart from simple strategies to evaluate vision backbones i.e., zero-shot and linear probe, existing PT+FT methods can be roughly categorized to two types: (1) Full-tuning [3, 10, 17, 54, 61, 65] trains the entire model in an end-to-end manner, requiring a relatively large computation costs. (2) Parameter-Efficient Fine-Tuning (PEFT) [30, 31, 44, 67, 72, 87] methods such as Adapter [30], LoRA [31], and prompt-based tuning [44], have shown promising results by freezing the majority of the pre-trained model while only training a small set of parameters. These methods achieve strong performance while reducing computational costs compared to full-tuning. Among them, adapter-based and prompt-based approaches can be uniformly represented as a Multi-Layer Perceptron (MLP) structure [26], which uses the universal approximation theorem to fit nonlinear functions. Existing studies typically increase the number of layers and neurons in MLP for performance improvement, which adds to more parameters. Meanwhile, regularization is utilized to prevent over-fitting by driving some of the parameters toward zero. This creates a trade-off between adding parameters and reducing their impact. Balancing this requires hyper-parameter tuning and model selection, which undoubtedly increases training costs [7, 32, 63, 66], especially for downstream tasks. A fine-tuning method with excellent generalization should perform equally well across different pre-trained weights or datasets without additional hyperparameter adjustments, even using exactly the same

*corresponding author

training strategies for fine-tuning. We reproduced the shortcomings of classic methods such as Adapter and LoRA in this regard in Sec. 4.

In summary, despite the success of PEFT methods, the issues of MLP within them can be attributed to (1) reliance on fixed activation, which require combination with linear layers to have the ability to fit non-linear functions; (2) high bottleneck ratios and a reliance on fixed activation functions, which limit their ability to generalize well in these challenging scenarios.

To achieve better generalization in downstream tasks, while preserving the plug-and-play feature, we propose a new adapter-based module LoKi by (1) introducing a ***learnable*** activation layer with a tunable parameter ratio between the activation and linear layers, and (2) using linear layers for dimension mapping to facilitate a lower bottleneck ratio. In this way, the activation in LoKi can be disentangled from linear layers, making it more flexible and less sensitive. To maintain simplicity, the encoder-decoder architecture of LoKi is implemented with single-layer linear networks, while a minimal Kolmogorov-Arnold Network (KAN) is set as the activation with only two KAN linear layers. The encoder provides dense feature representation for the activation layer, which, through its learnable parameters, is capable of better fitting complex nonlinear functions. The decoder then ensures that the output feature dimensions align with the pre-trained model, facilitating seamless integration. Compared to Adapter, LoKi has a lower bottleneck ratio, fewer parameters, and avoids the slow training speed and high memory consumption of KAN. We conduct experiments on image classification and video action recognition tasks using pre-trained vision models. LoKi achieves highly competitive performance among six common fine-tuning methods (Adapter, LoRA, Prompt Tuning, Full-tuning, Linear Probe, Bias Tuning) on seven datasets. It is noteworthy that on datasets with distinct distribution, LoKi's advantage is even more pronounced. Due to a more reasonable distribution of parameters, LoKi achieves a balance between the parameter amounts and memory consumption (Adapter's parameters are primarily concentrated in the linear layers, whereas KAN allocates its parameters to the activation functions.). The contribution of this paper can be summarized as follows:

- We propose LoKi, a novel adapter-based module with learnable activation layers and tunable parameter ratios, addressing the limitations of traditional MLP-based PEFT methods.
- We design a lightweight architecture combining single-layer linear networks with minimal Kolmogorov-Arnold Network (KAN), achieving better nonlinear function fitting with fewer parameters.
- Extensive experiments demonstrate that LoKi achieves superior generalization performance across various datasets and pre-trained models without additional hyper-parameter tuning.

2. Related Work

Image pre-trained models ViT and its variants [8, 14, 15, 53, 70, 78, 86] are the mainstream backbones for pre-trained models in CV, achieving sota performance. Pre-trained models are typically trained on large-scale labeled datasets and then used as initializations for downstream tasks, where they are fine-tuned on new datasets. Due to the high cost of annotation, self-supervised learning is increasingly being applied to pre-trained models [9, 12, 24, 27, 76, 81], providing them with massive amounts of data and thereby enhancing their performance. In multi-modal models, ViT is also commonly used as the image branch [37, 46, 47, 61, 77, 79, 84]. As a result, there are now many image pre-trained models with strong representation capabilities that use ViT as their backbone. In this paper, we take ViT-B/16 [15] as our backbone and conduct tests on the fine-tuning modules using both CLIP (ViT-B/16) [61] and ViT(IN21k) [15] pre-trained weights.

PEFT Among the fine-tuning methods based on pre-trained models, PEFT is an important research direction, as it freezes most of the pre-trained model to reduce time and memory consumption in downstream tasks. Adapter [30, 51] inserts bottleneck structure of MLPs into the pre-trained model, and training these new parameters can achieve the performance of full-tuning. LoRA [31] separates the weight changes from the original weights and performs low-rank decomposition on the weight changes. Mathematically, the decomposed low-rank matrix is equivalent to the MLP without the activation layer. Prompt tuning [44, 48] does not add new parameters to the backbone but instead concatenates learnable parameters in the input. These methods can be uniformly represented as adding an MLP module to the model [26]. LST [69] adds a side network alongside the pre-trained model and only train this side network. This reduces memory consumption but also results in lower accuracy compared to other methods like LoRA. There are also methods [22, 68, 87] that do not add new parameters but instead select a small subset of parameters from the pre-trained model for training. For example, BitFit [87] only trains the bias terms of the model, FAR [72] and FishMask [67] dynamically select some parameters for fine-tuning. Recently, there has been work [64] that considers PEFT as a reconstruction and expansion of the parameter subspace. In visual tasks, some works [2, 11, 33, 88, 89] have applied PEFT from the NLP field to CV and achieved strong performance. However, these efforts conduct PEFT within the same modality as the pre-trained models, meaning image pre-trained models undergo PEFT for image tasks, and language models are fine-tuned

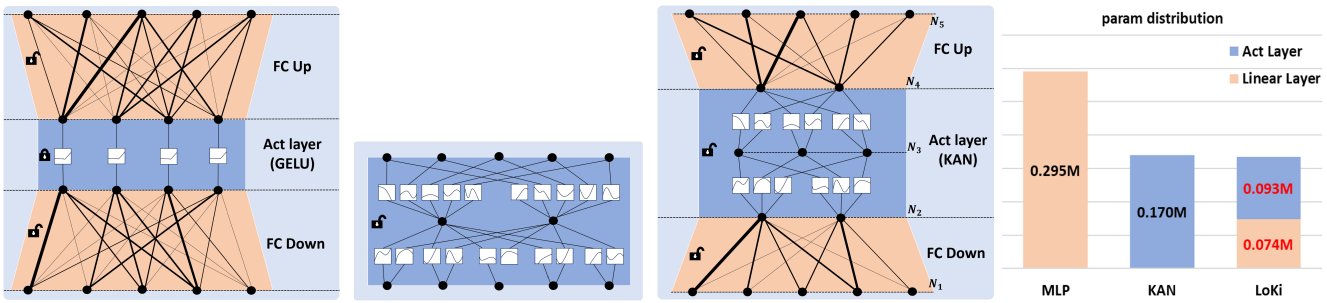

Figure 2. Module structure and parameter distribution. In MLP, parameters are concentrated in the linear layers, while in KAN, parameters are focused on the activation functions. In LoKi, parameters are present in both the linear and activation layers, and the parameter ratio can be changed based on different configurations of the LoKi module.In subfigure (c), N_i represents the number of nodes in that layer. The parameter distribution graph shows the quantity of parameters in individual Adapter, KAN, and LoKi modules used in Sec. 4.

for language tasks. Due to the lack of temporal modeling in image models, cross-modal PEFT applications from image models to video tasks are rare. Mainstream video models [1, 3, 10, 17, 25, 35, 54, 65, 73, 80], although they use weights from image pre-trained models, typically undergo full-tuning to achieve better performance, which usually requires hundreds to thousands of training epochs, with training costs much higher than fine-tuning for image tasks. Some works [52, 58, 74, 82] have begun to attempt PEFT on image pre-trained models for video tasks, mostly using Adapter, and have achieved performance exceeding traditional video task models.

KAN applied to mainstream models With the introduction of KAN [55], many efforts have been made to integrate KAN into mainstream models to enhance performance. Some works [6, 18, 19, 23, 28, 45, 71, 83] attempt to directly replace parts of models with KAN and its variants. KAT [83] proposed the use of group KAN to replace the MLP in Transformer. In UKAN [45], KAN is encapsulated as a Tokanized KAN Block to replace parts of the convolutional network in U-Net [62]. Both have achieved promising performance. There are also studies [85] that have compared KAN and MLP on multiple tasks, suggesting that under similar amounts of parameters or computational resources, KAN's performance does not match that of MLP. Our work introduces a new PEFT module that includes KAN, which does not alter the model itself, thus avoiding extensive computations. It is applied in a suitable scenario. To our knowledge, there is currently no such work.

3. Methodology

3.1. Structure of LoKi

LoKi consists of an encoder, a learnable activation layer, and a decoder. To maintain the simplicity of LoKi, we use single-layer linear networks for the encoder and decoder, corresponding to FC Down and FC Up in Fig. 2(c). The learnable activation layer, referring to the Kolmogorov-Arnold representation theorem [5, 39], utilizes a 2-layer KAN network, corresponding to Act layer in Fig. 2(c). The encoder provides a more dense feature representation for the activation layer, thereby reducing the number of parameters and computational load in the activation layer. The activation layer contains a considerable number of learnable parameters, which allow the nonlinear functions within the activation layer to better fit the target nonlinear functions. Finally, the decoder ensures that the feature dimensions of LoKi are consistent with the pre-trained model, allowing it to return to the computation of the pre-trained model. This can be mathematically expressed as:

$$\mathrm{LoKi}(\mathbf{X}) = (\mathbf{W}_1 \circ \mathbf{\Phi}_1 \circ \mathbf{\Phi}_0 \circ \mathbf{W}_0)\,\mathbf{X}, \tag{1}$$

where $\mathbf{W}_0$ and $\mathbf{W}_1$ represent linear transformations, corresponding to FC Down and FC Up in Fig. 2(c), respectively. $\circ$ signifies the composite function. $\mathbf{\Phi}_0$ and $\mathbf{\Phi}_1$ signify the nonlinear transformation, which is the sum of a set of 1D functions. Specifically, $\mathbf{\Phi}_i$ is expressed as:

$$\mathbf{\Phi_i} = \{\phi_{q,p}\}, \quad p = 1, \cdots, n_{\text{in}}, \quad q = 1, \cdots, n_{\text{out}}, \tag{2}$$

In Eq. (2), $\phi_{q,p}$ has learnable parameters, and $\mathbf{\Phi}_i$ is the sum of $\mathbf{n}_{\text{in}} \times \mathbf{n}_{\text{out}}$ instances of $\phi_{q,p}$. In Eq. (1), $\mathbf{\Phi}_0$ and $\mathbf{\Phi}_1$ have $\mathbf{n}_{\text{in}}$ and $\mathbf{n}_{\text{out}}$ corresponding to N_2,N_3 and N_3,N_4 in Fig. 2(c), respectively. More specifically, $\phi_{q,p}$ is the sum of a set of B-splines and a SiLU activation function, which can be expressed as Eq. (3), where c_j is optimizable and j is a hyperparameter.

$$\phi_{q,p}(x) = x/(1 + e^{-x}) + \sum_j c_j B_j(x) \tag{3}$$

In Fig. 2(c), we list the number of nodes in each layer of LoKi, denoted by N_1 to N_5. Here, N_1=N_5=D, where D is the dimension of the hidden layer in the pre-trained model. For simplicity, N_2 and N_4 are set to the same size, although they can be different in actual use. N_2=N_4=$r_1 \times N_1$, with r_1 representing the scaling ratio of the encoder and decoder.

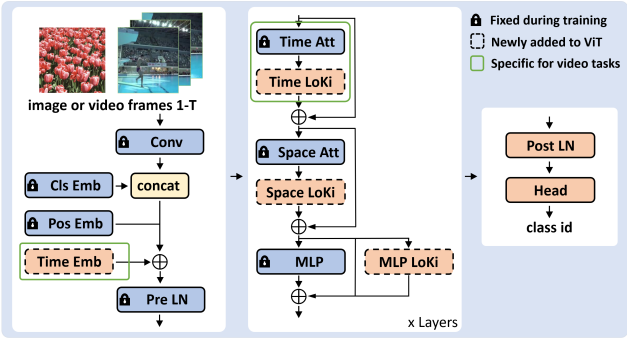

Figure 3. The model architecture that incorporates LoKi into ViT. In video tasks, to accommodate the modeling of video frames in the temporal dimension, we additionally incorporate temporal modules, as shown within the red box in the figure.

N_3=$r_2 \times N_2$, where r_2 is the scaling ratio of the activation function network. According to the Kolmogorov-Arnold representation theorem, N_3=$2N_2$+1. The number of parameters N_L in a single LoKi's linear part is:

$$N_L = 2r_1 N_1(N_1 + 1) \tag{4}$$

The number of parameters N_A in activation part is:

$$N_A = 2r_1^2 r_2 N_1^2(G + K + 3) + r_1 N_1(r_2 + 1) \tag{5}$$

where G denotes the number of spline intervals and K denotes the order of the spline, they are hyperparameters. In the tests in Sec. 4, we follow [85] to set G=4 and K=3. The ratio of parameters in the linear layer to the activation layer can be flexibly adjusted through r_1 and r_2. We conduct ablation experiments to study the values of r_1 and r_2 in Sec. 4. The structural and parameter distribution comparison between LoKi, MLP, and KAN is shown in Fig. 2. The parameter count calculation for MLPs can refer to Eq. (4). At a similar level of accuracy, r_1 in the MLP is typically set to no less than 0.5,while in LoKi it can be set to 0.0625 or lower. Therefore, even if there are no parameters in the activation layer of the MLP, its parameter count is still significantly higher than LoKi's. The calculation method for the number of parameters in KAN can refer to Eq. (5). For a fair comparison, we adjusted the number of parameters in KAN to be similar to that of LoKi.

3.2. The insertion position of LoKi

Referring to the method of Adapter, the position where LoKi is placed in ViT is shown in Fig. 3. For image tasks, we place LoKi behind the self-attention layer and alongside the MLP. For video tasks, we additionally incorporate a temporal module, which consists of Time Attention that shares parameters with Space Attention and Time LoKi. In the input, Time embeddings required for Time Attention are also needed. For the attention calculation on the temporal dimension of video frames, simply adjust the order of the data dimensions from $(BT, N+1, D)$ to $(B(N+1), T, D)$. The rest of the calculations will follow the order of the $(BT, N+1, D)$ dimension, just as in ViT, where B represents the batch size, T represents the number of video frames, N represents the number of patches, D represents the dimensionality of the hidden layer in ViT.

4. Experiments

Datasets We evaluate LoKi on two fundamental tasks. For image classification, seven datasets are validated: Cifar10, Cifar100 [40], Food101 [4], MNIST [42], DDSM [43], DTD [13], and FGVC Aircraft [56]. For video action recognition, we use HMDB51 [41] and Diving48 [49], as they focus on spatial modeling and temporal modeling, respectively. It is noteworthy that all the datasets are relatively small and are prone to overfitting when used with large pre-trained models.

Experimental Setup In image classification, we use both ViT(IN21K) and CLIP(ViT-B/16) pre-trained weights. On each dataset, we measure the accuracy of the fine-tuning method by the average top-1 acc obtained on the two pre-trained weights. In typical classification tasks, the generalization performance of methods is often measured solely by the average accuracy across multiple datasets. However, we measure the generalization of the method by the difference in top-1 accuracy between the two pre-trained weights. Obviously, the smaller the difference, the more stable and, consequently, the better the generalization of the method. This setting is able to provide a more comprehensive assessment of LoKi. Based on the zero-shot results of the CLIP model, as seen in Tab. 1, we consider datasets with an accuracy greater than 50% to have a data distribution more aligned with that of commonly used pre-trained models. For these datasets, we fine-tune for 5 epochs. For datasets with an accuracy less than 50%, we fine-tune for 50 epochs. We conduct experiments using Full-tuning, Linear Probe, Bias Tuning, Prompt Tuning, LoRA, Adapter, and LoKi, all under the same training strategy. As a comparison with LoKi, we also experiment with the vanilla KAN, which has a similar number of parameters to LoKi and is used like an adapter module. In video action recognition, we use CLIP(ViT-B/16). For HMDB51, we compare LoKi with other finetuning methods, and for the Diving48, we compare LoKi with state-of-the-art baselines. In both tasks, LoKi adopts the setting of r_1=0.0625 and N_3=$2N_2$+1 (i.e., $r_2 \approx 2$).

4.1. Results on Image Classification Task

The comparison of accuracy and generalization among fine-tuning methods on image classification is shown in Tab. 1. We arrange the datasets in order of decreasing zero-shot

Type	Method	Tunable Params(M)	Mem (G)	CIFAR 10	Food 101	CIFAR 100	MNIST	DTD	FGVC Aircraft	DDSM	Average
				avg[acc on IN21K , acc on CLIP] (abs[acc on IN21K - acc on CLIP])							
-	Zero-shot	-	-	91.6	89.2	68.7	56.0	46.0	27.1	9.2	-
	Full-tuning	86	5.9	70.0 (53.6)	47.0 (75.0)	52.8 (67.9)	91.9 (15.0)	38.4 (48.9)	40.7 (67.3)	53.1 (4.7)	56.2 (47.5)
	Linear Probe	<1	2.4	94.9 (3.0)	87.9 (7.1)	80.8 (7.2)	95.7 (2.8)	73.4 (5.6)	47.2 (19.8)	56.6 (1.6)	76.6 (6.7)
PEFT	Bias Tuning [87]	<1	4.5	98.4 (0.7)	90.3 (4.8)	89.9 (3.9)	99.0 (0.4)	76.9 (4.5)	69.8 (11.0)	67.4 (5.6)	84.5 (4.4)
	Prompt Tuning [33]	<1	6.4	97.6 (0.7)	88.7 (6.6)	87.8 (4.2)	98.3 (0.8)	73.6 (2.4)	63.0 (20.9)	61.1 (2.4)	81.4 (5.4)
	LoRA [31]	<1	4.7	97.7 (1.6)	87.3 (1.0)	87.4 (5.3)	98.9 (0.3)	74.9 (5.2)	49.4 (29.5)	39.8 (3.3)	76.5 (6.6)
	Adapter [30]	7	4.8	86.7 (24.1)	63.3 (53.2)	65.1 (55.3)	98.6 (1.8)	47.8 (53.4)	46.5 (65.5)	67.5 (16.1)	67.9 (38.5)
	KAN [55]	4	32.9	80.2 (37.3)	56.3 (65.5)	60.6 (63.7)	97.3 (3.7)	48.5 (52.7)	43.7 (52.7)	67.0 (8.9)	64.8 (40.6)
	LoKi	4	10.4	98.1 (0.0)	89.4 (4.9)	89.1 (1.5)	99.3 (0.0)	76.0 (3.8)	71.6 (4.5)	74.2 (1.4)	85.4 (2.3)

Table 1. Comparison of accuracy and generalization among different fine-tuning methods on various image datasets. The zero-shot accuracy is reported by [61] (except that DDSM is reproduced in our own), which is used to measure data distribution discrepancies between the downstream dataset and the pre-trained data. In the corresponding results for each dataset, the number on top represents the **average** top-1 acc obtained using IN21K and CLIP pre-trained weights. The number in parentheses below represents the **absolute difference** in top-1 acc between the two pre-trained weights, which measures the generalization ability. Red represents the best results, and blue represents the second-best results.

Type	Method	Pretrain	Param (M)	Tunable Params(M)	Top1 acc	Top5 acc	Views
Full-tuning	TimeSformer [3]	IN21K	121	121	74.9	-	32×1×3
	Video Swin-B [54]	IN21K	88	88	84.0	-	32×1×3
	BEVT [75]	IN1K+K400	88	88	87.2	-	32×1×3
	ORViT [29]	IN21K	160	160	88.0	-	32×1×1
PEFT	DUALPATH [60]	CLIP	96	10	88.7	-	32×1×3
	AIM-ViT-B/16 [82]	CLIP	97	11	77.4 83.0 88.9	97.8 98.4 -	8×1×1* 16×1×1* 32×1×1
	Loki(ViT)	CLIP	92	6	78.4 85.7 89.5	98.2 99.0 99.0	8×1×1 16×1×1 32×1×1

Table 2. Results on Diving48. Views = #frames × #clips × #spatial. "*" indicates that the accuracy from this view was reproduced by the official source code.

accuracy according to the CLIP model. Each fine-tuning method's corresponding results for each dataset consist of two rows of numbers, with the number on top represents the average top-1 acc obtained using IN21K and CLIP pre-trained weights, which is used to measure the accuracy of the method. The number in parentheses below represents the difference in top-1 acc between the two pre-trained weights, which is used to measure the generalization of the method. The specific accuracy of the fine-tuning methods on different weights is presented in the appendix. LoKi has the highest average accuracy and the lowest accuracy variance across multiple datasets, which is the best on the average of seven datasets. Since Adapter and KAN encompass a similar number of parameters with LoKi, both have lower average accuracy due to their larger accuracy variance. Through the analysis of the results on each weight, it is observed that the accuracy on the CLIP weights is not satisfactory. In Fig. 4, we present the training and validation curves of Adapter, KAN, and LoKi on the CLIP weights fine-tuning across three datasets. On DDSM, none of the three baselines show signs of overfitting. With a more reasonable distribution of parameters in its structure, LoKi, despite having the fewest parameters (56.7% of Adapter's and 99.2% of KAN), achieves more than 10% higher accuracy than Adapter and KAN, proving that LoKi has sufficient fitting capability. It's worth noting that KAN consumes 33G of memory, which is significantly more than other methods. On DTD, Adapter exhibits severe overfitting, with the train acc reaching above 98% during the training process, while the val acc is only 21.1%. At the 12th epoch of training, the

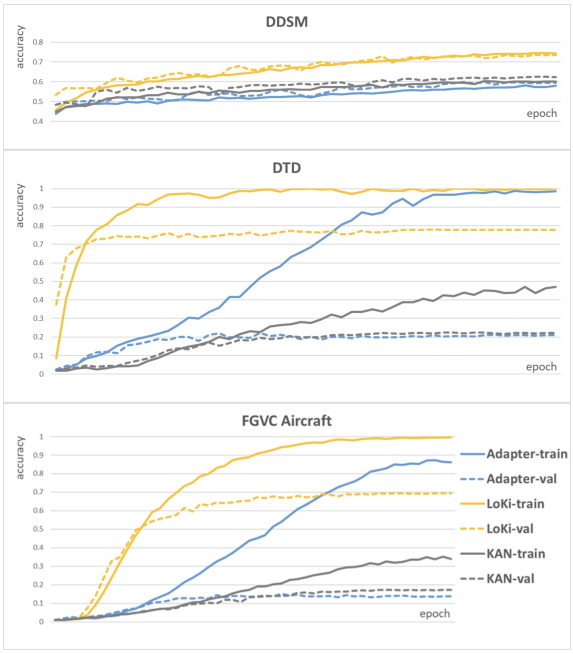

Figure 4. Training and validation set accuracy of LoKi, Adapter, and KAN on DDSM, DTD, and FGVC Aircraft based on CLIP(ViT-B/16).

train acc is 30.5%, and the val acc is 21.0%. Afterwards, although the train acc gradually rises to 98%, the val acc remains around 21%. KAN shows a lack of adaptation to the learning rate and insufficient fitting capability on this dataset, with the final train acc stuck around 47%, and the val acc at 22.1%. KAN's overfitting situation is slightly better than Adapter's. In contrast, LoKi demonstrates superior generalization by converging faster and achieving better performance. On FGVC Aircraft, the convergence of the three baselines is similar to that on DTD. We speculate that the poor generalization of Full-tuning, Adapter, and KAN is due to their modification of a larger number of parameters.

We further visualize the attention maps of several methods on FGVC fine-tuned with ViT(IN21K) pre-trained weights, as shown in Fig. 5, which demonstrates the effectiveness of LoKi qualitatively. Since Linear Probe modifies parameters after the last attention layer, the attention map for this method is derived from the original ViT(IN21K) weights, serving as a baseline. It fails to distinguish between different types of aircraft and is heavily affected by background interference. Full-tuning significantly changes the parameters, resulting in an attention map that can focus more on the aircraft itself. Other PEFT methods based on ViT(IN21K), show only minor improvements over the baseline and are not able to accurately focus on the aircraft. LoKi achieves a much concentrated attention than all the other methods since it is attentive to distinctive parts of the aircraft such as the tail and engines with minimal background interference.

Type	Method	Tunable Params(M)	Top1 acc
-	Full-tuning	86	59.4
	Linear Probe	0.1	61.2
PEFT	VPT ViT-B/16 [33]	7	62.4
	AdaptFormer [11]	8	63.7
	Pro-tuning [57]	9	63.3
	VideoPrompt [34]	6	66.4
	ST-Adapter [58, 60]	7	65.9
	LoKi(ViT)	6	73.3

Table 3. Results on HMDB51.

Methods	Tunable Params(M)	Top1 acc
Frozen Space & MLP	<0.1	23.2
Full-tuning	85.8	75.7
Frozen Space & MLP + Space LoKi	0.5	79.3
+ Time LoKi	0.9	86.4
+ MLP LoKi	1.4	87.6

Table 4. Plug-and-play effectiveness of LoKi. LoKi maximizes the capacity of frozen space. Even only adding Space LoKi results in better performance than Full-tuning.

In general, on image classification task, LoKi demonstrates strong generalization and excellent fitting capabilities. Particularly on datasets with large distribution differences, LoKi's overall performance is superior. On the datasets with the greatest distribution differences, DDSM and FGVC Aircraft, LoKi's accuracy and generalization are the best. On the DTD dataset, LoKi ranks second in both accuracy and generalization, as shown in Tab. 1.

4.2. Results on Video Action Recognition Task

The results of LoKi on Diving48 are illustrated in Tab. 2. LoKi(ViT) encompasses the fewest tunable parameters and, among the models reported to use ViT-B/16 as the backbone, it has the highest accuracy. To further compare with the state-of-the-art Adapter-based method AIM, we reproduce AIM on $8\times1\times1$ and $16\times1\times1$ views with the official code. LoKi(ViT) only contains 56.9% tunable parameters of AIM's, and the top-1 accuracy is higher than AIM in all three views. Since the backbone parameter scale of these two models are significantly higher than LoKi(ViT)'s, we did not include them in the comparison results. The experimental results for HMDB51 are illustrated in Tab. 3. For a fair comparison, we listed the results of models that also use ViT-B/16 as the backbone. LoKi(ViT)'s accuracy is higher than other fine-tuning methods, indicating that LoKi is capable to video datasets that emphasize spatial modeling.

4.3. Ablation Study of LoKi

Effectiveness of LoKi We take frozen ViT and the full-tuning ViT as two baselines. In frozen ViT, only the classification head and the last Layer Norm layer are open for training. By gradually adding Space LoKi, Time LoKi, and MLP LoKi to the first baseline, we validate LoKi on Diving48 with CLIP(ViT-B/16). As shown in Tab. 4, due to

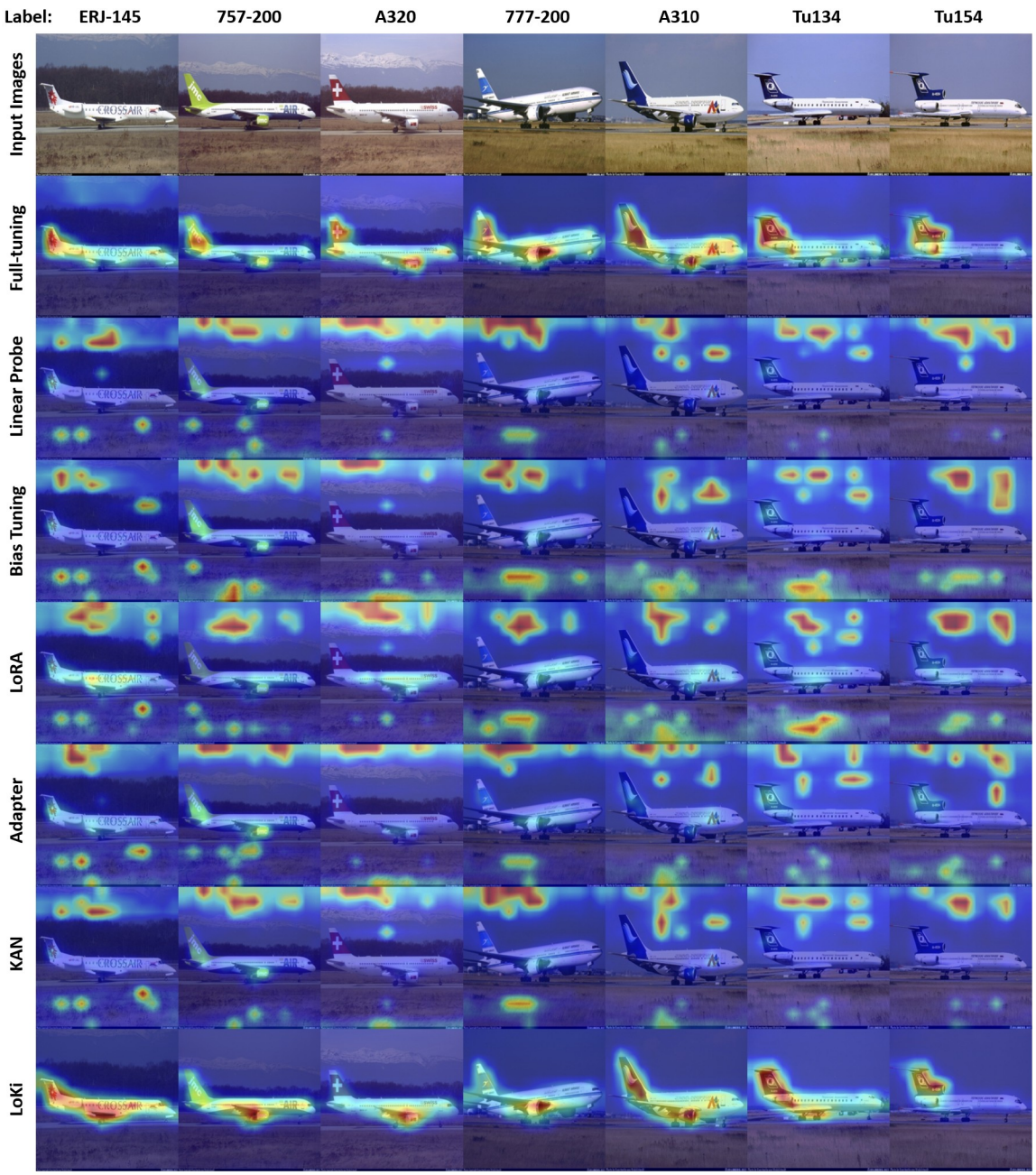

Figure 5. Attention map visualizations of fine-tuning methods on FGVC Aircraft. LoKi focuses on key parts of aircrafts such as the tail and engines, with minimal background interference. Full-tuning and other methods still allocate some redundant attention to the background.

the limited capacity of the classification head and Layer Norm layer, the accuracy of this baseline is only 23.2%, significantly lower than the fully fine-tuned method, but the number of tunable parameters is also much fewer, less than 0.1% of full-tuning. On the fully-tuning ViT, it has 85.8M trainable parameters and requires multiple experiments to find a suitable learning rate. When using the same learning rate as other control groups, its training is prone to collapse. The final result is 75.7%, after struggling with learning rate modification. Space LoKi has only 0.6% of the trainable parameters of full-tuning, but the performance is already 3.6% higher than full-tuning. Since Diving48 focuses on fine-grained temporal modeling, the accuracy increases by 7.1% after adding Time LoKi. This model has 1% of the tunable parameters of full-tuning, with an accuracy 10.7% higher. Further adding MLP LoKi enhances the performance, achieving an accuracy of 87.6%. Based on the results of this experiment, we add Space LoKi and MLP LoKi to the image tasks, and additionally incorporate Time LoKi in the video tasks.

Shape	Tunable Params(M)	Top1 acc
(768, [16, 33, 33, 33, 33, 16], 768)	2.5	83.3
(768, [16, 33, 33, 33, 16], 768)	2.1	84.6
(768, [16, 33, 33, 16], 768)	1.7	86.0
(768, [16, 33, 16], 768)	1.4	87.6

Table 5. Study of LoKi depth. All numbers in the column "Shape" represent the number of neurons in each layer of LoKi, with elements in parentheses indicating linear layers and elements in square brackets indicating KAN linear layers.

Shape	Tunable Params(M)	Mem (G)	Top1 acc
(768, [8, 17, 8], 768)	0.6	9.6	85.9
(768, [16, 33, 16], 768)	1.4	11.4	87.6
(768, [32, 65, 32], 768)	3.4	15.3	88.3
(768, [48, 97, 48], 768)	6.1	18.6	89.5
(768, [64, 129, 64], 768)	9.6	22.3	88.6

Table 6. Effect of bottleneck ratio r_1 of LoKi. "Shape" represents the same as in Tab. 5.

Shape	Tunable Params(M)	Top1 acc
(768, [16, 4, 16], 768)	1.0	82.9
(768, [16, 8, 16], 768)	1.1	83.4
(768, [16, 16, 16], 768)	1.2	86.2
(768, [16, 33, 16], 768)	1.4	87.6
(768, [16, 64, 16], 768)	1.7	86.4

Table 7. Effect of bottleneck ratio r_2 of LoKi. "Shape" represents the same as in Tab. 5.

Shape of LoKi We conduct experiments on the shape of LoKi's activation layer to explore an appropriate scale for application in subsequent experiments. Specifically, we conduct experiments on Diving48 with $32\times1\times1$ views. The results of the LoKi's depth are illustrated in Tab. 5. As the depth of the LoKi module increases, its number of parameters increases while the accuracy decreases. We speculate that a deeper LoKi can fit more complex nonlinear models, but it would also be more difficult to converge, and in our typical encounters, the LoKi with the minimum number of layers is sufficient to handle the task. Since KAN includes at least 2 layers of B-Spline functions, which corresponds to 3 layers of neurons, $[16, 33, 16]$ corresponds to the minimum number of layers in LoKi's activation layer. The adjustment of LoKi's width involves two hyperparameters, r_1 and r_2, as described in Sec. 3.1. The effect of r_1 is shown in Tab. 6. Increasing r_1 can lead to an improvement in accuracy, with the highest accuracy of 89.5% achieved when the width is $(768, [48, 97, 48], 768)$. Subsequently, adding more neurons to the activation layer of LoKi results in a decrease in accuracy. The effect of r_2 is shown in Tab. 7. Its results are similar to those of r_1, with accuracy increasing and then decreasing as r_2 grows. Ultimately, we chose the ratio specified by the Kolmogorov-Arnold representation theorem, which is 2n+1, reflected in the shape as the ratio of 16 to 33.

Dataset	AIM	LoKi(ViT)	Views
K400 [36]	83.9	80.0	8x3x1
SSv2 [21]	66.4	56.3	8x1x3

Table 8. Comparisons on K400 & SSV2 with AIM.

5. Discussion

We also test on larger datasets K400 [36] and SSv2 [21]. K400 contains about 240K training videos and about 20K validation videos across 400 classes. SSv2 contains about 169K training videos and about 25K validation videos across 174 classes. LoKi did not achieve accuracy similar to Adapter, as illustrated in Tab. 8. Training on large datasets is less prone to overfitting, and in this case, Adapter has an advantage over LoKi. However, such a large amount of data is sufficient for training pre-trained models, it is not suitable for fine-tuning models. For instance, [1, 16, 20, 50, 54] directly use the K400 dataset for pre-training. In typical industrial scenarios, organizing and training such a large amount of data is costly, so we believe that the application prospects of LoKi will be broader than those of Adapter.

LoKi generates very few new model weights for each downstream task. In image classification and video action recognition tasks, there are only 4M and 6M different parameters between different datasets, accounting for 5% and 7% of ViT-B/16, respectively. When fine-tuning for multiple downstream tasks, this will significantly reduce the storage of model weights and is also beneficial for communication-efficient distributed learning and privacy-protecting federated learning applications.

6. Conclusion

LoKi, as a new PEFT module, exhibits excellent generalization in downstream tasks with small datasets and significant differences in data distribution, and its accuracy is similar to or better than other PEFT methods. In terms of explainability, LoKi's attention map is clearer than other methods. LoKi is suitable for small datasets, is not prone to overfitting, and has strong adaptability to training strategies and pre-trained weights. These characteristics align well with the current situation where downstream tasks often involve small data volumes and significant differences in data distribution.

Acknowledgements. This research was partly supported by grants of National Natural Science Foundation of China (NSFC, Grant No. 62171281), Science and Technology Commission of Shanghai Municipality (STCSM, Grant Nos. 20DZ1200203, 2021SHZDZX0102, 22DZ2229005).

References

[1] 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. 1, 3, 8

[2] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022. 2

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, page 4, 2021. 1, 3, 5

[4] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part VI 13*, pages 446–461. Springer, 2014. 4

[5] Jürgen Braun and Michael Griebel. On a constructive proof of kolmogorov's superposition theorem. *Constructive approximation*, 30:653–675, 2009. 3

[6] Roman Bresson, Giannis Nikolentzos, George Panagopoulos, Michail Chatzianastasis, Jun Pang, and Michalis Vazirgiannis. Kagnns: Kolmogorov-arnold networks meet graph learning. *arXiv preprint arXiv:2406.18380*, 2024. 3

[7] Jason Brownlee. *Better deep learning: train faster, reduce overfitting, and make better predictions*. Machine learning mastery, 2018. 1

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. 2020. 2

[9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. 2021. 2

[10] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 1, 3

[11] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022. 2, 6

[12] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv e-prints*, 2021. 2

[13] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014. 4

[14] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. 2021. 2

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, and Sylvain Gelly. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 1, 2

[16] 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. 8

[17] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 1, 3

[18] Remi Genet and Hugo Inzirillo. A temporal kolmogorov-arnold transformer for time series forecasting. *arXiv preprint arXiv:2406.02486*, 2024. 3

[19] Remi Genet and Hugo Inzirillo. Tkan: Temporal kolmogorov-arnold networks. *arXiv preprint arXiv:2405.07344*, 2024. 3

[20] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens Van Der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A single model for many visual modalities. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16102–16112, 2022. 8

[21] 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. 8

[22] Demi Guo, Alexander M Rush, and Yoon Kim. Parameter-efficient transfer learning with diff pruning. *arXiv preprint arXiv:2012.07463*, 2020. 2

[23] Xiao Han, Xinfeng Zhang, Yiling Wu, Zhenduo Zhang, and Zhe Wu. Kan4tsf: Are kan and kan-based models effective for time series forecasting? *arXiv preprint arXiv:2408.11306*, 2024. 3

[24] Yufei Han, Haoyuan Chen, Linwei Yao, Kuan Li, and Jianping Yin. Mat-vit:a vision transformer with mae-based self-supervised auxiliary task for medical image classification. In *2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2024. 2

[25] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. 3

[26] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*, 2021. 1, 2

[27] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. 2021. 2

[28] Yunhong He, Yifeng Xie, Zhengqing Yuan, and Lichao Sun. Mlp-kan: Unifying deep representation and function learning. *arXiv preprint arXiv:2410.03027*, 2024. 3

[29] Roei Herzig, Elad Ben-Avraham, Karttikeya Mangalam, Amir Bar, Gal Chechik, Anna Rohrbach, Trevor Darrell, and

Amir Globerson. Object-region video transformers. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 3148–3159, 2022. 1, 5

[30] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019. 1, 2, 5

[31] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 1, 2, 5

[32] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 646–661. Springer, 2016. 1

[33] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer, 2022. 1, 2, 5, 6

[34] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *European Conference on Computer Vision*, pages 105–124. Springer, 2022. 1, 6

[35] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. 3

[36] 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. 8

[37] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. 2021. 2

[38] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, and Wan Yen Lo. Segment anything. *IEEE*, 2023. 1

[39] Andre Nikolaevich Kolmogorov. *On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables*. American Mathematical Society, 1961. 3

[40] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 4

[41] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International conference on computer vision*, pages 2556–2563. IEEE, 2011. 4

[42] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 4

[43] Rebecca Sawyer Lee, Francisco Gimenez, Assaf Hoogi, Kanae Kawai Miyake, Mia Gorovoy, and Daniel L Rubin. A curated mammography data set for use in computer-aided detection and diagnosis research. *Scientific data*, 4(1):1–9, 2017. 4

[44] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. 2021. 1, 2

[45] Chenxin Li, Xinyu Liu, Wuyang Li, Cheng Wang, Hengyu Liu, and Yixuan Yuan. U-kan makes strong backbone for medical image segmentation and generation. *arXiv preprint arXiv:2406.02918*, 2024. 3

[46] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. Align before fuse: Vision and language representation learning with momentum distillation. 2021. 2

[47] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. 2022. 2

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

[49] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 513–528, 2018. 4

[50] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4804–4814, 2022. 8

[51] Zhaojiang Lin, Andrea Madotto, and Pascale Fung. Exploring versatile generative language model via parameter-efficient transfer learning. 2020. 2

[52] Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard De Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen clip models are efficient video learners. In *European Conference on Computer Vision*, pages 388–404. Springer, 2022. 3

[53] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. 2021. 2

[54] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3202–3211, 2022. 1, 3, 5, 8

[55] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*, 2024. 3, 5

[56] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 4

[57] Xing Nie, Bolin Ni, Jianlong Chang, Gaofeng Meng, Chunlei Huo, Shiming Xiang, and Qi Tian. Pro-tuning: Unified

prompt tuning for vision tasks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023. 6

[58] Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. St-adapter: Parameter-efficient image-to-video transfer learning. *Advances in Neural Information Processing Systems*, 35:26462–26477, 2022. 3, 6

[59] Renjie Pan, Jihao Dong, and Hua Yang. Discovering clone negatives via adaptive contrastive learning for image-text matching. In *The Thirteenth International Conference on Learning Representations*. 1

[60] Jungin Park, Jiyoung Lee, and Kwanghoon Sohn. Dual-path adaptation from image to video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2203–2213, 2023. 5, 6

[61] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, and Jack Clark. Learning transferable visual models from natural language supervision. 2021. 1, 2, 5

[62] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015. 3

[63] Shaeke Salman and Xiuwen Liu. Overfitting mechanism and avoidance in deep neural networks. *arXiv preprint arXiv:1901.06566*, 2019. 1

[64] Chongjie Si, Xiaokang Yang, and Wei Shen. See further for parameter efficient fine-tuning by standing on the shoulders of decomposition. *arXiv preprint arXiv:2407.05417*, 2024. 2

[65] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014. 1, 3

[66] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 1

[67] Yi-Lin Sung, Varun Nair, and Colin A Raffel. Training neural networks with fixed sparse masks. *Advances in Neural Information Processing Systems*, 34:24193–24205, 2021. 1, 2

[68] Yi-Lin Sung, Varun Nair, and Colin A Raffel. Training neural networks with fixed sparse masks. *Advances in Neural Information Processing Systems*, 34:24193–24205, 2021. 2

[69] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Lst: Ladder side-tuning for parameter and memory efficient transfer learning. *Advances in Neural Information Processing Systems*, 35:12991–13005, 2022. 2

[70] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, and Yinxiao Li. Maxvit: Multi-axis vision transformer. *arXiv e-prints*, 2022. 2

[71] Cristian J Vaca-Rubio, Luis Blanco, Roberto Pereira, and Màrius Caus. Kolmogorov-arnold networks (kans) for time series analysis. *arXiv preprint arXiv:2405.08790*, 2024. 3

[72] Danilo Vucetic, Mohammadreza Tayaranian, Maryam Ziaeefard, James J Clark, Brett H Meyer, and Warren J Gross. Efficient fine-tuning of bert models on the edge. In *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1838–1842. IEEE, 2022. 1, 2

[73] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016. 3

[74] Mengmeng Wang, Jiazheng Xing, Boyuan Jiang, Jun Chen, Jianbiao Mei, Xingxing Zuo, Guang Dai, Jingdong Wang, and Yong Liu. A multimodal, multi-task adapting framework for video action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5517–5525, 2024. 3

[75] Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiyang Dai, Mengchen Liu, Yu-Gang Jiang, Luowei Zhou, and Lu Yuan. Bevt: Bert pretraining of video transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14733–14743, 2022. 5

[76] Shaoru Wang, Jin Gao, Zeming Li, Jian Sun, and Weiming Hu. A closer look at self-supervised lightweight vision transformers. 2022. 2

[77] Wenhui Wang, Hangbo Bao, Li Dong, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. 2021. 2

[78] Wenhai Wang, Enze Xie, Xiang Li, Deng Ping Fan, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. 2021. 2

[79] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022. 2

[80] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321, 2018. 3

[81] Zhenda Xie, Yutong Lin, Zhuliang Yao, Zheng Zhang, and Han Hu. Self-supervised learning with swin transformers. 2021. 2

[82] Taojiannan Yang, Yi Zhu, Yusheng Xie, Aston Zhang, Chen Chen, and Mu Li. Aim: Adapting image models for efficient video action recognition. *arXiv preprint arXiv:2302.03024*, 2023. 1, 3, 5

[83] Xingyi Yang and Xinchao Wang. Kolmogorov-arnold transformer. *arXiv preprint arXiv:2409.10594*, 2024. 3

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

[85] Runpeng Yu, Weihao Yu, and Xinchao Wang. Kan or mlp: A fairer comparison. *arXiv preprint arXiv:2407.16674*, 2024. 3, 4

[86] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Francis Eh Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. 2021. 2

[87] Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*, 2021. 1, 2, 5

[88] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. 2022. 2

[89] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, (9):130, 2022. 2