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

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

The supplementary materials are structured as follows:

§A provides the implementation details of the two tasks in this work.
§B demonstrates more visualization results of LoKi on several fine-grained datasets.
§C provides the pseudo-code and the training process of LoKi.

A. Implementation Details

Image Classification We uniformly set the data preprocessing for training to random cropping and random horizontal flipping. The batch size is set to 32, and we use the AdamW optimizer with a learning rate of 4×10^{-4}, employing a learning rate schedule that decreases according to a cosine curve down to 10^{-4}. On each dataset, all methods require training a new classification head.

Among the baselines, Full-tuning involves setting all parameters to be trainable, whereas Linear Probe only trains the classification head parameters. In Bias Tuning, we follow the approach in [5] to train all the bias terms. For Prompt Tuning, we adopt the method in [1] by incorporating 200 tokens with learnable parameters into the input. In LoRA, we add matrices A and B with a rank of 4 next to the W_q, W_k, W_v, and W_o matrices in Attention Layers, and set their dropout ratio to 0.2. In the Adapter method, we follow the approach from [4], setting the bottleneck ratio for all Adapters to 0.25, with a shape of $(768, 192, 768)$. The Adapters with a residual connection is concatenated after every Attention Layer. The Adapters that are paralleled alongside each MLP do not have residual connections, and the scale ratio for these Adapters is set to 2.0. For KAN and LoKi, we simply replace the Adapter with KAN and LoKi, respectively, while keeping all other parameters identical. To maintain consistent parameter counts for KAN and LoKi, the size of KAN is set to $(768, 11, 768)$.

*corresponding author

We test these methods on seven datasets using ViT(IN21K) and CLIP(ViT-B/16) weights, respectively. The comparison of accuracy are presented in Tab. 1 and Tab. 2. In Tab. 2, we additionally annotate the changes in accuracy on CLIP(ViT-B/16) compared to fine-tuning on ViT(IN21K) within parentheses, which are consistent with the results in parentheses in Tab.1 of the main text.

Video Action Recognition We set the batch size to 4 in LoKi(ViT), with all other settings identical to those for image tasks. For our reproduction of the AIM model [4] at $8 \times 1 \times 1$ and $16 \times 1 \times 1$ views, we maintain all preprocessing and training strategies consistent with the official source code. The accuracy of other models is derived from their respective papers. We conducted experiments on the order of Space Attention and Time Attention discussed in TimeSformer and ViViT. Our experimental results indicate that the order of the temporal module and the spatial module has no significant impact on the outcome, which is consistent with the conclusion in ViViT.

B. Visualization

In Fig.5 of the main text, all methods except for Full-tuning and LoKi are severely interfered with by the background. We additionally selected images from the FGVC Aircraft dataset with complex backgrounds to demonstrate their attention maps. These images' background have targets that the original pre-trained weights focused on or colors similar to the airplane, as shown in Fig. 1. LoKi's attention map is hardly affected by the background interference. When there is a salient target in the image, LoKi's attention will focus on that target, even though there are smaller similar targets around. We demonstrate this in Fig. 2, in the original image, there are some small airplanes at the top, with the attention map focused on the salient target. We take a screenshot of the original image, in the sub-image, oil tanks have a similar shape to the fuselage, and some airplanes have only partial tails and engines visible. LoKi's attention map can

Method	CIFAR10	Food101	CIFAR100	MNIST	DTD	FGVC Aircraft	DDSM	Average
Full-tuning	96.8	84.5	86.7	99.4	62.8	74.3	55.4	80.0
Linear Probe	96.4	84.3	84.4	94.3	70.6	37.3	55.8	74.7
Bias Tuning	98.7	87.9	91.8	98.8	74.6	64.3	64.6	83.0
Prompt Tuning	97.9	85.4	89.9	97.9	72.4	52.5	59.9	79.4
LoRA	98.5	87.8	90.0	99.0	72.3	64.1	38.1	78.5
Adapter	98.7	**89.9**	**92.7**	**99.5**	74.5	**79.2**	**75.5**	**87.1**
KAN	**98.8**	89.0	92.4	99.1	**74.8**	70.0	71.4	85.1
LoKi	98.1	86.9	89.8	99.3	74.1	73.8	74.9	85.3

Table 1. Comparison of Different Methods Fine-Tuned on ViT(IN21K).

Method	CIFAR10	Food101	CIFAR100	MNIST	DTD	FGVC Aircraft	DDSM	Average
Full-tuning	43.2 (↓53.6)	9.5 (↓75.0)	18.8 (↓67.9)	84.4 (↓15.0)	13.9 (↓48.9)	7.0 (↓67.3)	50.7 (↓4.7)	32.5
Linear Probe	93.4 (↓3.0)	91.4 (↑7.1)	77.2 (↓7.2)	97.1 (↑2.8)	76.2 (↑5.6)	57.1 (↑19.8)	57.4 (↑1.6)	78.5
Bias Tuning	98.0 (↓0.7)	92.7 (↑4.8)	87.9 (↓3.9)	99.2 (↑0.4)	**79.1** (↑4.5)	**75.3** (↑11.0)	70.2 (↑5.6)	**86.1**
Prompt Tuning	97.2 (↓0.7)	**92.0** (↑6.6)	85.7 (↓4.2)	98.7 (↑0.8)	74.8 (↑2.4)	73.4 (↑20.9)	62.3 (↑2.4)	83.4
LoRA	96.9 (↓1.6)	86.8 (↓1.0)	84.7 (↓5.3)	98.7 (↓0.3)	77.5 (↑5.2)	34.6 (↓29.5)	41.4 (↑3.3)	74.4
Adapter	74.6 (↓24.1)	36.7 (↓53.2)	37.4 (↓55.3)	97.7 (↓1.8)	21.1 (↓53.4)	13.7 (↓65.5)	59.4 (↓16.1)	48.7
KAN	61.5 (↓37.3)	23.5 (↓65.5)	28.7 (↓63.7)	95.4 (↓3.7)	22.1 (↓52.7)	17.3 (↓52.7)	62.5 (↓8.9)	44.4
LoKi	**98.1** (0.0)	91.8 (↑4.9)	**88.3** (↓1.5)	**99.3** (0.0)	77.9 (↑3.8)	69.3 (↓4.5)	**73.5** (↓1.4)	85.5

Table 2. Comparison of Different Methods Fine-Tuned on CLIP(ViT-B/16).

mostly accurately focus on the fuselage, with a slight influence from the oil tanks. We also demonstrated images with multiple small targets, as shown in Fig. 3, where LoKi's attention can concentrate on each small target and clearly distinguish them with few omissions, and it is not easily distracted by the background, even if the objects in the background (such as colored smoke) are much larger than the small targets. We display the attention maps on other datasets in Fig. 4. LoKi is able to maintain its focus on the targets. We train our model using low-resolution images (32×32-pixel) and test it on high-resolution images (Stanford Dogs Dataset [2] and CUB-200-2011 Dataset [3]), even so, the attention maps still maintain good resolution.

C. Pseudo-code of the Adapted ViT Block

LoKi is very simple to implement, and can be plugged into pre-trained models. We illustrate the basic components of LoKi and how to apply LoKi to ViT using PyTorch-style pseudo-code. r_1 and r_2 are hyperparameters corresponding to Sec.3 of the main text. Just like the Adapter, LoKi uses a serial insertion method in the attention block and a parallel way in the MLP block.

```
Algorithm: LoKi

# Input:
# x: input features (B, N+1, dim)
# dim: feature dimension
# r1, r2: expansion ratios for LoKi
# num_head: number of attention heads
# t: temperature parameter
# B: batch size
# N: sequence length

# Model Parameters Initialization
encoder_weights = initialize(dim, dim*r1)
decoder_weights = initialize(dim*r1, dim)
kan_weights = initialize(
                  dim*r1, dim*r1*r2, dim*r1)

#Loading pre-trained weights
attention_weights = load(weights)
mlp_weights = load(weights)
norm_weights = load(weights)
for x in loader:  # Process each batch

    # 1. Attention Branch
    # Layer Normalization
    x1 = layernorm(x, norm_weights)

    # Multi-head Attention
    # (B, N+1, dim)
    attn_out = attention(x1, attention_weights)

    # LoKi Processing for Attention
    a = fc(attn_out, encoder_weights) # Encode
    a = kan_activation(a, kan_weights)# Activate
    a_loki = fc(a, decoder_weights)   # Decode

    # Residual Connection
    x = x + a_loki  # (B, N+1, dim)
```

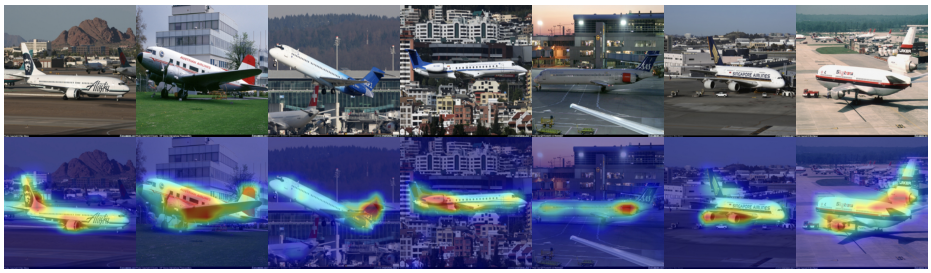

Figure 1. Attention map with complex backgrounds.

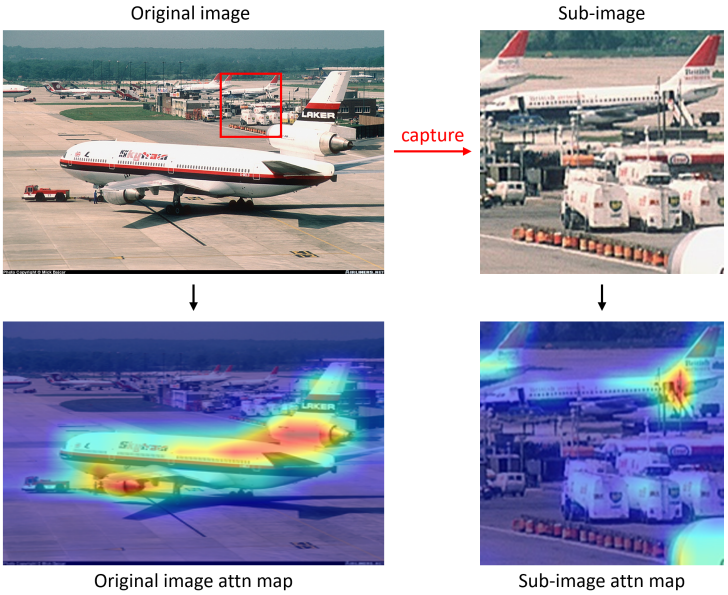

Figure 2. Attention map of salient objects and similar items.

```
    # 2. MLP Branch
    # Layer Normalization
    x2 = layernorm(x, norm_weights)

    # MLP Processing
    # (B, N+1, dim)
    mlp_out = mlp(x2, mlp_weights)

    # LoKi Processing for MLP
    m = fc(x2, encoder_weights)        # Encode
    m = kan_activation(m, kan_weights)# Activate
    m_loki = fc(m, decoder_weights)    # Decode

    # Residual Connections
    # (B, N+1, dim)
    output = x + mlp_out + m_loki

    # 3. Loss Computation
    loss = criterion(output, targets)

    # 4. Parameter Updates
    gradients = compute_gradients(loss)
    update_parameters(learning_rate, gradients)

# Output:
# output: transformed features (B, N+1, dim)
# loss: training loss value
```

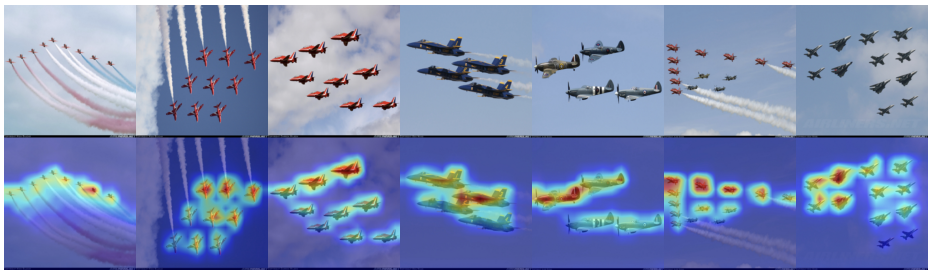

Figure 3. Attention map with multiple identical small targets

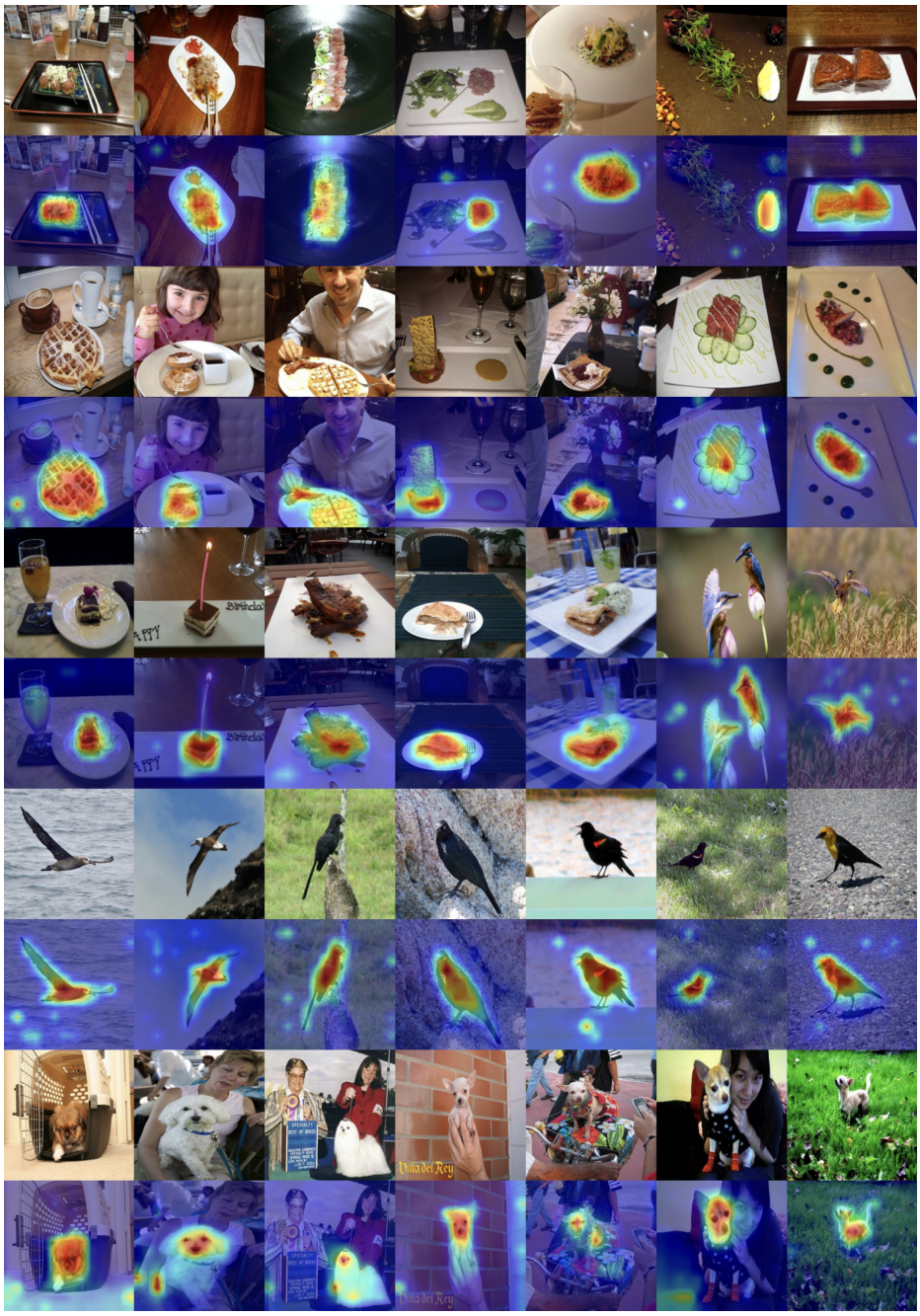

Figure 4. Attention maps on other datasets. The images of birds and dogs are both trained on low-resolution datasets (32×32-pixel) and tested on high-resolution datasets.

References

[1] 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] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*. Citeseer, 2011. 2

[3] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 2

[4] 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

[5] 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