Dual-Kernel Adapter: Expanding Spatial Horizons for Data-Constrained Medical Image Analysis

Ziquan Zhu [a], Hanruo Zhu [a], · Si-Yuan Lu [c], · Xiang Li [d], Yanda Meng [e], Gaojie Jin [b], · Lu Yin [f], · Lijie Hu [g], · Di Wang [f], · Lu Liu [b], Tianjin Huang [b,h]

a. University of Leicester, Leicester, UK b. University of Exeter, Exeter, UK c. Nanjing University of Posts and Telecommunications, Nanjing, China d. University of Bristol, Bristol, UK e. King Abdullah University of Science and Technology, Thuwal, Saudi Arabia f. University of Surrey, Guildford, UK g. Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE h. Eindhoven University of Technology, Eindhoven, NL

Contact: *t.huang2@exeter.ac.uk*

1 Motivations

Adapter-based fine-tuning has emerged as a popular strategy. However, many clinically important downstream tasks still operate in a pronounced low-data regime. This scenario naturally leads to the critical question:

Can standard Adapter perform effectively in medical imaging tasks under constrained data?

Uuderstanding Adapter in constrained-data settings

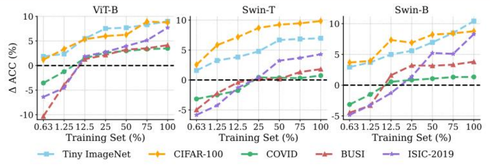

Figure 1: Performance of Adapter across Various Training Data Sizes.

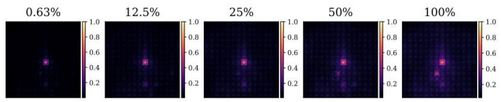

Figure 2: Effective Receptive Field of Standard Adapters Across Varying Training Set Ratios.

- ✓ **Degraded Adapter Performance with Less Training Data**: Adapter gains drop much more sharply on low-data medical datasets than on natural-vision datasets, suggesting limited robustness in out-of-domain settings.
- ✓ **Negative Effects of Adapters in Extremely Low Training Data in Medical Imaging**: At ≤1% data, standard Adapters can degrade medical-image performance, likely due to insufficient effective receptive field expansion.
- ✓ **Reduced Effective Receptive Field Under Constrained Training Data**: With less training data, Adapters yield smaller ERFs, limiting long-range spatial modeling and motivating ERF-aware Adapter design.

3 Results on Different Models

- ✓ **Medical-pretrained Models:** DKA consistently surpasses full fine-tuning, linear probing, and other PEFT baselines across ISIC-2019 and BUSI under all data scales on medical-pretrained backbones, including RadImageNet-pretrained ResNet-50 and MedSAM.

Methods	0.63%	1.25%	100%
Full Fine-tuning	52.70	55.17	76.63
Linear Probing	51.27	53.71	67.39
BitFit	48.84	51.62	70.38
Prompt	50.12	53.29	73.31
LoRA	50.03	52.51	72.92
Adapter	51.32	54.04	74.26
DKA	**53.69**	**56.58**	**78.56**

Methods	0.63%	1.25%	100%
Full Fine-tuning	36.21	45.04	70.62
Linear Probing	34.72	42.76	66.54
BitFit	27.85	36.92	64.43
Prompt	29.57	38.71	64.62
LoRA	32.39	40.35	67.04
Adapter	35.40	43.62	68.06
DKA	**37.13**	**46.27**	**72.53**

Table 1: Performance of DKA with medical-pretrained models.

- ✓ **Natural-pretrained Models:** DKA consistently outperforms all baselines across classification and segmentation tasks. These observations confirm the effectiveness of DKA when adapting natural-pretrained models to medical domains.

Methods	BRATS			BUSI			ISIC-2018		
	0.63%	1.25%	100%	0.63%	1.25%	100%	0.63%	1.25%	100%
Full Fine-tuning	9.25	22.39	73.08	26.67	32.31	57.41	62.27	73.63	77.58
Linear Probing	7.95	20.20	69.86	25.53	31.96	54.07	60.90	71.06	74.10
BitFit (Zaken et al., 2021)	1.20	14.33	63.52	7.13	17.08	52.56	53.21	65.84	73.10
Prompt (Jia et al., 2022)	1.22	15.21	64.53	9.19	18.77	53.57	56.56	67.40	73.61
LoRA (Hu et al., 2022)	3.84	16.19	68.48	14.10	22.39	53.96	58.70	69.03	73.84
Adapter (Houlsby et al., 2019)	6.16	18.95	72.02	18.18	25.90	55.01	59.78	72.80	76.71
Adapterformer (Chen et al., 2022)	5.99	18.77	72.54	17.41	25.68	55.14	59.67	72.85	76.58
Convpass (Jie et al., 2024)	7.13	19.64	73.32	19.84	28.52	56.09	60.19	73.56	77.54
CIAT (Zhu et al., 2021)	3.80	16.81	70.41	13.40	20.20	54.76	58.26	69.52	75.09
AIM (Yang et al., 2023)	4.58	17.61	71.98	15.40	23.54	54.78	59.24	72.05	76.65
DKA	**9.47**	**23.02**	**74.96**	**26.85**	**34.52**	**58.90**	**63.13**	**74.27**	**78.53**

Methods	COVID			BUSI			ISIC-2019		
	0.63%	1.25%	100%	0.63%	1.25%	100%	0.63%	1.25%	100%
Full Fine-tuning	87.43	88.00	98.43	71.17	76.73	94.62	60.04	61.21	82.05
Linear Probing	86.84	87.50	94.85	73.48	77.64	89.78	59.15	59.44	71.83
BitFit (Zaken et al., 2021)	73.91	79.65	96.95	57.19	60.20	88.27	50.80	53.22	79.84
Prompt (Jia et al., 2022)	77.91	83.75	98.45	61.34	64.30	93.07	52.55	53.59	81.02
LoRA (Hu et al., 2022)	80.43	85.91	98.73	63.64	67.41	94.75	51.08	53.96	81.75
Adapter (Houlsby et al., 2019)	83.29	86.26	98.33	63.18	73.68	93.33	52.77	54.88	79.54
Adapterformer (Chen et al., 2022)	82.46	84.32	98.18	63.42	72.75	92.65	51.62	52.71	78.19
Convpass (Jie et al., 2024)	84.72	86.94	98.45	64.83	74.63	93.97	54.72	56.25	80.45
CIAT (Zhu et al., 2021)	77.34	82.85	96.54	60.28	65.07	89.86	48.35	49.96	72.18
AIM (Yang et al., 2023)	80.92	83.55	97.23	62.72	70.34	90.12	50.12	52.72	77.39
DKA	**89.01**	**91.06**	**99.21**	**74.23**	**79.46**	**95.89**	**60.52**	**62.32**	**83.09**

Table 2: Performance of DKA with natural-pretrained models.

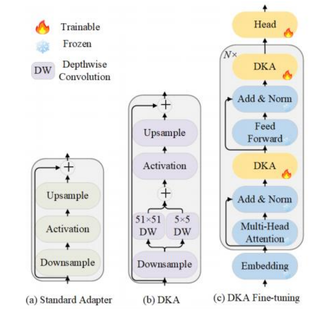

Figure 3: . Overview of the DKA Module. (a) Standard Adapter. (b) The proposed DKA module. (c) DKA Fine-tuning.

Dual-Kernel Adapter (DKA) 2

- ✓ DKA is inserted into a frozen pretrained backbone, while only the DKA parameters and the task-specific head are updated during fine-tuning, ensuring parameter efficiency.
- ✓ After the down-projection step, features are reshaped into image space and processed by two parallel depthwise convolution branches to capture complementary spatial information.
- ✓ A large 51×51 depthwise branch is used to enlarge the receptive field and capture broader contextual dependencies, while a small 5×5 branch preserves fine-grained local details.

Equation of DKA :

$$f_{\mathrm{DKA}}(\boldsymbol{x}) = \boldsymbol{x} + \mathrm{Up}(\sigma(\mathrm{DWConv}_{\mathrm{large}}(\mathrm{Down}(\boldsymbol{x})) + \mathrm{DWConv}_{\mathrm{small}}(\mathrm{Down}(\boldsymbol{x}))))$$

4 Large Kernel Matters

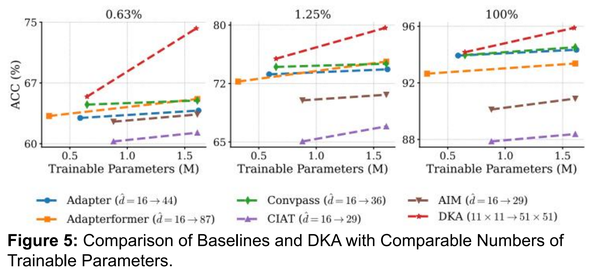

Figure 5: Comparison of Baselines and DKA with Comparable Numbers of Trainable Parameters.

ERF Visualiztion 5

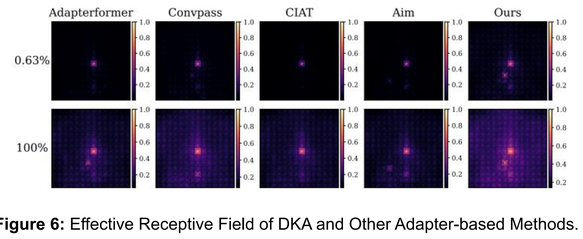

Figure 6: Effective Receptive Field of DKA and Other Adapter-based Methods.

6 Asynchronous Learning Rates Matter

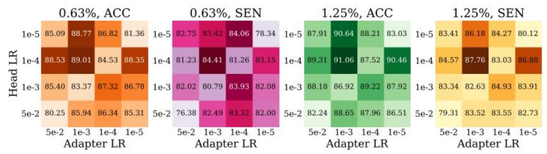

Figure 7: Performance Comparison Across Varying Learning Rates for DKA and Classification Head.

7 Kernel Size Selection

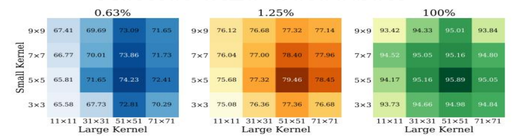

Figure 8: Performance of Different Kernel Size Combination.

Methods	0.63%	1.25%	100%
Learnable kernel sizes	88.98	91.05	99.21
$51\times51+5\times5$ (Ours)	89.01	91.06	99.21

Table 3: Comparison of Learnable Kernel Sizes and Our Design.

8 Middle Dimension

$\hat{d}$	0.63%	1.25%	100%
1	65.02	70.69	87.86
4	69.65	74.38	91.42
8	72.18	76.68	93.61
16	**74.23**	**79.64**	**95.89**
32	73.75	79.23	95.29

$\hat{d}$	0.63%	1.25%	100%
64	57.93	68.68	74.10
96	60.25	71.95	76.18
128	62.30	73.56	77.48
192	**63.13**	**74.27**	**78.06**
256	62.72	73.88	77.95

Table 3: Performance Comparison of Different Middle Dimensions.

Single vs. Dual Convolutions 9

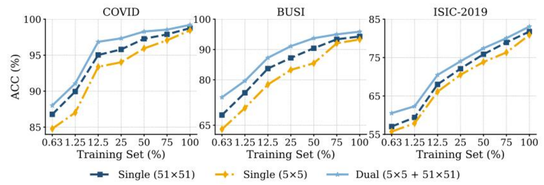

Figure 9: Single vs. Dual Convolutions.