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

Ziquan Zhu[1*], **Hanruo Zhu**[1*], **Si-Yuan Lu**[2*], **Xiang Li**[3], **Yanda Meng**[4], **Gaojie Jin**[5]
Lu Yin[6], **Lijie Hu**[7], **Di Wang**[7], **Lu Liu**[5], **Tianjin Huang**[5,9†]

[1] School of Computing and Mathematical Sciences, University of Leicester, Leicester, UK
[2] School of Communications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China
[3] Department of Computer Science, University of Bristol, Bristol, UK
[4] Department of Bioengineering, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia
[5] Department of Computer Science, University of Exeter, Exeter, UK
[6] Computer Science Research Centre, University of Surrey, Guildford, UK
[7] Machine Learning Department, Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE
[8] Division of Computer, Electrical and Mathematical Sciences and Engineering, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia
[9] Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, NL

Adapters have become a widely adopted strategy for efficient fine-tuning of large pretrained models, particularly in resource-constrained settings. However, their performance under extreme data scarcity—common in medical imaging due to high annotation costs, privacy regulations, and fragmented datasets—remains underexplored. In this work, we present the first comprehensive study of adapter-based fine-tuning for large pretrained models in low-data medical imaging scenarios. We find that, contrary to their promise, conventional Adapters can degrade performance under severe data constraints, performing even worse than simple linear probing when trained on less than 1% of the corresponding training data. Through systematic analysis, we identify a sharp reduction in Effective Receptive Field (ERF) as a key factor behind this degradation. Motivated by these findings, we propose the Dual-Kernel Adapter (`DKA`), a lightweight module that expands spatial context via large-kernel convolutions while preserving local detail with small-kernel counterparts. Extensive experiments across diverse classification and segmentation benchmarks show that `DKA` significantly outperforms existing Adapter methods, establishing new leading results in both data-constrained and data-rich regimes. Code is available at `https://github.com/misswayguy/DKA`.

1. Introduction

The rapid proliferation of large pretrained models has significantly advanced various fields such as natural language processing [1] and computer vision [2], yet it has also amplified challenges related to computational overhead [3], memory consumption [4], and the complexity of downstream adaptation [5], especially when deploying these models in specialized domains like medical imaging [6–8]. To address these issues, adapter-based fine-tuning [9–13] has emerged as a popular strategy, enabling efficient adaptation by adjusting only a small subset of model parameters rather than performing full fine-tuning.

In medical imaging, assembling large, well-annotasted datasets is notoriously costly: expert radiologists must painstakingly delineate structures in high-resolution 2-D and 3-D scans, and inter-

*Equal contribution.
†Corresponding author: T.Huang2@exeter.ac.uk.

Third Conference on Parsimony and Learning (CPAL 2026).

observer variability further inflates the annotation burden [14, 15]. Strict privacy regulations such as HIPAA [16] and the GDPR [17], coupled with heterogeneous institutional policies, severely limit data sharing, fragmenting what little data exist [18, 19]. As a result, 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?

In this paper, we first provide a comprehensive evaluation and analysis of applying conventional Adapter [20] under constrained-data scenarios across various datasets and backbone architectures. Our study reveals several key insights:

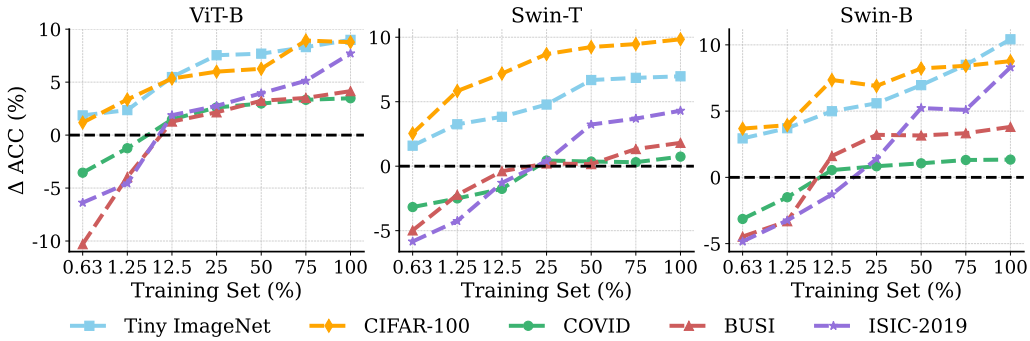

Figure 1: **Performance of Adapter across Various Training Data Sizes.** ΔACC = $\text{ACC}_{LinearProbing+Adapter} - \text{ACC}_{LinearProbing}$. Experiments are conducted on the pretrained ViT-B, Swin-T, and Swin-B backbones for Tiny ImageNet and CIFAR-100 (In-Domain), COVID, BUSI, and ISIC-2019 (Out-of-Domain).

- *Data Size Significantly Impacts Adapter Performance.* We observe that the benefits of using Adapters diminish substantially as the size of the training dataset decreases. This effect is particularly pronounced for medical imaging (see Figure 1).
- *Adapters Can Harm Performance Under Severe Low Data Settings.* When the training data size is reduced to 1% or less, specifically in the context of adapting large pretrained models to medical imaging, Adapters perform worse than linear probing. This indicates that, under extreme data constraints, Adapters may negatively affect model adaptation (see Figure 1).
- *Effective Receptive Field (ERF) Decreases with Reduced Training Data.* Visualization of the ERF demonstrates that smaller training datasets result in reduced ERF, offering a plausible explanation for the observed performance degradation (see Figure 2).

Inspired by these insights, we propose a Dual-Kernel Adapter (`DKA`) that explicitly enlarges the ERF of standard Adapter modules. Each `DKA` module introduces a dual-branch convolution design: one branch leverages a depthwise large-kernel convolution to broaden the ERF, while the other employs a depthwise small-kernel convolution to preserve fine-grained local details. We evaluate `DKA` on a range of medical imaging tasks, including both classification and segmentation, across diverse datasets and large pretrained models. Experimental results demonstrate that `DKA` consistently outperforms existing methods in both data-constrained and data-rich settings.

Summary of Contributions

⋆ We present the first systematic study of adapter-based fine-tuning for large pretrained models in low-data medical-imaging scenarios, showing that the conventional Adapter can actually degrade performance under severe data scarcity.

⋆ We introduce the `Dual-Kernel Adapter` (`DKA`), a lightweight module that pairs large- and small-receptive-field convolutions in parallel, simultaneously broadening spatial context and preserving fine-grained detail.

⋆ Extensive experiments on multiple segmentation and classification benchmarks demonstrate that DKA sets new state of the art, and ablation studies reveal that using asynchronous learning rates between adapters and linear head is critical to its gains.

2. Understanding Standard Adapter in Constrained-Data Settings

Adapters have gained popularity in medical image analysis due to their parameter efficiency and adaptability for fine-tuning large pretrained models. However, their performance under constrained data conditions remains underexplored. In this section, we investigate this critical aspect using a diverse set of datasets, including Tiny ImageNet [21], CIFAR-100 [22], COVID [23], BUSI [24], and ISIC-2019 [25]. Our experiments utilize three backbones—ViT-B [26], Swin-T [27], and Swin-B [27]—all pretrained on the ImageNet [28], to evaluate the impact of Adapters across varying data sizes, ranging from 0.63% to 100% of the training data. We report the difference in accuracy between the application of Adapters and the non-application of Adapters, which is denoted by $\Delta\text{ACC} = \text{ACC}_{LinearProbing+Adapter} - \text{ACC}_{LinearProbing}$. Our key observations are summarized as follows.

① **Degraded Adapter Performance with Less Training Data.** In Figure 1, we consistently observe that the performance gains provided by Adapters diminish across multiple tasks and pretrained models. Notably, this decline is significantly more pronounced in medical datasets (COVID, BUSI, ISIC-2019) compared to natural-vision datasets (TinyImageNet, CIFAR-100). A plausible explanation is that medical tasks represent out-of-domain scenarios, requiring feature representations that diverge considerably from those learned by the original pretrained models. This challenge is further exacerbated in low-data settings, where learning domain-specific features becomes more difficult. In contrast, natural-vision tasks remain largely in-domain, aligning more closely with the pretraining distribution, and thus benefiting more from adapter-based fine-tuning.

② **Negative Effects of Adapters in Extremely Low Training Data in Medical Imaging.** We further observe that in medical imaging tasks, when training data is limited to 1% or less, the performance gain from applying Adapters becomes negative, indicating a detrimental impact on model performance. Unlike natural images, medical images typically exhibit low contrast, ambiguous boundaries, and small or irregular pathological structures [29], which usually demand a large effective receptive field (ERF) to capture long-range contextual dependencies. However, standard Adapter do not possess a strong inductive bias toward expanding the ERF. **We hypothesize** that *under limited supervision, this limitation restricts the Adapter's capacity to learn spatially dispersed features and long-range contextual dependence, thereby contributing to the observed degradation in performance.*

③ **Reduced Effective Receptive Field Under Constrained Training Data.** To validate whether reduced supervision limits the Adapter's ability to capture spatially dispersed features and long-range contextual dependencies, we visualize the ERF of Adapters trained on varying proportions of the training set, ranging from 0.63% to 100%, using the COVID dataset and the pretrained ViT-B model. Following the definition in [30], the ERF of a neural network layer refers to the region encompassing all input pixels that exert a non-negligible influence on a given output unit. As shown in Figure 2, the ERF becomes progressively smaller as the amount of training data decreases. This observation supports our hypothesis that limited supervision restricts the Adapter's ability to learn spatially diverse patterns and long-range contextual relationships, which are particularly crucial in medical imaging tasks. We provide complementary ERF analysis on an alternative backbone (Swin-T) in Appendix F.4, which leads to consistent conclusions, reinforcing the generality of our observations.

These findings indicate that standard Adapter fail to enlarge the ERF in low-data settings, which in turn degrades model performance. Consequently, it is crucial to develop new Adapter architectures with a built-in inductive bias toward expanding the ERF.

3. Dual-Kernel Adapter

To mitigate the adverse effects of standard Adapter in low-data regimes, we propose the `Dual-Kernel Adapter` (`DKA`), which explicitly integrates a large-kernel convolution to expand the ERF and a small-

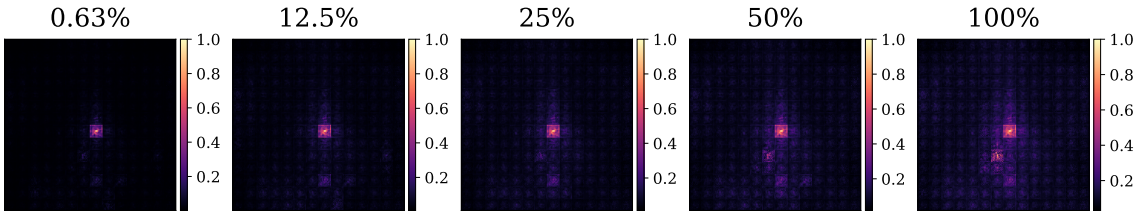

Figure 2: **Effective Receptive Field of Standard Adapters Across Varying Training Set Ratios.** Experiments are conducted on the COVID dataset using the pretrained ViT-B.

kernel convolution to preserve local spatial details. Prior studies have demonstrated that large kernels introduce a strong inductive bias toward capturing broader contextual information by expanding the ERF [31–33].

As illustrated in Figure 3, DKA first reduces the dimensionality of the input features through a linear down-projection. After this step, the patch tokens are reshaped back to their 2D spatial layout so that depthwise convolutions can be applied in the image domain. The reduced features are then processed in parallel by two depthwise convolution branches—a computationally efficient form of convolution that applies a distinct filter to each input channel [34]. One branch employs a large kernel to significantly enlarge the receptive field, facilitating the modeling of long-range dependencies. The other branch uses a smaller kernel to retain fine-grained spatial features essential for capturing localized structures. The outputs from the two branches are aggregated via element-wise summation, followed by a GELU activation, a linear up-projection, and a residual connection to the input.

Formally, the DKA operation can be expressed as:

$$f_{\texttt{DKA}}(\boldsymbol{x}) = \boldsymbol{x} + \mathrm{Up}\big(\sigma\big(\mathrm{DWConv}_{\mathrm{large}}(\mathrm{Down}(\boldsymbol{x})) + \mathrm{DWConv}_{\mathrm{small}}(\mathrm{Down}(\boldsymbol{x}))\big)\big) \tag{1}$$

where $\mathrm{Down}(\cdot)$ and $\mathrm{Up}(\cdot)$ denote linear projection layers, $\mathrm{DWConv}_{\mathrm{large}}$ and $\mathrm{DWConv}_{\mathrm{small}}$ are depthwise convolutions with kernel sizes 51 and 5 respectively, and σ is the GELU activation.

4. Experiments

To comprehensively evaluate the performance of DKA, we conduct extensive experiments on medical imaging tasks, spanning classification and segmentation benchmarks across diverse datasets, and additionally, models pretrained on both natural and medical domains.

Pretrained Models. We consider two representative categories of pretrained models. Natural-pretrained Models. For classification, we conduct experiments using ViT-B [26] and Swin-B [27], both pretrained on the ImageNet. For segmentation, we adopt Segmenter-B [35] in the OpenMMLab MMSegmentation [36]. Medical-pretrained Models. To assess domain adaptability, we further evaluate DKA on medical backbones, including RadImageNet-pretrained ResNet-50 [37] for classification and MedSAM [38] for segmentation.

Datasets and Metrics. We evaluate the proposed DKA across both medical classification and segmentation tasks to ensure broad applicability. Medical Image Classification. We utilize

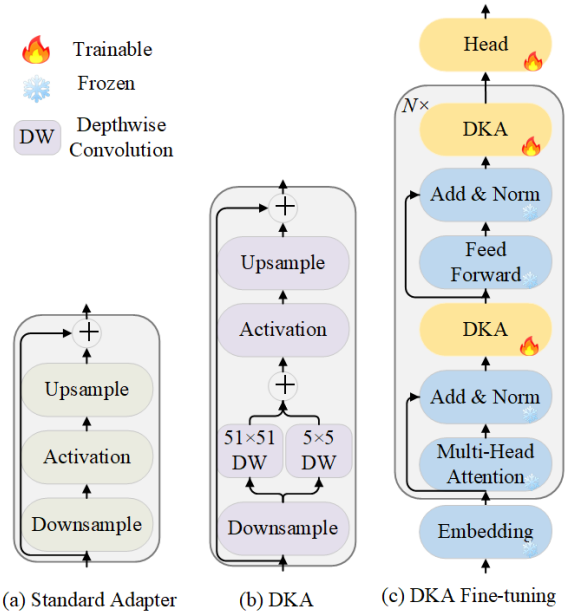

Figure 3: **Overview of the DKA Module.** (a) Standard Adapter. (b) The proposed DKA module. (c) DKA Fine-tuning.

three widely-adopted medical image classification datasets: COVID [23], BUSI [24], and ISIC-2019 [25]. The COVID dataset comprises chest X-ray images for COVID-19 diagnosis. The BUSI dataset includes breast ultrasound images categorized as benign, malignant, or normal. The ISIC-2019 dataset is a large-scale dermoscopic dataset designed for multi-class skin lesion classification. For each dataset, we report *Top-1 Accuracy* (*ACC*), *F1 Score* (*F1*); and *Sensitivity* (*SEN*) under varying proportions of training data. Medical Image Segmentation. We further assess DKA on three representative segmentation datasets: BRATS [39], BUSI [24], and ISIC-2018 [40]. BRATS focuses on brain tumor segmentation using multi-modal MRI scans. BUSI provides breast ultrasound images with annotated tumor regions. ISIC-2018 contains dermoscopic images with pixel-level annotations for skin lesion segmentation. We evaluate segmentation performance using standard metrics: *mean Intersection over Union* (*mIoU*) and *Dice coefficient* (*Dice*). More details are provided in Appendix B.1.

Baselines. We compare DKA against a broad range of baselines, categorized into three groups: Standard Fine-tuning Methods: (1) *Linear Probing*: freezing the backbone and tuning only the head; (2) *Full Fine-tuning*: updating all model parameters during downstream training. Adapter-tuning Methods: (1) *Adapter* [*20*]: adding adapters within each transformer block; (2) *AdapterFormer* [*41*]: adding parallel adapters with learnable scaling to each MLP layer; (3) *CIAT* [*42*]: adding adapters before transformer blocks and parallel adapters within blocks; (4) *Convpass* [*43*]: enhancing adapters with parallel 3×3 convolution branches; (5) *AIM* [*44*]: combining sequential and parallel adapters across spatial and temporal attention pathways. Other Parameter-efficient Fine-tuning (PEFT) Methods: (1) *Prompt Tuning* [*45*]: fine-tuning extra learnable tokens; (2) *LoRA* [*46*]: injecting low-rank trainable matrices into attention modules; (3) *BitFit* [*47*]:fine-tuning only the bias terms of pretrained models.

Implementation Details. Following the common training protocol, we freeze all pretrained model weights and update only the parameters of DKA and head during fine-tuning. Specifically, DKA modules are inserted within transformer blocks following the placement strategy described in [48]. For the DKA module, the middle dimension $\hat{d}$ is set to 16 for classification tasks and 192 for segmentation tasks, as discussed in Section 4.4. The learning rates are set to 1e-4 for the task head and 1e-3 for the DKA modules, as shown in Section 4.4. Classification models are trained for 100 epochs, while segmentation models are trained for 300 epochs, balancing the need for convergence across task complexities. For experiments under constrained supervision (i.e., training with less than 100% of the training set), we perform a 5-fold cross-validation on the training set while keeping the test set fixed. All reported results are averaged across the cross-validation folds. More details can be found in Appendix B.2.

4.1. Superior Performance on Constrained Data

We evaluate DKA under constrained-data settings (0.63% and 1.25%) and the full-data setting. Experiments are organized by the type of pretrained backbone: natural-image models (ViT-B, Segmenter-B) and medical-image models (RadImageNet-ResNet-50, MedSAM). This design enables a clear examination of DKA's generalization across different pretraining sources.

Natural-pretrained Models. We first evaluate DKA using natural-pretrained backbones, as summarized in Tables 1 and 2, DKA consistently outperforms all baselines across classification and segmentation tasks. Notably, under low-data regimes (0.63% and 1.25%), DKA even surpasses full fine-tuning and linear probing, while other PEFT methods exhibit clear performance degradation. These observations confirm the effectiveness of DKA when adapting natural-pretrained models to medical domains. Additional results on the natural-pretrained ViT-B and Segmenter-B are provided in Appendix D and C. Furthermore, similar performance improvements are observed when DKA is applied to other natural-pretrained models such as Swin-B (see Appendix F.1 for details).

Table 2: **Comparison of Baselines and DKA on Three Segmentation Datasets Under Varying Data Sizes.** Experiments are based on the pretrained Segmenter-B. mIoU is reported as percentages. The best results are highlighted in bold.

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 [47]	1.20	14.33	63.52	7.13	17.08	52.56	53.21	65.84	73.10
Prompt [45]	1.22	15.21	64.53	9.19	18.77	53.57	56.56	67.40	73.61
LoRA [46]	3.84	16.19	68.48	14.10	22.39	53.96	58.70	69.03	73.84
Adapter [20]	6.16	18.95	72.02	18.18	25.90	55.01	59.78	72.80	76.71
Adapterformer [41]	5.99	18.77	72.54	17.41	25.68	55.14	59.67	72.85	76.58
Convpass [43]	7.13	19.64	73.32	19.84	28.52	56.09	60.19	73.56	77.54
CIAT [42]	3.80	16.81	70.41	13.40	20.20	54.76	58.26	69.52	75.09
AIM [44]	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**

Table 1: **Comparison of Baselines and DKA on Three Classification Datasets Across Varying Data Sizes.** Results are reported in terms of ACC (%). Experiments are based on the pretrained ViT-B. The best results are highlighted in bold.

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 [47]	73.91	79.65	96.95	57.19	60.20	88.27	50.80	53.22	79.84
Prompt [45]	77.91	83.75	98.45	61.34	64.30	93.07	52.55	53.59	81.02
LoRA [46]	80.43	85.91	98.73	63.64	67.41	94.75	51.08	53.96	81.75
Adapter [20]	83.29	86.26	98.33	63.18	73.68	93.33	52.77	54.88	79.54
Adapterformer [41]	82.46	84.32	98.18	63.42	72.75	92.65	51.62	52.71	78.19
Convpass [43]	84.72	86.94	98.45	64.83	74.63	93.97	54.72	56.25	80.45
CIAT [42]	77.34	82.85	96.54	60.28	65.07	89.86	48.35	49.96	72.18
AIM [44]	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**

Medical-pretrained Models. We further evaluate DKA on medical-pretrained backbones, including RadImageNet-pretrained ResNet-50 [37] for classification and MedSAM [38] for segmentation. As reported in Table 3, DKA consistently surpasses full fine-tuning, linear probing, and other PEFT baselines across ISIC-2019 and BUSI under all data scales. These improvements demonstrate that the gains of DKA are not limited to natural image pretraining, but also generalize effectively to domain-specific medical large pretrained models. More results are reported in Appendix E.

Table 3: **Performance of DKA with medical-pretrained models.** (a) Classification results on the ISIC-2019 using RadImageNet-pretrained ResNet-50 by reporting ACC (%). (b) Segmentation results on the BUSI using MedSAM by reporting mIoU (%). The best results are highlighted in bold.

(a) Classification Results.

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**

(b) Segmentation Results.

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**

4.2. ERF Visualization

To assess whether DKA effectively expands the effective receptive field (ERF), we visualize the ERFs of DKA alongside those of other adapter-tuning baselines, including Adapter [20], AdapterFormer [41], Convpass [43], CIAT [42], and AIM [44], under both the constrained (0.63%) and the full (100%) training data settings. As shown in Figure 4, DKA consistently exhibits the broadest ERF across both settings, demonstrating its superior ability to capture extensive spatial context even under constrained-data conditions. In contrast, other baselines yield more localized ERFs, particularly in the constrained-data setting, which likely contributes to their comparatively weaker performance.

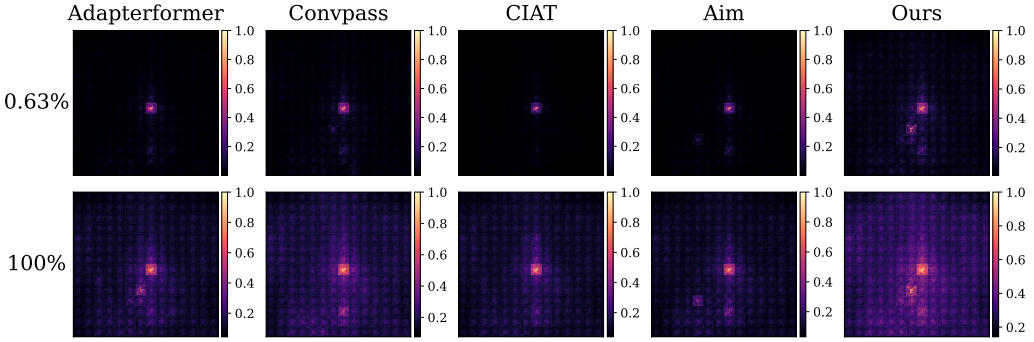

Figure 4: **Effective Receptive Field of DKA and Other Adapter-based Methods.** Experiments are conducted on the COVID dataset based on the pretrained ViT-B under both constrained (0.63%) and full (100%) training settings.

4.3. Large Kernel Matters Instead of Trainable Parameters

Given that DKA introduces a slightly higher number of trainable parameters, a natural question arises: *Are the observed performance gains primarily attributable to the increased parameter count or to the effect of the large kernel?* To investigate this, we perform a controlled comparison where the increase in trainable parameters arises either from enlarging the kernel size or from expanding the intermediate dimension $\hat{d}$. Specifically, we adjust the middle dimension $\hat{d}$ of other adapter-based baselines so that their total trainable parameter counts align with that of DKA with increasing kernel sizes. We fix the middle dimension $\hat{d} = 16$ in DKA throughout all classification experiments, ensuring that any parameter increase stems solely from the enlarged kernel. As shown in Figure 5, the results reveal that: ❶ DKA consistently outperforms all baselines across the 0.63%, 1.25%, and 100% training data settings under similar parameter constraints; ❷ the performance improvements resulting from increasing the kernel size (from 11 to 51) are significantly steeper than those obtained by merely enlarging the hidden dimension, highlighting the effectiveness of large kernels in enhancing DKA's representational capacity.

4.4. Ablation Study

Kernel Size Selection. We investigate the impact of different kernel size combinations in DKA's dual-branch convolution design. Specifically, we sweep over five candidate sizes (3×3, 5×5, 7×7, 9×9, 11×11, 31×31, 51×51, 71×71) for both small- and large-depthwise convolution branches. As shown in Figure 6, the combination of 5×5 (small) and 51×51 (large) consistently yields the best accuracy on both the data-constrained (0.63% and 1.25%) setting and the full data setting (100%). Notably, kernel sizes that are too small or too large result in performance degradation, particularly under low-data conditions. These results support our dual-branch design choice, balancing fine-grained detail extraction and large receptive field expanding. Extra experimental results are included in Appendix F.5.

Asynchronous Learning Rates Matter In standard adapter-based fine-tuning, it is common practice to use the same learning rate for both the adapter and the head. However, given their different roles and characteristics, this one-size-fits-all strategy may not be optimal. To investigate the effect

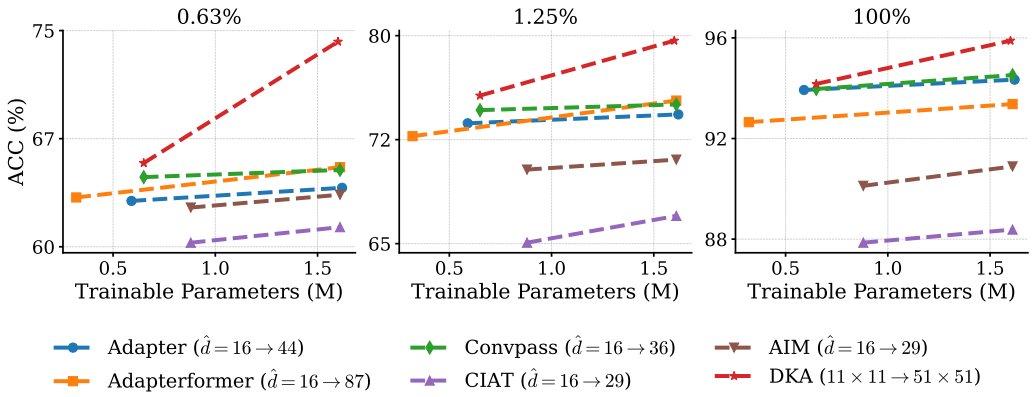

Figure 5: **Comparison of Baselines and DKA with Comparable Numbers of Trainable Parameters.** Experiments are conducted on 0.63%, 1.25%, and 100% subsets of the BUSI dataset using the pretrained ViT-B. The symbol $\hat{d}$ denotes the intermediate dimensionality in adapter-based methods.

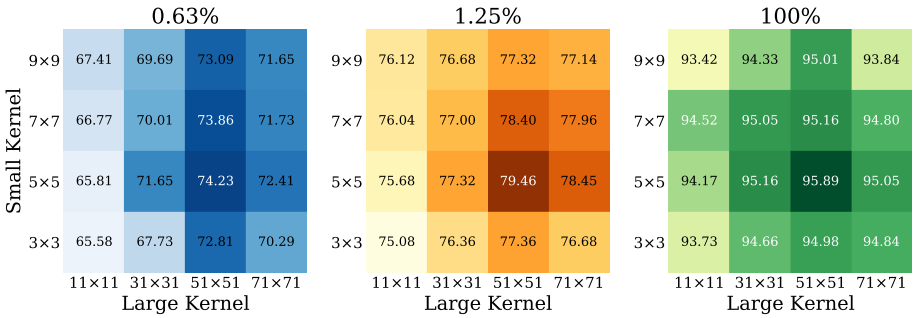

Figure 6: **Performance of Different Kernel Size Combination.** Experiments are conducted on the BUSI dataset using the pretrained ViT-B across three training setting (0.63%, 1.25%, and 100%). ACC (%) is reported.

of asynchronous learning rates for the adapter and head, we conduct experiments on the COVID dataset using a pretrained ViT-B. We systematically vary the learning rates assigned to the adapter and head, and assess the results under both the 0.63% and 1.25% training data. As shown in Figure 7, asynchronous learning rate schedules-where the adapter and head use different learning rates—often outperform symmetric configurations. We observe that the best results are not achieved when both components share the same learning rate, suggesting that the adapter and head benefit from distinct optimization dynamics. This trend holds across data scales, highlighting the importance of tuning these components independently. More results are available in Appendix F.2.

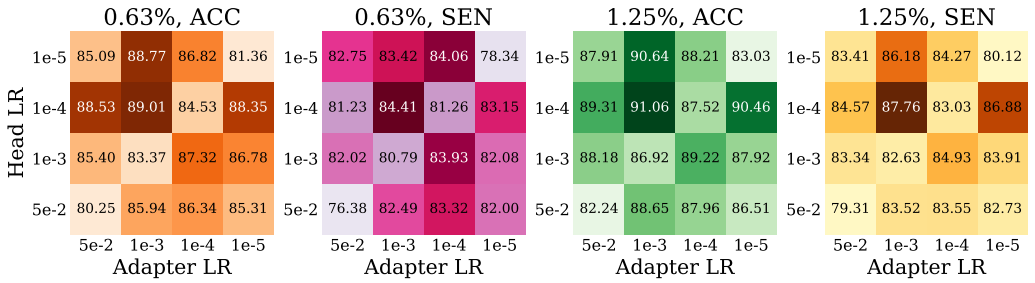

Figure 7: **Performance Comparison Across Varying Learning Rates for DKA and Classification Head.** Experiments are conducted on the COVID dataset using the pretrained ViT-B under 0.63% and 1.25% training data. Results are reported in terms of ACC (%) and SEN (%).

Single vs. Dual Convolutions. To evaluate the effectiveness of using both large- and small-depthwise convolutions in DKA, we compare the full dual-branch design ($5\times5 + 51\times51$) against single-branch variants that use only one of the two kernels. Experiments are conducted on the COVID, BUSI, and

Table 4: **Performance Comparison of Different Middle Dimensions $\hat{d}$ in DKA.** (a) Classification performance using the pretrained ViT-B on the BUSI dataset by reporting ACC (%). (b) Segmentation performance using the pretrained Segmenter-B on the ISIC-2018 dataset by reporting mIoU (%).

(a) Middle Dimension for Classification.

$\hat{d}$	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

(b) Middle Dimension for Segmentation.

$\hat{d}$	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

ISIC-2019 datasets across varying training set sizes. As shown in Figure 8, the dual-branch design consistently outperforms both single-branch variants, particularly in low-data regimes. While the 5×5 branch better captures localized detail and the 51×51 branch improves global coverage, neither alone matches the full-dual structure. See Appendix F.6 for additional results.

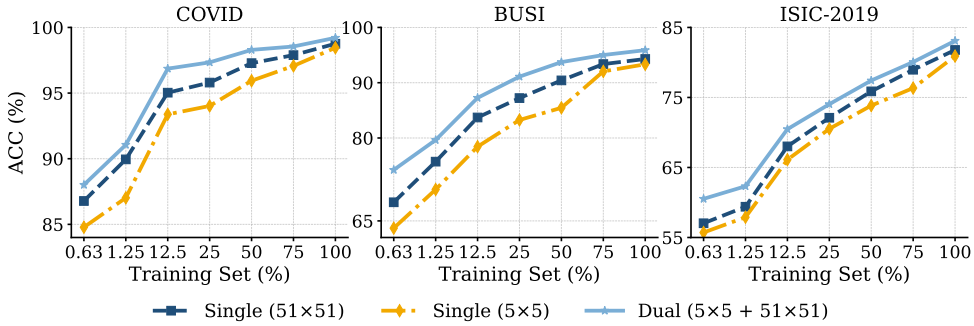

Figure 8: **Ablation for Dual-Convolution Design.** Experiments are based on the pretrained ViT-B across three classification datasets. ACC (%) is reported.

Middle Dimension. We investigate the impact of the middle dimension $\hat{d}$ in the DKA module for both classification and segmentation tasks. As shown in Table 4, increasing $\hat{d}$ generally leads to improved performance. For classification on the BUSI dataset, performance steadily improves from $\hat{d} = 1$ to $\hat{d} = 16$, reaching a peak at $\hat{d} = 16$, after which performance slightly declines or saturates, indicating potential overfitting or redundancy. For segmentation on the ISIC-2018 dataset, the optimal middle dimension is also moderate, with $\hat{d} = 192$ offering the best trade-off between parameter efficiency and accuracy. This observation aligns with the findings in [41], which also highlight the task-specific nature of optimal middle dimensions, emphasizing that different tasks require different parameter settings for optimal performance.

Learnable Kernel Sizes. To further investigate the learnable kernel sizes, following the design of Selective Kernel Networks [49], we implemented a learnable kernel-selection variant of our DKA. Specifically, we constructed two depthwise branches with different kernel sizes and fused them using a lightweight attention gate (global average pooling → two-layer MLP → softmax weights) that dynamically selects between the local and global branches for each input. This enables input-adaptive kernel aggregation. The results are reported in Table 5. While the learnable-kernel variant achieves competitive performance, our fixed $51 \times 51 + 5 \times 5$ design remains slightly more robust in low-data regimes (0.63% and 1.25%), likely because the learned gating requires additional data to generalize reliably. Under full data (100%), both approaches perform similarly. Overall, adaptive kernels are feasible, but our chosen configuration offers greater robustness and reliability in the low-data medical scenarios targeted in this work.

Table 5: **Comparison of Learnable Kernel Sizes and Our dDesign.** Experiments are conducted on the COVID dataset using the pretrained ViT-B.

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

5. Related Work

Adapter-based Fine-tuning adapts large pretrained models to downstream tasks by inserting and training lightweight modules while keeping the original model parameters frozen, offering significant computational and memory advantages over full fine-tuning [50–52]. Early Adapter methods introduced task-specific bottleneck layers between transformer blocks [20], with subsequent innovations improving architectural designs [9, 53] and multi-modal applications [54–56]. Recent medical imaging adaptations demonstrate how Adapter modules can effectively transfer pretrained knowledge to diagnostic tasks while maintaining model integrity [57–59]. Recent large-kernel designs such as LKA [60] enhance global receptive fields within convolutional backbones However, adapter-based methods typically assume access to a moderate amount of labeled data [61, 62]. Their effectiveness in data-constrained settings remains underexplored, raising critical questions about their performance in such scenarios.

Limited Data in Medical Imaging remains a major challenge, as labeled samples are scarce due to privacy, high annotation cost, and limited experts [63, 64]. This constraint is especially critical in clinical practice, where collecting large and diverse datasets is difficult [65–67]. To address scarcity, strategies include data augmentation [68–71], transfer learning [72–74], semi-supervised learning [75–79], and self-supervised representation learning [80–83]. These approaches leverage unlabeled data or external sources to improve generalization under low-resource settings [84, 85], and recent work highlights large pretrained models to further reduce annotation needs [86–88]. Nevertheless, most rely on full fine-tuning or moderately sized datasets [89, 90], while their integration with adapter-based methods under severe scarcity remains underexplored but highly relevant in practice.

6. Conclusion

In this paper, we revisit adapter-based fine-tuning for medical image analysis and uncovered key limitations in low-data settings: existing Adapters often struggle to capture relevant features under data scarcity, partly due to their constrained receptive field. To address this, we introduced `DKA`, a dual-branch adapter module that integrates large- and small-kernel depthwise convolutions to enhance the receptive field while preserving local detail. Experiments on both medical image classification and segmentation tasks, evaluated across various datasets and backbones show that `DKA` consistently outperform both full fine-tuning and other PEFT baselines by a good margin, particularly in the constrained-data setting, without significantly increasing parameter count.

7. Acknowledgments

The authors acknowledge the use of resources provided by the UKRI SLAIDER project, the MRC SLAIDER-QA project, the Isambard-AI National AI Research Resource (AIRR), and the Dutch national e-infrastructure, supported by the SURF Cooperative (Project EINF-17091). Isambard-AI is operated by the University of Bristol and funded by the UK Government's Department for Science, Innovation and Technology (DSIT) via UK Research and Innovation and the Science and Technology Facilities Council [ST/AIRR/I-A-I/1023]. Finally, we thank the anonymous reviewers for their insightful comments, which significantly improved the quality of this paper.

References

[1] KR1442 Chowdhary and KR Chowdhary. Natural language processing. *Fundamentals of artificial intelligence*, pages 603–649, 2020.

[2] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022.

[3] Neil C Thompson, Kristjan Greenewald, Keeheon Lee, Gabriel F Manso, et al. The computational limits of deep learning. *arXiv preprint arXiv:2007.05558*, 10, 2020.

[4] N Mahendran. Analysis of memory consumption by neural networks based on hyperparameters. *arXiv*, 2021.

[5] Wenyu Jiang, Zhenlong Liu, Zejian Xie, Songxin Zhang, Bingyi Jing, and Hongxin Wei. Exploring learning complexity for efficient downstream dataset pruning. *arXiv preprint arXiv:2402.05356*, 2024.

[6] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for modern deep learning research. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13693–13696, 2020.

[7] Cheng Ji, Fan Wu, Zongwei Zhu, Li-Pin Chang, Huanghe Liu, and Wenjie Zhai. Memory-efficient deep learning inference with incremental weight loading and data layout reorganization on edge systems. *Journal of Systems Architecture*, 118:102183, 2021.

[8] Jiaji Wang, Shuihua Wang, and Yudong Zhang. Deep learning on medical image analysis. *CAAI Transactions on Intelligence Technology*, 10(1):1–35, 2025.

[9] Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Guihong Cao, Daxin Jiang, Ming Zhou, et al. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*, 2020.

[10] Junde Wu, Ziyue Wang, Mingxuan Hong, Wei Ji, Huazhu Fu, Yanwu Xu, Min Xu, and Yueming Jin. Medical sam adapter: Adapting segment anything model for medical image segmentation. *Medical Image Analysis*, page 103547, 2025.

[11] Shizhan Gong, Yuan Zhong, Wenao Ma, Jinpeng Li, Zhao Wang, Jingyang Zhang, Pheng-Ann Heng, and Qi Dou. 3dsam-adapter: Holistic adaptation of sam from 2d to 3d for promptable medical image segmentation. *arXiv e-prints*, pages arXiv–2306, 2023.

[12] Jihong Hu, Yinhao Li, Hao Sun, Yu Song, Chujie Zhang, Lanfen Lin, and Yen-Wei Chen. Lga: A language guide adapter for advancing the sam model's capabilities in medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 610–620. Springer, 2024.

[13] Tianrun Chen, Ankang Lu, Lanyun Zhu, Chaotao Ding, Chunan Yu, Deyi Ji, Zejian Li, Lingyun Sun, Papa Mao, and Ying Zang. Sam2-adapter: Evaluating & adapting segment anything 2 in downstream tasks: Camouflage, shadow, medical image segmentation, and more. *arXiv preprint arXiv:2408.04579*, 2024.

[14] Kenji Suzuki. Overview of deep learning in medical imaging. *Radiological physics and technology*, 10(3):257–273, 2017.

[15] Felix Ritter, Tobias Boskamp, André Homeyer, Hendrik Laue, Michael Schwier, Florian Link, and H-O Peitgen. Medical image analysis. *IEEE pulse*, 2(6):60–70, 2011.

[16] U.S. Department of Health & Human Services. Summary of the hipaa privacy rule, 2003. URL `https://www.hhs.gov/hipaa/for-professionals/privacy/laws-regulations/index.html`.

[17] European Union. General data protection regulation (gdpr) – regulation (eu) 2016/679, 2016. URL https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[18] Raphael Schäfer, Till Nicke, et al. Overcoming data scarcity in biomedical imaging with a foundational multi-task model. *Nature Machine Intelligence*, 5:1–10, 2023.

[19] Rongguang Wang, Pratik Chaudhari, and Christos Davatzikos. Embracing the disharmony in medical imaging: A simple and effective framework for domain adaptation. *Medical Image Analysis*, 72:102–110, 2021.

[20] 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.

[21] Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge. *CS231n*, 2015.

[22] Neha Sharma, Vibhor Jain, and Anju Mishra. An analysis of convolutional neural networks for image classification. *Procedia computer science*, 132:377–384, 2018.

[23] Muhammad EH Chowdhury, Tawsifur Rahman, Amith Khandakar, Rashid Mazhar, Muhammad Abdul Kadir, Zaid Bin Mahbub, Khandakar Reajul Islam, Muhammad Salman Khan, Atif Iqbal, Nasser Al Emadi, et al. Can ai help in screening viral and covid-19 pneumonia? *Ieee Access*, 8:132665–132676, 2020.

[24] Yingtao Zhang, Min Xian, Heng-Da Cheng, Bryar Shareef, Jianrui Ding, Fei Xu, Kuan Huang, Boyu Zhang, Chunping Ning, and Ying Wang. Busis: a benchmark for breast ultrasound image segmentation. In *Healthcare*, volume 10, page 729. MDPI, 2022.

[25] Nils Gessert, Maximilian Nielsen, Mohsin Shaikh, René Werner, and Alexander Schlaefer. Skin lesion classification using ensembles of multi-resolution efficientnets with meta data. *MethodsX*, 7:100864, 2020.

[26] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2021.

[27] 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. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[28] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[29] Yichi Zhang, Zhenrong Shen, and Rushi Jiao. Segment anything model for medical image segmentation: Current applications and future directions. *Computers in Biology and Medicine*, page 108238, 2024.

[30] André Araujo, Wade Norris, and Jack Sim. Computing receptive fields of convolutional neural networks. *Distill*, 4(11):e21, 2019.

[31] Tianjin Huang, Lu Yin, Zhenyu Zhang, Li Shen, Meng Fang, Mykola Pechenizkiy, Zhangyang Wang, and Shiwei Liu. Are large kernels better teachers than transformers for convnets? In *International Conference on Machine Learning*, pages 14023–14038. PMLR, 2023.

[32] Shiwei Liu, Tianlong Chen, Xiaohan Chen, Xuxi Chen, Qiao Xiao, Boqian Wu, Tommi Kärkkäinen, Mykola Pechenizkiy, Decebal Mocanu, and Zhangyang Wang. More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. *arXiv preprint arXiv:2207.03620*, 2022.

[33] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11963–11975, 2022.

[34] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[35] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7262–7272, 2021.

[36] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation`, 2020.

[37] Xueyan Mei, Zelong Liu, Philip M Robson, Brett Marinelli, Mingqian Huang, Amish Doshi, Adam Jacobi, Chendi Cao, Katherine E Link, Thomas Yang, et al. Radimagenet: an open radiologic deep learning research dataset for effective transfer learning. *Radiology: Artificial Intelligence*, 4(5):e210315, 2022.

[38] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. *Nature Communications*, 15(1):654, 2024.

[39] Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34 (10):1993–2024, 2014.

[40] Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2019.

[41] 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.

[42] Yaoming Zhu, Jiangtao Feng, Chengqi Zhao, Mingxuan Wang, and Lei Li. Counter-interference adapter for multilingual machine translation. *arXiv preprint arXiv:2104.08154*, 2021.

[43] Shibo Jie, Zhi-Hong Deng, Shixuan Chen, and Zhijuan Jin. Convolutional bypasses are better vision transformer adapters. In *ECAI 2024*, pages 202–209. IOS Press, 2024.

[44] 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.

[45] 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.

[46] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[47] 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.

[48] Dongshuo Yin, Leiyi Hu, Bin Li, Youqun Zhang, and Xue Yang. 5%> 100%: Breaking performance shackles of full fine-tuning on visual recognition tasks. *arXiv preprint arXiv:2408.08345*, 2024.

[49] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 510–519, 2019.

[50] Lingling Xu et al. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *arXiv preprint arXiv:2312.12148*, 2023.

[51] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023.

[52] Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

[53] Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021.

[54] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5227–5237, 2022.

[55] 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.

[56] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, 132(2):581–595, 2024.

[57] Tianrun Chen, Lanyun Zhu, Chaotao Deng, Runlong Cao, Yan Wang, Shangzhan Zhang, Zejian Li, Lingyun Sun, Ying Zang, and Papa Mao. Sam-adapter: Adapting segment anything in underperformed scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3367–3375, 2023.

[58] Raman Dutt, Linus Ericsson, Pedro Sanchez, Sotirios A Tsaftaris, and Timothy Hospedales. Parameter-efficient fine-tuning for medical image analysis: The missed opportunity. *arXiv preprint arXiv:2305.08252*, 2023.

[59] Chenyu Lian, Hong-Yu Zhou, Yizhou Yu, and Liansheng Wang. Less could be better: Parameter-efficient fine-tuning advances medical vision foundation models. *arXiv preprint arXiv:2401.12215*, 2024.

[60] Ziquan Zhu, Si-Yuan Lu, Tianjin Huang, Lu Liu, and Zhe Liu. Lka: L arge k ernel a dapter for enhanced medical image classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 394–404. Springer, 2025.

[61] Raman Dutt, Ondrej Bohdal, Sotirios A Tsaftaris, and Timothy Hospedales. Fairtune: Optimizing parameter efficient fine tuning for fairness in medical image analysis. *arXiv preprint arXiv:2310.05055*, 2023.

[62] Gang Liu, Jinlong He, Pengfei Li, Genrong He, Zhaolin Chen, and Shenjun Zhong. Pefomed: Parameter efficient fine-tuning of multimodal large language models for medical imaging. *arXiv preprint arXiv:2401.02797*, 2024.

[63] Torgyn Shaikhina and Natalia A Khovanova. Handling limited datasets with neural networks in medical applications: A small-data approach. *Artificial intelligence in medicine*, 75:51–63, 2017.

[64] Phillip Chlap, Hang Min, Nym Vandenberg, Jason Dowling, Lois Holloway, and Annette Haworth. A review of medical image data augmentation techniques for deep learning applications. *Journal of medical imaging and radiation oncology*, 65(5):545–563, 2021.

[65] GW Ewing. The limitations of big data in healthcare. *MOJ Proteomics Bioinform*, 5(2):00152, 2017.

[66] Choong Ho Lee and Hyung-Jin Yoon. Medical big data: promise and challenges. *Kidney research and clinical practice*, 36(1):3, 2017.

[67] Der-Chiang Li, Chiao-Wen Liu, and Susan C Hu. A learning method for the class imbalance problem with medical data sets. *Computers in biology and medicine*, 40(5):509–518, 2010.

[68] Fabio Garcea, Alessio Serra, Fabrizio Lamberti, and Lia Morra. Data augmentation for medical imaging: A systematic literature review. *Computers in Biology and Medicine*, 152:106391, 2023.

[69] Evgin Goceri. Medical image data augmentation: techniques, comparisons and interpretations. *Artificial Intelligence Review*, 56(11):12561–12605, 2023.

[70] Amy Zhao, Guha Balakrishnan, Fredo Durand, John V Guttag, and Adrian V Dalca. Data augmentation using learned transformations for one-shot medical image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8543–8553, 2019.

[71] Tauhidul Islam, Md Sadman Hafiz, Jamin Rahman Jim, Md Mohsin Kabir, and MF Mridha. A systematic review of deep learning data augmentation in medical imaging: Recent advances and future research directions. *Healthcare Analytics*, page 100340, 2024.

[72] Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. *Advances in neural information processing systems*, 32, 2019.

[73] Hee E Kim, Alejandro Cosa-Linan, Nandhini Santhanam, Mahboubeh Jannesari, Mate E Maros, and Thomas Ganslandt. Transfer learning for medical image classification: a literature review. *BMC medical imaging*, 22(1):69, 2022.

[74] Padmavathi Kora, Chui Ping Ooi, Oliver Faust, U Raghavendra, Anjan Gudigar, Wai Yee Chan, K Meenakshi, K Swaraja, Pawel Plawiak, and U Rajendra Acharya. Transfer learning techniques for medical image analysis: A review. *Biocybernetics and biomedical engineering*, 42(1):79–107, 2022.

[75] Tri Huynh, Aiden Nibali, and Zhen He. Semi-supervised learning for medical image classification using imbalanced training data. *Computer methods and programs in biomedicine*, 216:106628, 2022.

[76] Asma Chebli, Akila Djebbar, and Hayet Farida Marouani. Semi-supervised learning for medical application: A survey. In *2018 international conference on applied smart systems (ICASS)*, pages 1–9. IEEE, 2018.

[77] Dong Wang, Yuan Zhang, Kexin Zhang, and Liwei Wang. Focalmix: Semi-supervised learning for 3d medical image detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3951–3960, 2020.

[78] Kai Han, Victor S Sheng, Yuqing Song, Yi Liu, Chengjian Qiu, Siqi Ma, and Zhe Liu. Deep semi-supervised learning for medical image segmentation: A review. *Expert Systems with Applications*, 245:123052, 2024.

[79] Yi Zhou, Xiaodong He, Lei Huang, Li Liu, Fan Zhu, Shanshan Cui, and Ling Shao. Collaborative learning of semi-supervised segmentation and classification for medical images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2079–2088, 2019.

[80] Linus Ericsson, Henry Gouk, Chen Change Loy, and Timothy M Hospedales. Self-supervised representation learning: Introduction, advances, and challenges. *IEEE Signal Processing Magazine*, 39(3):42–62, 2022.

[81] Jianbo Jiao, Richard Droste, Lior Drukker, Aris T Papageorghiou, and J Alison Noble. Self-supervised representation learning for ultrasound video. In *2020 IEEE 17th international symposium on biomedical imaging (ISBI)*, pages 1847–1850. IEEE, 2020.

[82] Yiwen Ye, Yutong Xie, Jianpeng Zhang, Ziyang Chen, Qi Wu, and Yong Xia. Continual self-supervised learning: Towards universal multi-modal medical data representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11114–11124, 2024.

[83] Rayan Krishnan, Pranav Rajpurkar, and Eric J Topol. Self-supervised learning in medicine and healthcare. *Nature Biomedical Engineering*, 6(12):1346–1352, 2022.

[84] Fudan Zheng, Jindong Cao, Weijiang Yu, Zhiguang Chen, Nong Xiao, and Yutong Lu. Exploring low-resource medical image classification with weakly supervised prompt learning. *Pattern Recognition*, 149:110250, 2024.

[85] Shuai Lin, Pan Zhou, Xiaodan Liang, Jianheng Tang, Ruihui Zhao, Ziliang Chen, and Liang Lin. Graph-evolving meta-learning for low-resource medical dialogue generation. In *proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 13362–13370, 2021.

[86] Michael Moor, Oishi Banerjee, Zahra Shakeri Hossein Abad, Harlan M Krumholz, Jure Leskovec, Eric J Topol, and Pranav Rajpurkar. Foundation models for generalist medical artificial intelligence. *Nature*, 616(7956):259–265, 2023.

[87] Shaoting Zhang and Dimitris Metaxas. On the challenges and perspectives of foundation models for medical image analysis. *Medical image analysis*, 91:102996, 2024.

[88] Wasif Khan, Seowung Leem, Kyle B See, Joshua K Wong, Shaoting Zhang, and Ruogu Fang. A comprehensive survey of foundation models in medicine. *IEEE Reviews in Biomedical Engineering*, 2025.

[89] Ana Davila, Jacinto Colan, and Yasuhisa Hasegawa. Comparison of fine-tuning strategies for transfer learning in medical image classification. *Image and Vision Computing*, 146:105012, 2024.

[90] Muhammad Osama Khan and Yi Fang. Revisiting fine-tuning strategies for self-supervised medical imaging analysis. *arXiv preprint arXiv:2307.10915*, 2023.

[91] A Paszke. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[92] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[93] Zifeng Wang, Zhenbang Wu, Dinesh Agarwal, and Jimeng Sun. Medclip: Contrastive learning from unpaired medical images and text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2022, page 3876, 2022.

[94] Fereshteh Shakeri, Yunshi Huang, Julio Silva-Rodríguez, Houda Bahig, An Tang, Jose Dolz, and Ismail Ben Ayed. Few-shot adaptation of medical vision-language models. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 553–563. Springer, 2024.

[95] Julio Silva-Rodriguez, Sina Hajimiri, Ismail Ben Ayed, and Jose Dolz. A closer look at the few-shot adaptation of large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23681–23690, 2024.

[96] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

Appendix

A. The Use of Large Language Models (LLMs)

Large language models (LLMs) were not used as part of the core methodology, experiments, or original research contributions. They were only used as an assistive tool for language editing and formatting.

B. Experiment Settings

B.1. Datasets

B.1.1. Classification Datasets

COVID [23]: The COVID-19 Radiography Database, developed through a collaborative effort involving researchers from Qatar University, the University of Dhaka, and medical experts from Pakistan and Malaysia, includes approximately 3616 COVID-19 cases, 10,192 normal cases, 6012 lung opacity cases (representing non-COVID lung infections), and 1345 viral pneumonia cases. This dataset provides a comprehensive collection of chest X-ray (CXR) images for the diagnosis of COVID-19 and other lung conditions.

BUSI [24]: The BUSI (Breast Ultrasound Images) dataset, collected in 2018, comprises approximately 780 ultrasound images from 600 female patients aged 25 to 75. It includes around 437 normal, 210 benign, and 133 malignant breast lesion images, each with an average resolution of approximately 500×500 pixels. The dataset is organized into three primary categories based on the clinical classification of breast lesions: normal, benign, and malignant.

ISIC-2019 [25]: The ISIC-2019 dataset, part of the annual ISIC (International Skin Imaging Collaboration) challenges, includes 25,331 dermoscopic images compiled from previous ISIC challenges (2017 and 2018). It spans nine diagnostic categories: melanoma, melanocytic nevus, basal cell carcinoma, actinic keratosis, benign keratosis (including solar lentigo, seborrheic keratosis, and lichen planus-like keratosis), dermatofibroma, vascular lesion, squamous cell carcinoma, and a category for images that do not fit into any of the other classes, providing a comprehensive benchmark for multi-class skin lesion classification.

B.1.2. Segmentation Datasets

BRATS [39]: The BRATS (Brain Tumor Segmentation) dataset provides multi-institutional, clinically acquired multi-parametric MRI (mpMRI) scans of gliomas, including T1, post-contrast T1-weighted (T1Gd), T2, and T2-FLAIR volumes. It includes pathologically confirmed cases with expert-annotated tumor sub-regions, including the enhancing tumor (ET), tumor core (TC), and whole tumor (WT), providing a comprehensive benchmark for brain tumor segmentation.

BUSI [24]: The BUSI (Breast Ultrasound Images) dataset includes ground truth masks for precise lesion segmentation, facilitating the evaluation of automated lesion detection and boundary delineation methods. These masks correspond to the original ultrasound images, capturing the exact regions of interest for each lesion type. The dataset primarily supports the segmentation of benign and malignant breast lesions, providing a detailed representation of lesion morphology.

ISIC-2018 [40]: The ISIC-2018 dataset, released as part of the ISIC 2018 Task 1 challenge, contains a total of 3,694 dermoscopic images, each paired with a binary segmentation mask outlining the precise lesion boundaries. This dataset serves as a critical benchmark for evaluating automated skin lesion segmentation methods.

B.2. Implementation Details

Our experiments are conducted on NVIDIA RTX 3090 GPUs. The code is based on PyTorch [91]. Due to hardware memory constraints, we use different batch sizes for classification and segmentation tasks. Specifically, classification tasks use a batch size of 8, while segmentation tasks are restricted

to a batch size of 1, as larger batch sizes lead to out-of-memory errors. The test set is fixed at 20% of the total dataset, ensuring consistent evaluation across all experimental settings. We employ Adam optimizer [92] for both the head and DKA modules, albeit with different learning rates for each. When reducing the training data sizes, we proportionally decrease samples per class to maintain balance. Even in extreme low-data scenarios, we ensure that each class retains some images to prevent complete absence of any category. The weights of the down-projection layers and the biases and weights of the up-projection layers are initialized to zero, providing a stable starting point for training.

C. Additional Segmentation Results on Natural-pretrained Models

C.1. Segmentation Results with Additional Metrics

To complement the segmentation results in the main text, we provide additional evaluations in Table 6, where performance is reported in terms of Dice scores across three datasets (BRATS, BUSI, ISIC-2018) under different training ratios (0.63%, 1.25%, 100%). The results confirm that DKA consistently achieves higher Dice scores than all baseline methods across datasets and data scales, further demonstrating its robustness in capturing both local details and global context for medical image segmentation.

Table 6: **Comparison of Baselines and DKA on Three Segmentation Datasets Under Varying Data Sizes.** Experiments are based on the pretrained Segmenter-B. Results are reported in terms of Dice (%). The best results are highlighted in bold.

Methods	BRATS			BUSI			ISIC-2018		
	0.63%	1.25%	100%	0.63%	1.25%	100%	0.63%	1.25%	100%
Full Fine-tuning	16.94	36.59	84.45	42.11	48.83	72.94	76.75	84.81	87.37
Linear Probing	14.72	33.62	82.26	40.67	48.44	70.19	75.70	83.08	85.12
BitFit [47]	2.40	25.06	77.69	13.31	29.18	68.90	69.46	79.41	84.46
Prompt [45]	2.42	26.41	78.44	16.84	31.61	69.76	72.25	80.52	84.80
LoRA [46]	7.40	28.86	81.29	24.72	36.59	70.09	73.97	81.68	84.95
Adapter [20]	11.60	31.86	83.73	30.77	41.14	70.98	74.83	84.26	86.82
Adapterformer [41]	11.30	31.61	84.08	29.66	40.86	71.08	74.74	84.29	86.73
Convpass [43]	13.31	32.83	84.60	33.11	44.39	71.87	75.14	84.77	87.35
CIAT [42]	7.33	28.72	82.64	23.63	33.62	70.77	73.63	82.02	85.77
AIM [44]	8.75	30.49	83.71	26.69	38.11	70.79	74.41	83.75	86.78
DKA	**17.29**	**37.43**	**85.69**	**42.34**	**51.33**	**74.13**	**77.39**	**85.24**	**87.97**

C.2. Segmentation Results with More Data Scales

To further validate the performance of DKA in segmentation tasks, we present additional results across a wider range of data scales (12.5% to 75%) in Table 7. These results, covering three segmentation medical imaging datasets, provide a comprehensive assessment of our model's performance in mid-scale training regimes, complementing the extreme low-data (0.63% and 1.25%) and full-data (100%) findings discussed in Section 4.1). Consistent with our previous observations, DKA outperforms all baselines across all datasets and training set ratios, achieving the highest mIoU and Dice metrics. This demonstrates the effectiveness of DKA, highlighting its ability to effectively capture both local detail and global context across varying supervision levels, providing a reliable solution for medical image segmentation.

Table 7: **Additional Segmentation Results on Three Datasets with Varying Training Set Ratios.** Experiments are based on the pretrained Segmenter-B. Results are reported as percentages for mIoU and Dice. The best results are highlighted in bold.

Methods	BRATS				BUSI				ISIC-2018			
	12.5%	25%	50%	75%	12.5%	25%	50%	75%	12.5%	25%	50%	75%
mIoU (%)												
Full Fine-tuning	55.28	61.35	70.43	71.50	51.44	53.70	55.35	56.8	74.60	75.36	75.50	77.10
Linear Probing	53.22	59.43	67.75	68.11	48.93	50.76	51.97	53.41	71.65	71.99	72.18	73.63
BitFit [47]	47.21	53.46	60.30	61.59	36.30	41.07	42.45	48.09	65.27	66.36	68.00	71.55
Prompt [45]	48.38	54.08	62.73	63.01	38.49	43.38	46.91	50.30	68.54	69.16	69.63	72.31
LoRA [46]	50.77	57.19	64.80	67.75	41.22	47.72	50.42	52.97	69.64	70.83	71.61	72.95
Adapter [20]	52.52	59.90	69.22	70.65	45.25	49.66	52.11	54.19	73.54	73.87	74.48	76.29
Adapterformer [41]	52.12	59.36	69.11	70.37	45.22	49.25	52.40	54.44	73.45	73.79	74.15	75.66
Convpass [43]	54.08	61.08	69.67	71.26	49.30	51.77	53.61	55.27	74.31	74.98	75.13	76.94
CIAT [42]	51.70	58.23	67.59	68.45	40.74	47.72	51.71	53.85	70.83	71.42	72.34	74.57
AIM [44]	51.09	58.43	68.11	69.34	43.44	48.15	51.20	53.96	72.71	73.36	73.55	75.30
DKA	**56.54**	**62.92**	**71.95**	**72.92**	**52.30**	**54.76**	**56.43**	**58.08**	**75.29**	**75.83**	**76.02**	**78.06**
Dice (%)												
Full Fine-tuning	71.20	76.05	82.65	83.38	67.93	69.88	71.26	72.45	85.46	85.95	86.04	87.07
Linear Probing	69.47	73.33	80.77	81.03	65.71	67.34	68.39	69.63	83.48	83.71	83.85	84.81
BitFit [47]	64.86	69.43	76.86	76.14	53.57	58.29	61.45	66.42	78.20	79.21	80.13	81.78
Prompt [45]	65.21	70.20	77.10	77.31	55.59	60.51	63.86	67.07	81.33	81.77	82.10	83.93
LoRA [46]	67.35	72.77	78.64	80.77	58.37	64.61	67.04	69.26	82.10	82.92	83.45	84.36
Adapter [20]	68.87	74.92	81.81	82.80	62.30	66.36	68.52	70.29	84.75	84.97	85.37	86.55
Adapterformer [41]	68.53	74.49	81.73	82.61	62.28	66.00	68.76	70.50	84.69	84.92	85.16	86.15
Convpass [43]	70.20	75.84	82.13	83.22	66.04	68.22	69.80	71.19	85.26	85.70	85.80	86.97
CIAT [42]	68.17	73.60	80.66	81.27	57.89	64.61	68.17	70.01	82.92	83.33	83.95	85.43
AIM [44]	67.63	73.76	81.03	81.90	60.56	65.00	67.73	70.10	84.20	84.63	84.76	85.91
DKA	**72.24**	**77.24**	**83.69**	**84.34**	**68.68**	**70.77**	**72.15**	**73.48**	**85.91**	**86.25**	**86.38**	**87.68**

D. Additional Classification Results on Natural-pretrained Models

D.1. Classification Results with Additional Metrics

In addition to the accuracy results reported in the Section 4.1, we further provide a comprehensive evaluation of DKA and baselines using F1 and sensitivity (SEN) scores on three classification datasets: COVID, BUSI, and ISIC-2019. As summarized in Table 8, DKA consistently achieves the best or near-best performance across all data scales (0.63%, 1.25%, and 100%). In particular, under low-data regimes, DKA exhibits notable improvements over existing PEFT approaches such as BitFit, LoRA, and Adapter-based methods, highlighting its robustness in capturing discriminative features when supervision is scarce. These results further validate the effectiveness of DKA beyond standard accuracy and confirm its ability to improve both predictive balance (F1) and clinical reliability (SEN) in medical image classification.

Table 8: **Comparison of Baselines and DKA on Three Classification Datasets Across Varying Data Sizes.** Experiments are based on the pretrained ViT-B. SEN is reported as percentages, while F1 is presented as raw value. The best results are highlighted in bold.

Methods	COVID			BUSI			ISIC-2019		
	0.63%	1.25%	100%	0.63%	1.25%	100%	0.63%	1.25%	100%
F1									
Full Fine-tuning	0.845	0.855	0.970	0.697	0.752	0.939	0.293	0.317	0.539
Linear Probing	0.831	0.843	0.932	0.710	0.756	0.897	0.287	0.307	0.442
BitFit [47]	0.718	0.762	0.950	0.542	0.573	0.879	0.184	0.235	0.501
Prompt [45]	0.753	0.819	0.972	0.588	0.618	0.930	0.218	0.235	0.534
LoRA [46]	0.786	0.814	0.973	0.603	0.641	0.945	0.214	0.235	0.528
Adapter [20]	0.802	0.833	0.978	0.607	0.704	0.931	0.231	0.248	0.501
Adapterformer [41]	0.793	0.807	0.974	0.614	0.693	0.918	0.228	0.231	0.495
Convpass [43]	0.807	0.833	0.969	0.622	0.718	0.938	0.253	0.275	0.512
CIAT [42]	0.748	0.794	0.947	0.574	0.635	0.899	0.162	0.195	0.429
AIM [44]	0.776	0.804	0.953	0.602	0.682	0.907	0.182	0.227	0.433
DKA	**0.865**	**0.881**	**0.981**	**0.720**	**0.774**	**0.951**	**0.296**	**0.326**	**0.542**
SEN (%)									
Full Fine-tuning	83.57	84.15	97.24	69.27	74.26	93.46	27.76	30.64	52.35
Linear Probing	82.64	83.49	92.74	70.25	74.43	87.74	26.87	28.68	42.71
BitFit [47]	70.36	75.34	94.45	51.33	54.41	85.52	18.50	21.74	47.55
Prompt [45]	74.31	81.76	96.57	56.71	59.35	91.34	20.65	23.92	49.28
LoRA [46]	77.42	80.76	96.76	59.37	61.41	92.07	20.11	24.13	50.75
Adapter [20]	81.38	82.40	97.41	59.40	66.69	92.58	21.84	24.93	49.50
Adapterformer [41]	79.34	81.51	97.40	60.96	65.44	90.72	20.36	22.78	48.31
Convpass [43]	81.23	82.71	95.76	61.79	67.90	92.74	24.63	27.36	51.12
CIAT [42]	74.93	78.34	94.35	54.52	61.57	87.85	16.24	20.17	42.18
AIM [44]	76.04	80.76	94.50	58.75	65.46	89.45	18.06	21.07	47.16
DKA	**84.41**	**87.76**	**98.12**	**71.82**	**76.78**	**95.46**	**28.20**	**31.50**	**53.69**

D.2. Classification Results with More Data Scales

To further substantiate the advantages of DKA identified in the main text, we present additional classification results across a wider range of data scales (12.5% to 75%) in Table 9. These results reinforce the key findings, demonstrating that DKA consistently outperforms other baselines across all datasets (COVID, BUSI, ISIC-2019). This superior performance holds not only in extreme low-data regimes (0.63% and 1.25%) and full-data regimes (100%) but also across mid-scale settings, confirming that DKA maintains its advantage regardless of data availability. This performance reflects the effectiveness of its dual-branch design, which simultaneously captures local detail and broader context, providing a critical edge in medical image classification.

Table 9: **Additional Classification Results on Three Datasets with Varying Training Set Ratios.** Experiments are based on the pretrained ViT-B. ACC and SEN are reported as percentages, while F1 is presented as raw value. The best results are highlighted in bold.

Methods	COVID				BUSI				ISIC-2019			
	12.5%	25%	50%	75%	12.5%	25%	50%	75%	12.5%	25%	50%	75%
ACC (%)												
Full Fine-tuning	95.90	97.10	97.92	98.28	85.42	90.01	91.32	93.00	69.81	72.70	76.57	79.06
Linear Probing	92.84	93.43	94.16	94.34	79.23	84.03	86.54	87.22	63.66	65.75	66.60	68.28
BitFit [47]	83.02	90.51	93.57	93.50	69.01	75.00	84.12	90.98	60.96	63.17	66.33	72.10
Prompt [45]	87.32	94.06	96.38	97.53	72.52	80.40	87.91	91.15	62.46	67.24	69.67	75.09
LoRA [46]	88.50	95.88	97.46	98.43	77.45	85.50	90.77	93.25	62.62	68.15	73.47	77.56
Adapter [20]	94.32	96.04	97.14	97.67	80.51	86.18	89.78	90.73	65.54	68.51	70.55	73.41
Adapterformer [41]	90.00	92.19	94.67	97.39	78.00	85.19	87.75	89.37	63.94	69.45	69.73	71.88
Convpass [43]	94.89	96.48	97.32	97.73	81.79	86.89	89.65	90.94	66.96	69.56	71.27	74.25
CIAT [42]	88.98	92.73	94.23	95.45	72.47	79.32	83.56	87.55	59.84	65.16	67.55	70.25
AIM [44]	89.38	93.88	95.76	96.17	76.61	82.00	85.63	86.92	65.62	69.24	70.82	73.58
DKA	**96.86**	**97.34**	**98.29**	**98.55**	**87.26**	**91.10**	**93.73**	**95.01**	**70.47**	**74.04**	**77.42**	**80.06**
F1												
Full Fine-tuning	0.930	0.951	0.956	0.968	0.833	0.874	0.902	0.925	0.402	0.435	0.481	0.502
Linear Probing	0.898	0.916	0.922	0.929	0.782	0.828	0.856	0.861	0.335	0.351	0.379	0.392
BitFit [47]	0.812	0.860	0.903	0.917	0.636	0.743	0.821	0.875	0.276	0.344	0.363	0.411
Prompt [45]	0.848	0.921	0.938	0.953	0.683	0.795	0.862	0.905	0.319	0.384	0.397	0.446
LoRA [46]	0.866	0.934	0.956	0.965	0.747	0.847	0.886	0.932	0.331	0.387	0.439	0.462
Adapter [20]	0.920	0.946	0.958	0.967	0.783	0.842	0.871	0.897	0.354	0.382	0.423	0.457
Adapterformer [41]	0.883	0.905	0.920	0.953	0.767	0.826	0.856	0.873	0.340	0.408	0.416	0.427
Convpass [43]	0.923	0.941	0.958	0.962	0.794	0.840	0.878	0.907	0.372	0.409	0.421	0.458
CIAT [42]	0.843	0.905	0.911	0.925	0.708	0.788	0.813	0.844	0.293	0.364	0.375	0.417
AIM [44]	0.869	0.917	0.928	0.944	0.731	0.813	0.839	0.842	0.334	0.404	0.415	0.425
DKA	**0.949**	**0.958**	**0.970**	**0.978**	**0.856**	**0.891**	**0.931**	**0.943**	**0.407**	**0.456**	**0.488**	**0.514**
SEN (%)												
Full Fine-tuning	93.24	95.16	95.52	96.68	82.24	87.69	89.92	91.05	38.84	40.01	41.56	49.36
Linear Probing	88.72	90.13	90.85	92.06	76.50	81.25	84.31	85.28	32.67	34.81	36.12	38.21
BitFit [47]	79.33	85.27	89.38	91.15	63.85	72.62	80.77	86.31	28.23	35.85	41.62	44.38
Prompt [45]	83.56	91.52	93.24	94.95	68.27	77.25	85.47	90.54	31.54	37.63	43.10	46.75
LoRA [46]	85.15	92.90	95.65	96.22	71.12	81.34	87.93	90.23	31.88	37.31	42.61	46.90
Adapter [20]	91.93	93.65	95.36	96.29	76.53	85.78	87.39	88.03	34.65	37.31	41.16	44.12
Adapterformer [41]	87.35	89.80	91.04	95.54	73.68	82.74	85.79	87.16	32.52	39.10	39.75	41.56
Convpass [43]	92.00	94.51	94.92	95.36	76.78	80.73	85.03	88.21	35.36	39.84	41.58	43.99
CIAT [42]	84.52	89.49	91.72	93.61	69.64	76.49	80.83	84.21	21.79	24.31	36.25	36.50
AIM [44]	86.51	90.22	92.53	93.27	73.86	79.80	83.77	84.27	34.23	37.80	40.65	42.62
DKA	**95.24**	**96.27**	**97.18**	**97.59**	**83.88**	**89.62**	**93.46**	**93.89**	**39.72**	**44.07**	**47.50**	**50.39**

E. Additional Results on Medical-pretrained Models

To provide more comprehensive evidence beyond the main paper, we further report detailed results using medical-pretrained models. Table 10 presents classification performance on the ISIC-2019 dataset with RadImageNet-pretrained ResNet-50 [37], where DKA consistently improves over linear probing and standard adapter tuning across different training ratios in terms of ACC, F1, and SEN. Table 11 reports segmentation results on the BUSI dataset with MedSAM [38], showing that DKA achieves clear gains over both baselines in terms of mIoU and Dice. These extended results corroborate our main findings in Section 4.4, demonstrating that the advantages of DKA generalize robustly to medical-pretrained models and are not restricted to natural-pretrained backbones.

Table 10: **Additional Classification Results based on the RadImageNet-pretrained ResNet-50 backbone on the ISIC-2019 dataset under varying training ratios.** SEN is reported as percentages, while F1 is presented as raw value. The best results are highlighted in bold.

Methods	0.63%	1.25%	100%
F1			
Full Fine-tuning	0.223	0.262	0.483
Linear Probing	0.216	0.236	0.387
BitFit	0.172	0.208	0.411
Prompt	0.200	0.232	0.439
LoRA	0.203	0.230	0.431
Adapter	0.218	0.242	0.440
DKA	**0.230**	**0.278**	**0.498**
SEN (%)			
Full Fine-tuning	22.57	26.86	43.14
Linear Probing	20.28	24.52	37.61
BitFit	17.69	22.86	40.20
Prompt	19.83	24.42	44.05
LoRA	19.79	25.77	44.38
Adapter	20.34	25.13	44.57
DKA	**23.40**	**27.89**	**45.39**

Table 11: **Additional Segmentation Results based on MedSAM backbone on the BUSI segmentation task under varying training ratios.** Results are reported in terms of Dice (%). The best results are highlighted in bold.

Methods	0.63%	1.25%	100%
Dice (%)			
Full Fine-tuning	54.54	63.08	81.15
Linear Probing	50.09	60.90	79.68
BitFit	23.46	47.31	74.15
Prompt	25.19	50.65	78.82
LoRA	32.58	55.03	80.42
Adapter	52.28	61.81	80.35
DKA	**55.39**	**65.24**	**83.97**

F. Additional Ablations

F.1. DKA in Different Backbone

To evaluate whether the advantages of DKA transfer to other architectures, we repeat classification experiments on the pretrained Swin-B using the same datasets: COVID, BUSI, and ISIC-2019. As shown in Figure 9, DKA consistently outperforms full fine-tuning and all PEFT baselines across different data scales. Critically, while other PEFT methods struggle to match linear probing, and often degrade substantially in constrained-data regimes, DKA maintains strong performance in both settings.

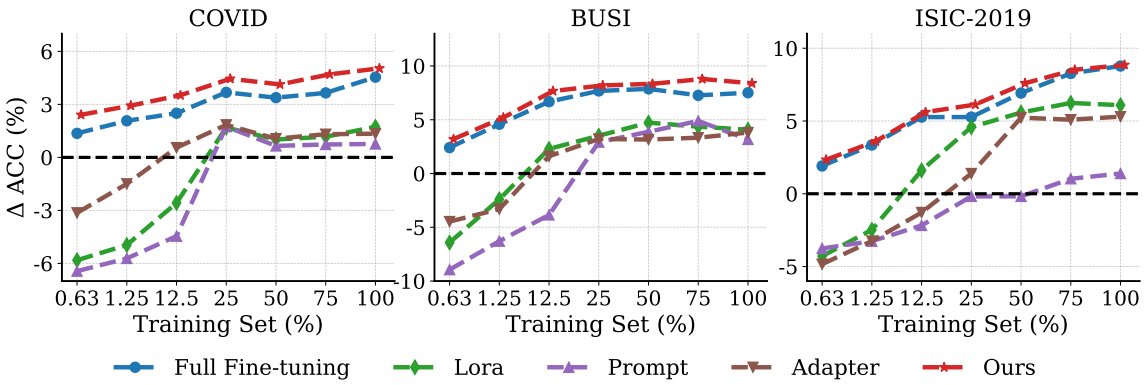

Figure 9: **Performance of Baselines and DKA across Various Training Data Sizes.** ΔACC = $\text{ACC}_{Baslines} - \text{ACC}_{LinearProbing}$. Experiments are based on the pretrained Swin-B for COVID, BUSI, and ISIC-2019.

F.2. Performance of Dual-Kernel Convolution and Asynchronous Learning Rates

To further evaluate the impact of our proposed enhancements, including the introduction of dual-convolution design and a learning rate split strategy within the DKA module, we present additional results in Table 12. This table compares DKA against two baselines (Adapter + Dual-Conv and Adapter + LR Split) across multiple datasets (COVID, BUSI, ISIC-2019) and training set ratios. Consistently, DKA outperforms both baselines, achieving the highest overall accuracy in each dataset, with notable gains observed even in low-data regimes. This confirms the effectiveness of our combined strategy in capturing richer local-global features and improving learning stability.

Table 12: **Performance Comparison of Enhanced Adapter Designs.** Experiments are conducted on the pretrained ViT-B across three medical imaging classification datasets and varying training set ratios by reporting ACC (%).

Datasets	Methods	Training Set Size						
		0.63%	1.25%	12.5%	25%	50%	75%	100%
COVID	Adapter + Dual-Conv	86.78	88.89	96.01	96.98	98.13	97.95	98.61
	Adapter + LR Split	88.04	88.95	96.20	97.04	97.64	98.07	98.79
	Ours	**89.01**	**91.06**	**96.86**	**97.34**	**98.29**	**98.55**	**99.21**
BUSI	Adapter + Dual-Conv	67.73	77.32	83.39	89.14	90.73	91.05	94.25
	Adapter + LR Split	70.37	77.64	84.71	89.46	91.10	93.37	94.93
	Ours	**74.23**	**79.64**	**87.26**	**91.10**	**93.73**	**95.01**	**95.89**
ISIC-2019	Adapter + Dual-Conv	59.49	60.51	70.12	73.27	76.60	78.52	82.25
	Adapter + LR Split	59.06	60.09	69.99	72.24	76.07	78.52	81.67
	Ours	**60.52**	**62.32**	**70.47**	**74.04**	**77.42**	**80.06**	**83.09**

F.3. DKA Position

To understand how the position of inserted DKA influences performance, we explore three placement strategies using the pretrained ViT-B with 12 transformer blocks: inserting DKA into the bottom 4 blocks (Blocks 0–3), middle 4 blocks (Blocks 4–7), and top 4 blocks (Blocks 8–11). We also include a reference setting where DKA is inserted into all layers. We extend our analysis to three medical imaging classification datasets: COVID, BUSI, and ISIC-2019. As shown in Table 13, the position of DKA significantly affects model performance across various training set sizes. Among partial configurations, placing DKA in the middle layers consistently outperforms the top and bottom placements across datasets. Notably, under low-data regimes (e.g., 0.63%), placing DKA in the top layers offers stronger performance than bottom or middle placement, highlighting the value of adapting higher-level representations when supervision is scarce. This trend is consistent with observations from prior work [44], which reported that inserting adapters in bottom blocks yields limited performance.

Table 13: **Effect of Position.** Experiments are conducted on the pretrained ViT-B across three medical imaging classification datasets and varying training set ratios by reporting ACC (%) with standard deviations.

Datasets	Positions	Training Set Size						
		0.63%	1.25%	12.5%	25%	50%	75%	100%
COVID	Bottom	86.04 ± 1.61	88.68 ± 1.38	93.69 ± 0.89	94.29 ± 0.75	95.52 ± 0.60	96.64 ± 0.45	97.34 ± 0.35
	Middle	86.80 ± 1.45	89.18 ± 1.25	95.43 ± 0.53	96.80 ± 0.41	97.76 ± 0.29	98.43 ± 0.25	98.79 ± 0.21
	Top	87.78 ± 1.21	90.50 ± 1.13	94.72 ± 0.69	96.12 ± 055	97.22 ± 0.38	98.23 ± 0.32	98.53 ± 0.26
	ALL	**89.01 ± 0.90**	**91.06 ± 0.84**	**96.86 ± 0.40**	**97.34 ± 0.32**	**98.29 ± 0.26**	**98.55 ± 0.24**	**99.21 ± 0.13**
BUSI	Bottom	71.47 ± 2.06	77.16 ± 1.47	84.86 ± 1.11	88.00 ± 0.65	91.19 ± 0.43	92.79 ± 0.33	93.94 ± 0.27
	Middle	73.48 ± 1.79	78.22 ± 1.28	86.62 ± 0.75	90.43 ± 0.44	93.43 ± 0.25	94.94 ± 0.19	95.46 ± 0.11
	Top	73.86 ± 1.64	78.16 ± 1.22	86.48 ± 0.84	90.08 ± 0.45	93.07 ± 0.28	94.62 ± 0.20	95.00 ± 0.13
	ALL	**74.23 ± 1.53**	**79.64 ± 1.17**	**87.26 ± 0.64**	**91.10 ± 0.39**	**93.73 ± 0.23**	**95.01 ± 0.12**	**95.89 ± 0.09**
ISIC-2019	Bottom	56.87 ± 2.47	58.68 ± 2.11	67.28 ± 1.52	70.37 ± 1.26	73.43 ± 1.05	76.16 ± 0.73	79.52 ± 0.46
	Middle	57.64 ± 2.33	60.01 ± 2.05	69.88 ± 1.34	73.95 ± 1.06	76.74 ± 0.74	79.43 ± 0.43	82.59 ± 0.20
	Top	59.89 ± 2.17	61.79 ± 1.95	69.32 ± 1.38	72.59 ± 1.18	76.04 ± 0.79	78.42 ± 0.52	81.90 ± 0.25
	ALL	**60.52 ± 2.02**	**62.32 ± 1.85**	**70.47 ± 1.26**	**74.04 ± 0.99**	**77.42 ± 0.67**	**80.06 ± 0.36**	**83.09 ± 0.14**

F.4. ERF of Adapter in Different Backbone

To further validate our findings on the impact of data scarcity on Adapter performance, we extend the effective receptive field (ERF) analysis to the pretrained Swin-T model [27], as shown in Figure 10. This complementary analysis reinforces our observations from the pretrained ViT-B model (Figure 2), revealing a similar contraction of ERFs as the training set size decreases. Specifically, under extreme data scarcity (e.g., 0.63% and 1.25% training data), the pretrained Swin-T exhibits a sharply reduced ERF, aligning with the earlier findings (see Section 2) that Adapters can disrupt pretrained feature representations under severe data constraints. This result suggests that the negative impacts observed in the main text are not limited to a single architecture but are likely a more general phenomenon affecting a wide range of vision backbones.

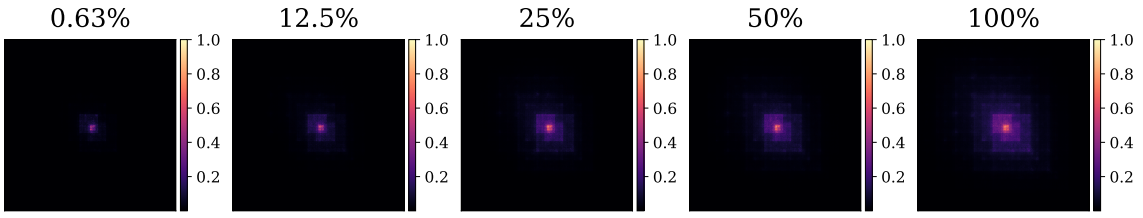

Figure 10: **Effective Receptive Field of Adapter under Different Training Set Ratios.** Experiments are conducted on the COVID dataset using the pretrained Swin-T.

F.5. Effect of Dilated vs. Standard Kernels

We further compare dilated convolutional kernels with our standard large-kernel design. In particular, we evaluate several dilated kernel settings, including 3×3 with dilation rates of $d = 3, 25$, 11×11 with $d = 5$, and 26×26 with $d = 2$, against the dual-kernel configuration of $5 \times 5 + 51 \times 51$. The classification results on the BUSI dataset are shown in Table 14, and the segmentation results on the ISIC-2018 dataset are reported in Table 15.

Across both datasets, the $5 \times 5 + 51 \times 51$ design consistently achieves the best performance in terms of ACC, F1, and SEN for classification, as well as mIoU and Dice for segmentation. Although dilated kernels provide a larger effective receptive field, they underperform compared to explicitly using a large standard kernel. This suggests that our dual-kernel design captures both local and global information more effectively than dilated alternatives, highlighting the efficiency of standard large kernels in the `DKA` framework.

Table 14: **Comparison of different kernel designs on the BUSI dataset for classification under varying training ratios.** Experiments are based on the pretrained ViT-B. ACC and SEN are reported as percentages, while F1 is presented as raw value. The best results are highlighted in bold. "*d*" represents the dilation rate in dilated convolution.

Kernel Designs	0.63%	1.25%	100%
ACC (%)			
3×3 (*d*=3) + 3×3 (*d*=25)	66.77	77.32	93.61
3×3 (*d*=3) + 11×11 (*d*=5)	67.41	77.64	94.25
3×3 (*d*=3) + 26×26 (*d*=2)	68.69	77.96	94.37
5×5 + 51×51	**74.23**	**79.46**	**95.89**
F1			
3×3 (*d*=3) + 3×3 (*d*=25)	0.641	0.751	0.931
3×3 (*d*=3) + 11×11 (*d*=5)	0.650	0.756	0.936
3×3 (*d*=3) + 26×26 (*d*=2)	0.663	0.758	0.942
5×5 + 51×51	**0.720**	**0.774**	**0.951**
SEN (%)			
3×3 (*d*=3) + 3×3 (*d*=25)	62.81	74.35	92.36
3×3 (*d*=3) + 11×11 (*d*=5)	63.99	74.43	93.44
3×3 (*d*=3) + 26×26 (*d*=2)	64.72	74.71	94.03
5×5 + 51×51	**71.82**	**76.78**	**95.46**

Table 15: **Comparison of different kernel designs on the ISIC-2018 dataset for segmentation under varying training ratios.** Experiments are based on the pretrained Segmenter-B. mIoU and Dice are reported as percentages. The best results are highlighted in bold. "*d*" represents the dilation rate in dilated convolution.

Kernel Designs	0.63%	1.25%	100%
mIoU (%)			
3×3 (*d*=3) + 3×3 (*d*=25)	60.25	72.71	76.43
3×3 (*d*=3) + 11×11 (*d*=5)	61.46	73.11	76.97
3×3 (*d*=3) + 26×26 (*d*=2)	62.27	73.43	77.48
5×5 + 51×51	**63.13**	**74.27**	**78.53**
Dice (%)			
3×3 (*d*=3) + 3×3 (*d*=25)	75.19	84.20	86.64
3×3 (*d*=3) + 11×11 (*d*=5)	76.13	84.46	86.99
3×3 (*d*=3) + 26×26 (*d*=2)	76.75	84.68	87.31
5×5 + 51×51	**77.39**	**85.24**	**87.97**

F.6. Effect of Diverse Kernel Combinations

To further analyze the contribution of different kernel configurations in DKA, we evaluate multiple kernel combinations for both classification and segmentation tasks. Specifically, we compare three settings: (i) $5 \times 5 + 11 \times 11 + 51 \times 51$, (ii) $5 \times 5 + 31 \times 31 + 51 \times 51$, and (iii) $5 \times 5 + 51 \times 51$. Table 16 reports results on the BUSI classification dataset, while Table 17 presents results on the ISIC-2018 segmentation dataset.

Across both tasks, we observe that the dual-kernel configuration $(5 \times 5 + 51 \times 51)$ consistently achieves the best trade-off, outperforming the three-branch alternatives in terms of ACC, F1, and SEN for classification, as well as mIoU and Dice for segmentation. These results indicate that adding intermediate kernels (e.g., 11×11 or 31×31) does not provide additional benefits, and a simpler dual-kernel design is sufficient to capture both local and global dependencies. This further validates the efficiency of our proposed kernel selection strategy within DKA.

Table 16: **Comparison of diverse kernel combinations on the BUSI dataset for classification under varying training ratios.** Experiments are based on the pretrained ViT-B. ACC and SEN are reported as percentages, while F1 is presented as raw value. The best results are highlighted in bold.

Kernel Combinations	0.63%	1.25%	100%
ACC (%)			
5×5 +11×11 + 51×51	72.68	77.00	94.89
5×5 +31×31 + 51×51	71.04	76.68	94.57
5×5 + 51×51	**74.23**	**79.46**	**95.89**
F1			
5×5 +11×11 + 51×51	0.709	0.743	0.942
5×5 +31×31 + 51×51	0.694	0.738	0.936
5×5 + 51×51	**0.720**	**0.774**	**0.951**
SEN (%)			
5×5 +11×11 + 51×51	69.17	73.56	94.11
5×5 +31×31 + 51×51	68.39	72.85	93.62
5×5 + 51×51	**71.82**	**76.78**	**95.46**

Table 17: **Comparison of diverse kernel combinations on the ISIC-2018 dataset for segmentation under varying training ratios.** Experiments are based on the pretrained Segmenter-B. mIoU and Dice are reported as percentages. The best results are highlighted in bold.

Kernel Combinations	0.63%	1.25%	100%
mIoU (%)			
5×5 +11×11 + 51×51	62.72	73.76	78.06
5×5 +31×31 + 51×51	62.40	73.63	77.54
5×5 + 51×51	**63.13**	**74.27**	**78.53**
Dice (%)			
5×5 +11×11 + 51×51	77.09	84.90	87.68
5×5 +31×31 + 51×51	76.84	84.81	87.35
5×5 + 51×51	**77.39**	**85.24**	**87.97**

F.7. Effect under Extreme Low-data Regimes

To further investigate the limits of DKA, we conduct experiments under an extreme low-data regime with only 0.125% of the training set available. This setup approximately corresponds to a 5-shot setting for COVID classification and genuine 1-shot settings for BUSI and ISIC-2019 (classification), as well as BRATS, BUSI, and ISIC-2018 (segmentation). The results are reported in Table 18 and Table 19.

Across all datasets, DKA consistently outperforms both Linear Probing and standard Adapter, even under such highly constrained supervision. For example, on ISIC-2019 classification, DKA improves ACC from 52.54% (Linear Probing) to 55.95%, and F1 from 0.227 to 0.263. Similarly, on BUSI segmentation, DKA achieves a Dice of 37.43%, substantially higher than Linear Probing (23.02%) and Adapter (24.93%). These results confirm that the proposed large-kernel design remains robust and effective even in the most challenging few-shot scenarios, highlighting its practicality for data-scarce medical applications.

Table 18: **Comparison of Linear Probing, Adapter, and DKA on three classification datasets under 0.125% training data.** Experiments are based on the pretrained ViT-B. ACC and SEN are reported as percentages, while F1 is presented as raw value. The best results are highlighted in bold.

Methods	COVID	BUSI	ISIC-2019
ACC (%)			
Linear Probing	76.51	37.06	52.54
Adapter	71.38	35.46	47.25
DKA	**78.74**	**39.62**	**55.95**
F1			
Linear Probing	0.735	0.273	0.227
Adapter	0.704	0.247	0.153
DKA	**0.787**	**0.340**	**0.263**
SEN (%)			
Linear Probing	72.53	33.87	21.58
Adapter	68.40	30.41	15.07
DKA	**75.64**	**37.13**	**24.39**

Table 19: **Comparison of Linear Probing, Adapter, and DKA on three segmentation datasets under 0.125% training data.** Experiments are based on the pretrained Segmenter-B. mIoU and Dice are reported as percentages. The best results are highlighted in bold.

Methods	BRATS	BUSI	ISIC-2018
mIoU (%)			
Linear Probing	3.25	19.64	53.21
Adapter	1.20	14.24	47.05
DKA	**6.36**	**32.83**	**60.25**
Dice (%)			
Linear Probing	6.94	23.02	69.46
Adapter	2.40	24.93	63.99
DKA	**11.96**	**37.43**	**75.19**

F.8. Effect on Medical-pretrained Vision-Language Models.

To further validate the generalization ability of DKA, we extend our evaluation to medical vision-language models. Specifically, we adopt MedCLIP [93] as the backbone and follow the few-shot image classification protocol commonly used in prior works on vision-language adaptation [94, 95]. We report results on the BUSI dataset under 1-shot, 4-shot, and 8-shot settings, where each configuration is averaged over five random seeds. As summarized in Table 20, DKA consistently surpasses linear probing and standard adapter tuning across all support sizes. These results demonstrate that the benefits of DKA are not limited to vision-only large pretrained models, but also extend to multimodal vision-language models, highlighting its robustness in broader medical AI scenarios.

Table 20: **Comparison of different methods under few-shot settings on the BUSI dataset based on MedCLIP.** Results are reported in terms of ACC, F1, and SEN.

Tuning Strategies	1-shot	4-shot	8-shot
ACC (%)			
Linear Probing	66.25	74.21	78.41
Adapter	65.86	74.08	79.32
DKA	**71.20**	**79.82**	**82.27**
F1			
Linear Probing	0.631	0.712	0.776
Adapter	0.625	0.706	0.782
DKA	**0.697**	**0.762**	**0.815**
SEN (%)			
Linear Probing	62.55	69.72	77.45
Adapter	61.87	69.46	78.96
DKA	**68.17**	**75.85**	**81.87**

F.9. Frequency-domain Analysis of Kernel Sizes

To better understand the complementary roles of small and large kernels in DKA, we analyze their frequency responses using the radial power spectral density (PSD). Specifically, we compute the spectral centroid f_c and the normalized frequency f_{90} that captures 90% of the cumulative energy. Table 21 shows the results for 5×5 and 51×51 kernels. The smaller kernel exhibits a higher spectral centroid ($f_c = 0.618$ vs. 0.503), indicating stronger sensitivity to high-frequency details. In contrast, the larger kernel shifts energy towards lower frequencies, facilitating global context modeling. This analysis provides further evidence for the effectiveness of combining diverse kernel sizes in DKA.

Table 21: Frequency-domain analysis of kernel sizes. Results are reported as mean±std over different trained models.

Kernel Size	f_c	f_{90}
5×5	0.618 ± 0.018	0.924 ± 0.011
51×51	0.503 ± 0.004	0.904 ± 0.003

F.10. Inference Latency and Memory Usage

We also report the inference efficiency of different methods in terms of latency and memory consumption on the BUSI dataset using the pretrained ViT-B. As summarized in Table 22, DKA introduces only marginal overhead compared to the standard adapter framework, with inference latency increasing by less than 0.5 ms and memory usage by less than 6 MB. These differences are negligible in practice, indicating that the proposed large-kernel design achieves substantial performance gains with minimal computational cost.

Table 23: **Comparison of methods on the BUSI segmentation dataset with batch size $= 4$ based on the pretrained Segmenter-B.**

Methods	0.63%	1.25%	100%
Full Fine-tuning	27.02	33.11	57.91
Linear Probing	25.81	32.47	54.60
Adapter	18.67	26.50	55.46
DKA	**27.26**	**35.21**	**59.35**

Table 22: **Comparison of inference latency and memory usage on the BUSI dataset based on the pretrained ViT-B.** DKA introduces only negligible overhead compared to standard adapter variants.

Methods	**Inference Latency (ms)**	**Memory (MB)**
Linear Probing	6.78	352.81
Adapter	11.54	433.70
Adapter + 5×5 Conv	11.66	433.90
Adapter + 51×51 Conv	11.69	439.56
DKA	**11.97**	**439.63**

F.11. Effects of Larger Batch Size

We further conducted additional segmentation experiments using a larger batch size of 4 under the same configuration. As shown in Table 23, increasing the batch size does not change the relative ranking: DKA consistently outperforms Full Fine-tuning, Linear Probing, and Adapter across all label ratios. This confirms that our improvements are robust to batch-size variations and are not tied to a specific optimization setting.

F.12. Parameter Efficiency Analysis

The DKA module integrates two depthwise convolution branches with kernel sizes k_1 and k_2 within each adapter. These convolutions are applied independently on each of the $\hat{d}$ channels, contributing $\hat{d}(k_1^2 + k_2^2)$ parameters per module. Since DKA is inserted twice in every Transformer block (after the attention and feedforward layers), the total number of additional parameters grows linearly with the number of blocks. Even with two kernels (e.g., $k_1 = 51$, $k_2 = 5$), the total number of trainable parameters introduced by all DKA modules remains less than 2% of the pretrained backbone, maintaining strong parameter efficiency while providing strong performance gains, especially in low-data regimes.

F.13. Complementary CAM Visualization

To complement the ERF analysis, we additionally compare Grad-CAM [96] responses of the standard Adapter and our DKA on BUSI. As shown in Figure 11, the standard Adapter exhibits diffuse and background-driven activations, whereas DKA produces coherent, lesion-centered responses. This confirms that the enlarged ERF of DKA captures meaningful contextual information rather than merely expanding activation range.

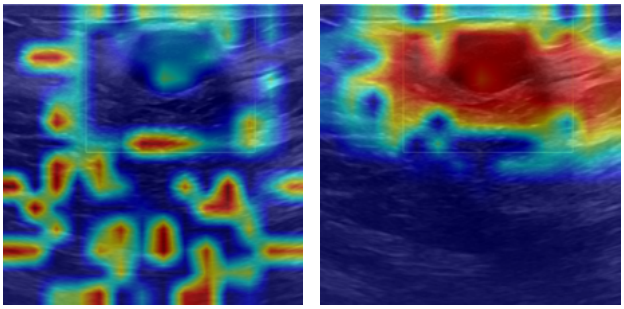

Figure 11: **Grad-CAM comparison on BUSI.** Left: standard Adapter; right: `DKA` with more focused, lesion-related activations.

G. Pseudocode

Algorithm 1 Pseudo-code of a Transformer block with DKA

```
class DKA:
    def __init__(self, dim, middle_dim, kernel_large, kernel_small):
        self.downsample = Linear(dim, middle_dim)
        self.conv_large = DepthwiseConv(middle_dim, kernel_large)
        self.conv_small = DepthwiseConv(middle_dim, kernel_small)
        self.activation = GELU()
        self.upsample = Linear(middle_dim, dim)

    def forward(self, x):
        # Store the input for the residual connection
        residual = x
        x = self.downsample(x)

        # Dual-Path Convolutions (Large + Small)
        x_large = self.conv_large(x)
        x_small = self.conv_small(x)
        x = x_large + x_small

        x = self.activation(x)
        x = self.upsample(x)
        x = x + residual

        return x

class TransformerBlock_with_DKA:
    def __init__(self, dim, num_heads, mlp_ratio, middle_dim, kernel_large, kernel_small):
        # Original ViT components
        self.attn = MultiheadAttention(dim, num_heads)
        self.norm1 = LayerNorm(dim)
        self.mlp = MLP(dim, mlp_ratio)
        self.norm2 = LayerNorm(dim)

        # DKA Adapter
        self.dka = DKA(dim, middle_dim, kernel_large, kernel_small)

    def forward(self, x):
        residual = x
        x = self.norm1(x)
        x = self.attn(x)
        x = x + residual

        x = self.dka(x)

        residual = x
        x = self.norm2(x)
        x = self.mlp(x)
        x = x + residual

        x = self.dka(x)

        return x
```