TELESCOPEMIX: DEPTH-AWARE SPATIAL AND CHANNEL ADAPTATION OF VISION TRANSFORMERS FOR MEDICAL IMAGING

Anonymous authors
Paper under double-blind review

ABSTRACT

Adapting pretrained Vision Transformers to medical imaging is challenging when target datasets are small, imbalanced, and drawn from modalities that differ substantially from natural images. We introduce TeleScopeMix, a parameter-efficient adapter that combines depthwise spatial filtering, efficient cross-channel mixing, and a depth-dependent capacity schedule. Our analysis of supervised and self-supervised backbones indicates that adapter utilization generally increases toward later Transformer blocks, motivating greater channel-mixing capacity at depth. Across various medical-image classification datasets and Vision Transformer backbones, TeleScopeMix with optimizing fewer than 1% of the backbone parameters achieves comparable or even better performance compared to full fine-tuning method.

1 INTRODUCTION

Vision Transformers (ViTs) (Dosovitskiy et al., 2021) have emerged as powerful alternatives to convolutional neural networks for medical-image classification. By processing images as sequences of patch tokens, ViTs can model long-range dependencies and global context. Pretrained Transformer representations have achieved strong classification performance across diverse medical imaging modalities, including dermoscopy, histopathology, chest radiography, and gastrointestinal imaging (Aruk et al., 2026; Wang et al., 2022; Regmi et al., 2023), establishing them as promising foundations for specialized medical applications.

Specifically, we can finetune a model pre-trained on large-scale natural image datasets through transfer learning. By reusing these learned representations, we see sharp improvements in overall performance (Raghu et al., 2019; Matsoukas et al., 2022). Adapting large pretrained ViTs to medical imaging nevertheless presents practical challenges. Medical datasets are often considerably smaller than the natural-image corpora used for pretraining, exhibit severe class imbalance, and differ in image statistics across modalities. Fully fine-tuning every backbone parameter is therefore computationally expensive and may introduce excessive trainable capacity relative to the available target-domain data. Parameter-efficient fine-tuning (PEFT) instead freezes most of the pretrained backbone and optimizes a small set of task-specific parameters. Methods such as low-rank adaptation (LoRA) (Hu et al., 2021) substantially reduce the number of trainable parameters, but primarily modify existing feature transformations and do not explicitly introduce the two-dimensional locality bias required for medical imaging. This motivates an alternative form of parameter-efficient adaptation that operates directly on the spatial organization of image tokens.

Convolutional adapters provide one way to introduce spatial inductive bias without modifying the internal attention operations of a pretrained Transformer. In particular, depthwise convolutions apply an independent spatial filter to each feature channel and consequently require substantially fewer parameters than dense convolutions. Prior work has shown that lightweight depthwise-convolutional bypasses can inject local spatial information into Transformer representations with limited parameter overhead (Zhang et al., 2025).

Main Contributions. Our contributions are threefold. First, we introduce ScopeMix, a uniformly allocated depth-wise adapter. We compare ScopeMix across supervised and self-supervised ViTs

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

and show a general tendency of stronger adapter utilization toward later Transformer blocks. Second, we introduce TeleScopeMix, a parameter-efficient convolutional adapter that combines depth-wise spatial filtering, explicit channel mixing, and a depth-dependent capacity schedule. We evaluate ScopeMix and TeleScopeMix on HAM10000 against current methods, and show that they improve predictive performance while maintaining trainable-parameter efficiency. Third, we evaluate TeleScopeMix across multiple medical-image domains spanning radiography, MRI, endoscopy, blood smear, ultrasounds, and histopathology. Across these diverse domains, we show that TeleScopeMix remains competitive with full fine-tuning while optimizing fewer than 1% of backbone parameters.

2 Related Work

2.1 Vision Transformers

Vision Transformers (ViTs) (Dosovitskiy et al., 2021) have become widely used image encoders, with architectures such as DeiT (Touvron et al., 2021) and Swin Transformer (Liu et al., 2021) improving their data efficiency and scalability. MOHSA (Zhang et al., 2024) explores the head overlap to improve the performance of Vision Transformer models. In parallel, self-supervised methods including MAE (He et al., 2022), iBOT (Zhou et al., 2022), and DINOv3 (Siméoni et al., 2025) have enabled ViTs to learn transferable visual representations from unlabeled data.

We evaluate lightweight convolutional adapters for finetuning frozen ViT backbones, and compare performance across supervised and self-supervised pretraining paradigms.

2.2 Convolution-Enhanced Adaptation of Vision Transformers

Vision Transformers are effective at modeling interactions across an entire image, but they do not inherently emphasize neighboring pixels or patches as strongly as convolutional networks. This limitation can become important when training examples are scarce. Prior hybrid architectures address it in different ways. CvT (Wu et al., 2021) uses convolution during token formation and attention projection. ConViT (d'Ascoli et al., 2021) encourages attention to begin with a local spatial preference. LocalViT (Li et al., 2023) introduces convolution processing within the Transformer's feature transformation stage. Although their implementations differ, these studies support the general idea that combining local feature extraction with global token interaction can improve learning when data availability is constrained.

Convolution can also serve as a separate adaptation path without extensively changing a pretrained transformer. (Jie et al., 2024) introduces compact convolutional branches while retaining the pretrained ViT backbone, allowing adaptation to be concentrated in a small set of parameters. (Zhang et al., 2025) connects transformer blocks through depth-wise convolutional shortcuts to provide an additional route for local spatial information. The method proposed by (Zhang et al., 2025) is designed to improve the data efficiency and faster convergence of Vision Transformers by training the augmented architecture from scratch.

In this work, we focus on a parameter-efficient fine-tuning mechanism for pretrained Vision Transformers and its uses in medical imaging.

3 Methodology

A standard Vision Transformer (ViT) processes an input sequence composed of patch embeddings together with optional special tokens. Let $\mathbf{X}_n \in \mathbb{R}^{(N+S)\times D}$ denote the input to Transformer block n, where N is the number of image patch tokens, D is the embedding dimension, and S is the number of special tokens. These special tokens may include a class token and, depending on the architecture, one or more register tokens. The input sequence may therefore be written as

$$\mathbf{X}_n = \left[\mathbf{x}_{\text{cls}}, \mathbf{x}_{\text{reg}}^{(1)}, \ldots, \mathbf{x}_{\text{reg}}^{(R)}, \mathbf{x}_1, \ldots, \mathbf{x}_N\right], \tag{1}$$

where $X_{\text{patch}}^{(n)} = \mathbf{x}_1, \ldots, \mathbf{x}_N$ denote the image patch tokens.

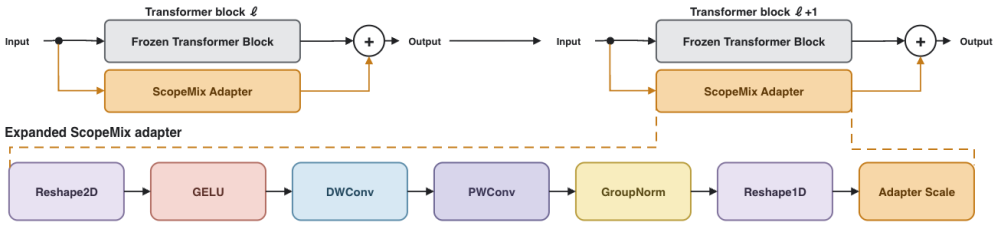


Figure 1: Overview of ScopeMix, a lightweight convolutional bypass that introduces spatial and channel mixing around each frozen Transformer block and combines its output through a learned residual scale.

3.1 Proposed Adaptation

We adapt a fully frozen Transformer backbone with the following depthwise adapter. Recall from Eq. (1), the hidden representation entering Transformer block ℓ is partitioned into S special tokens and $N = HW$ patch tokens, denoted $X^{(n)}_{\text{patch}}$.

Following Zhang et al. (2025), we define $\text{Reshape}_{1\to2}$ as a reshape operation which maps the sequence of N patch embeddings to their corresponding $H \times W$ spatial arrangement. The $\text{Reshape}_{2\to1}$ operation converts back to its 1-D representation. We further extend Zhang et al. (2025)'s formulation through three modifications: (1) normalization after spatial filtering and channel mixing, (2) parameter-efficient PWConv to introduce cross-channel interactions, and (3) learnable layerwise residual scaling to regulate the adapter contribution. Figure 1 illustrates the ScopeMix adapter and its integration into a frozen transformer block.

$$V^{(n)} = \text{GELU}\Big(\text{Reshape}_{1\to2}\Big(X^{(n)}_{\text{patch}}\Big)\Big) \tag{2}$$

$$C^{(n)}_{\text{2D}} = \text{Norm}\Big(\text{PWConv}\Big(\text{DWConv}\Big(V^{(n)}\Big)\Big)\Big). \tag{3}$$

$$X^{(n+1)}_{\text{patch}} = f_{\text{enc}}(X_n)_{\text{patch}} + \alpha_n \,\text{Reshape}_{2\to1}\Big(C^{(n)}_{\text{2D}}\Big). \tag{4}$$

3.1.1 Normalization

Firstly, we move normalization to after the depthwise convolution. The depthwise convolution instead receives the activated patch representation directly, allowing the spatial filtering operation to act directly on the activated Transformer features. Normalization is then applied after spatial filtering and channel mixing, stabilizing the combined adapter output immediately before it is reshaped and injected into the Transformer residual stream. Furthermore, we choose normalization using GroupNorm instead of BatchNorm. For group g with mean μ_g and variance σ^2_g over its elements,

$$C_{c,h,w} = \gamma_c \frac{P_{c,h,w} - \mu_{g(c)}}{\sqrt{\sigma^2_{g(c)} + \epsilon}} + \delta_c, \qquad \mathcal{A}_\ell(U) = C. \tag{5}$$

Following Wu & He (2018), we apply Group Normalization after the depthwise convolution. GroupNorm is preferred over BatchNorm because adapter training uses small batches and its statistics are batch-independent at both training and evaluation time, removing the train/eval statistic mismatch that a batch-dependent normalizer introduces under heavy class-balanced resampling.

3.1.2 Pointwise Convolution

The inclusion of a PWConv operation is motivated by Chollet (2017), as depthwise convolutions operate independently on each channel and therefore do not model cross-channel interactions. The

depthwise-filtered representation is passed through a pointwise 1×1 convolution. However, a full PWConv costs $d^2 + d$ parameters and would be by far the largest term in the adapter. We instead consider two alternatives to full PWConv: grouped pointwise mix and low-rank pointwise mix.

Grouped PWConv. We implement the pointwise group convolution introduced in ShuffleNet (Zhang et al., 2018b) and partition the d channels into G contiguous groups, and mix only within groups. A smaller G yields fewer, larger groups and therefore more cross-channel mixing at higher parameter cost. With $M \in \mathbb{R}^{d \times d/G}$ and $g(c)$ the group of channel c,

$$P_{c,h,w} \;=\; \sum_{c' \in \mathcal{G}_{g(c)}} M_{c,c'}\, Y_{c',h,w} \;+\; b_c, \qquad \text{cost} \;=\; G\left(\frac{d}{G}\right)^2 + d \;=\; \frac{d^2}{G} + d. \tag{6}$$

The two extremes recover familiar operations: $G = 1$ is a single group spanning all channels, i.e. a dense PWConv costing $d^2 + d$, while $G = d$ places every channel in its own group and performs no cross-channel mixing at all. The diagram in Appendix A.1 illustrates this process.

Low-Rank Pointwise Convolution. Instead of using a dense weight matrix $\mathbf{W} \in \mathbb{R}^{d \times d}$, we propose replacing the dense matrix with a rank-r factorization,

$$\mathbf{W} \approx \mathbf{W}_1 \mathbf{W}_2, \qquad \mathbf{W}_1 \in \mathbb{R}^{d \times r}, \quad \mathbf{W}_2 \in \mathbb{R}^{r \times d}, \quad r \ll d. \tag{7}$$

Given a feature vector $\mathbf{x}_{h,w} \in \mathbb{R}^d$ at spatial location (h, w), the low-rank PWConv is therefore

$$\mathrm{LRPWConv}(\mathbf{x}_{h,w}) = (\mathbf{x}_{h,w} \mathbf{W}_1)\, \mathbf{W}_2. \tag{8}$$

The first projection compresses the channel representation from d to r dimensions, while the second projects it back to d dimensions. This reduces the number of PWConv weights from d^2 to $2dr$, where rank r directly controls the capacity. The diagram in Appendix A.2 illustrates this projection.

3.1.3 Layerwise Adapter Scaling

As shown in Eq. (4), the adapter output at Transformer block ℓ is modulated by a learnable residual coefficient α_ℓ before being added to the frozen encoder representation. We initialize α_ℓ to 10^{-4} so that training begins close to the pretrained model. This limits the initial magnitude of the task-specific perturbation and reduces the risk that untrained adapter activations overwhelm the pretrained representation. During optimization, each coefficient can increase or decrease independently, allowing the model to determine the required adaptation strength at each depth. More detail regarding adapter scale is written in Appendix B.

3.2 Depth-Wise Adapter Utilization

A central question in parameter-efficient adaptation is whether each Transformer block requires the same amount of trainable capacity. We examine depth-wise adapter utilization by measuring the representational changes induced at each block. To control for differences in adapter capacity, we first train a uniform configuration of ScopeMix in which every block receives an adapter with the same architecture and parameter budget.

Centered kernel alignment. We use centered kernel alignment (CKA) (Kornblith et al., 2019) to compare the pretrained and adapted representations. For a fixed evaluation set, let H_ℓ^{pre} and H_ℓ^{adapt} denote the representations produced after block ℓ by the pretrained backbone and adapted model, respectively. Their layerwise similarity is defined as

$$\mathrm{CKA}_\ell = \mathrm{CKA}\left(H_\ell^{\mathrm{pre}}, H_\ell^{\mathrm{adapt}}\right). \tag{9}$$

Given the centered kernel matrices $\mathbf{K}_\ell$ and $\mathbf{L}_\ell$ constructed from these representations, kernel CKA is computed as

$$\mathrm{CKA}(\mathbf{K}_\ell, \mathbf{L}_\ell) = \frac{\mathrm{HSIC}(\mathbf{K}_\ell, \mathbf{L}_\ell)}{\sqrt{\mathrm{HSIC}(\mathbf{K}_\ell, \mathbf{K}_\ell)\, \mathrm{HSIC}(\mathbf{L}_\ell, \mathbf{L}_\ell)}}. \tag{10}$$

Higher CKA indicates that adaptation largely preserves the pretrained representation, whereas lower CKA indicates greater representational divergence. Since absolute CKA values may differ across backbones because of architecture, pretraining objective, and overall domain shift, we primarily analyze how CKA changes with depth within each model. Layer-wise CKA profiles provide measures of where adaptation is concentrated in the Transformer.

3.3 TELESCOPING CHANNEL-MIXING SCHEDULE

The depth-wise utilization analysis motivates allocating adaptation capacity non-uniformly across the Transformer. We define TeleScopeMix, a ScopeMix-based adapter. It uses the same architecture as ScopeMix (Figure 1), and varies only the amount of cross-channel adaptation. The diagram in Appendix A.3 illustrates the allocation strategy.

We divide the L Transformer blocks into three approximately equal depth regions. The earliest third receives the lowest-capacity grouped pointwise configuration, i.e. the largest number of groups, 4 times the number of groups used in the final tier. The middle third uses 2 times the number of groups of the final tier. The final third receives the highest-capacity configuration, using the fewest number of groups. Because parameter cost scales as d^2/G 6, using fewer groups in the deeper blocks concentrates the more expensive cross-channel operation exactly where models exhibit the strongest learned adapter utilization, while the many-group early blocks remain inexpensive.

4 EXPERIMENTS AND RESULTS

4.1 DATASETS AND EVALUATION METRICS

We primarily evaluate our approach on the HAM10000 (Tschandl et al., 2018) dataset. This dataset is the most imbalanced dataset in our evaluation, and is a good middle ground for dataset size.

To further evaluate generalization across medical imaging domains, we consider eight additional classification datasets spanning radiography, magnetic resonance imaging (MRI), Optical Coherence Tomography (OCT), histopathology, etc. These datasets span diverse imaging distributions, including RGB images, stain-specific histopathology and blood smear images, and grayscale MRI and X-ray scans. The datasets range from 1,600 to 327,680 images and include binary and multi-class tasks with balanced and imbalanced class distributions. Table 1 summarizes their modalities and sizes. Datasets are presented in ascending order of size.

Table 1: Summary of the medical image classification datasets. A dataset is considered imbalanced when its largest class contains more than 50% more instances than its smallest class. Dataset size includes all training, validation, and test samples.

Dataset	Size	Modality	Class Distribution
RetinaMNIST (Yang et al., 2023)	1,600	Fundus Camera	Imbalanced
Kvasir (Pogorelov et al., 2017)	4,000	Gastrointestinal endoscopy	Balanced
PneumoniaMNIST (Yang et al., 2023)	5,856	Chest X-Ray	Imbalanced
BRISC2025 (Fateh et al., 2026)	6,000	Brain MRI	Semi-Balanced
HAM10000 (Tschandl et al., 2018)	10,015	Dermatoscopic Images	Imbalanced
BloodMNIST (Yang et al., 2023)	17,092	Blood Cell Microscope	Imbalanced
PathMNIST (Yang et al., 2023)	107,180	Colorectal histopathology	Imbalanced
OCTMNIST (Yang et al., 2023)	109,309	Retinal OCT	Imbalanced
PatchCamelyon (Veeling et al., 2018; Ehteshami Bejnordi et al., 2017)	327,680	Lymph-node histopathology	Balanced

4.2 TRAINING AND IMPLEMENTATION DETAILS

Optimization. For the HAM10000 experiments, we use Base-sized backbones to evaluate ScopeMix and TeleScopeMix at higher model capacity. For the cross-dataset evaluation, we instead use DINOv3-Small to assess generalization with a more compact backbone. All configurations are trained for 100 epochs at a resolution of 384×384 with a batch size of 16. We use AdamW, cosine annealing with warm restarts (Loshchilov & Hutter, 2017), and cross-entropy loss with label smoothing $\varepsilon = 0.05$.

Full fine-tuning uses learning rates of 2×10^{-5} for the backbone and 10^{-4} for the classifier. LoRA uses 2×10^{-5} for its low-rank parameters, whereas convolutional adapters use 10^{-2}; adapter scales, normalization parameters, and classification heads use 10^{-4}. We use weighted random sampling to reduce class imbalance across the dataset. Then, we employ RandAugment (Cubuk et al., 2020), random cropping, horizontal flipping, rotation, color jitter, random erasing, Mixup ($\alpha = 0.2$) (Zhang et al., 2018a), and CutMix ($\alpha = 1.0$) (Yun et al., 2019) to further augment the data. Finally, input images are normalized channel-wise using training set statistics. Models were selected by validation accuracy and evaluated over five seeds; results report test-set mean $\pm$ standard deviation.

Table 2: Comparison of full fine-tuning and parameter-efficient adaptation methods on the HAM10000 test set. Precision, recall, F1, and AUROC are macro-averaged across the seven diagnostic classes. Trainable parameters include the classification head and all method-specific parameters. The top 2 results in each column is shown in bold.

Method	Accuracy	Bal. Acc.	Macro AUROC	Macro Prec.	Macro F1	Trainable Params. (M)
Full fine-tuning	$\mathbf{0.9348 \pm 0.0028}$	0.8514 ± 0.0160	0.9704 ± 0.0072	$\mathbf{0.9235 \pm 0.0097}$	$\mathbf{0.8828 \pm 0.0078}$	85.67
LoRA ($r = 8$)	0.8573 ± 0.0090	0.8523 ± 0.0066	0.9796 ± 0.0008	0.7465 ± 0.0101	0.7885 ± 0.0080	0.30
LoRA ($r = 16$)	0.8701 ± 0.0119	0.8522 ± 0.0212	0.9812 ± 0.0033	0.7692 ± 0.0208	0.8025 ± 0.0219	0.60
LoRA ($r = 32$)	0.8774 ± 0.0072	0.8633 ± 0.0136	0.9831 ± 0.0016	0.7769 ± 0.0116	0.8114 ± 0.0117	1.19
DWConv (Zhang et al., 2025)	0.8686 ± 0.0075	0.8373 ± 0.0137	0.9803 ± 0.0022	0.7802 ± 0.0081	0.8007 ± 0.0083	0.11
Grouped ScopeMix ($G = 16$)	0.9183 ± 0.0047	0.8631 ± 0.0150	$\mathbf{0.9869 \pm 0.0023}$	0.8673 ± 0.0140	0.8619 ± 0.0107	0.52
Grouped ScopeMix ($G = 8$)	$\mathbf{0.9262 \pm 0.0050}$	$\mathbf{0.8685 \pm 0.0133}$	0.9863 ± 0.0028	$\mathbf{0.8816 \pm 0.0108}$	$\mathbf{0.8706 \pm 0.0130}$	0.93
Low-rank ScopeMix ($r = 8$)	0.9089 ± 0.0067	0.8578 ± 0.0079	$\mathbf{0.9872 \pm 0.0012}$	0.8501 ± 0.0130	0.8508 ± 0.0086	0.25
Low-rank ScopeMix ($r = 16$)	0.9111 ± 0.0038	$\mathbf{0.8688 \pm 0.0079}$	0.9850 ± 0.0022	0.8495 ± 0.0147	0.8560 ± 0.0055	0.39

4.3 Comparison of Fine-Tuning and Adaptation Methods

We first compare full fine-tuning against several parameter-efficient adaptation strategies. The evaluated configurations include full fine-tuning, LoRA with ranks $r \in \{8, 16, 32\}$, the DWConv adapter of Zhang et al. (2025), Grouped ScopeMix with $G \in \{8, 16\}$ groups, and Low-rank ScopeMix with ranks $r \in \{8, 16\}$. These experiments evaluate the classification performance and parameter efficiency of adapting a pretrained DINOv3 backbone to the HAM10000 dataset.

All configurations use the same data partitions, training augmentations, optimization schedule, and evaluation procedure described in Section 4.2.

Overall classification performance. Table 2 compares the predictive performance of the nine configurations. Among the evaluated methods, Grouped ScopeMix with $G = 8$ achieves the best balance in overall performance and compute efficiency, obtaining a second best accuracy of 92.62% and balanced accuracy of 86.85%. Relative to full fine-tuning, it improves balanced accuracy by 1.71 percentage points and macro AUROC by 1.65 percentage points, all while optimizing only 1.07% of total model parameters. Compared with alternative PEFT configurations, Grouped ScopeMix achieves the highest overall performance while keeping similar parameter efficiency.

Effect of channel mixing. The DWConv adapter (Zhang et al., 2025) independently processes each feature channel and therefore only provides spatial adaptation. Both grouped and low-rank PWConv mixing supplement this spatial processing with learnable channel interaction. The performance improvement of ScopeMix over the DWConv adapter shows that explicit channel mixing is beneficial when adapting pretrained Transformer features to dermatoscopic imagery.

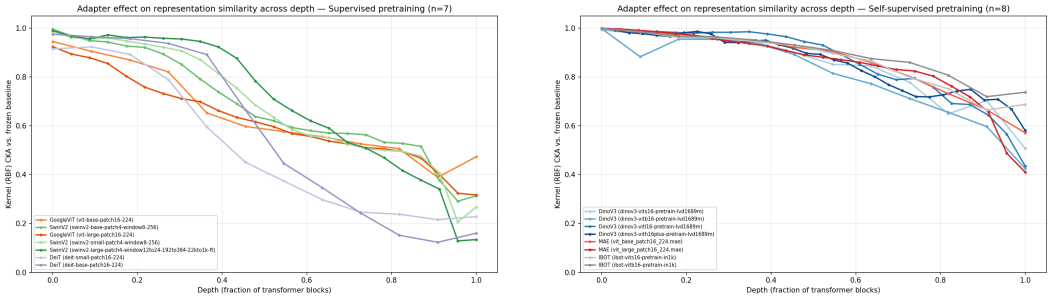
4.4 Layerwise Representation Analysis and Telescoping Motivation

A central question in adapting pretrained Transformers is whether parameters should be allocated uniformly across blocks or varied according to depth. To motivate our telescoping strategy, we measure how strongly the adapted representations diverge from those of the frozen pretrained backbone at successive layers.

Pretrained–Adapted Representation Similarity Across Depth Figure 2 reports the layerwise kernel CKA similarity between the frozen pretrained backbone and its adapted counterpart.

Across the evaluated backbones, pretrained–adapted similarity generally decreases nonlinearly with depth. This pattern is strongest in the self-supervised backbones, whose CKA profiles remain relatively high through the earlier blocks before declining sharply at around two-thirds of total transformer depth. The supervised backbones exhibit a sharp decline much earlier in depth at roughly one-thirds of total transformer depth, with another sharp decline at roughly two-thirds of total depth. The nonlinear decline of CKA across depth suggests that later blocks do undergo greater representation change, and motivates the idea that later blocks require more parameter allocation.

Motivation for telescoping adapter capacity. The learned adapter-scale analysis in Appendix B and CKA profiles jointly suggest that adaptation is distributed non-uniformly across Transformer



(a) Supervised pretraining.

(b) Self-supervised pretraining.

Figure 2: Layerwise kernel CKA similarity between the original pretrained Transformer and its adapted counterpart. Higher values indicate that adaptation preserves the pretrained representation, while lower values indicate greater representational divergence.

depth. In brief, the learned residual scales tend to assign greater weight to later adapter branches, while CKA indicates greater representational divergence in later blocks.

We therefore introduce a telescoping schedule that uses smaller groups and lower-capacity PWConv in the early layers and progressively larger groups toward the later layers. The subsequent ablation evaluates whether this allocation preserves predictive performance.

4.5 TeleScopeMix Analysis

The preceding experiments identify Grouped ScopeMix with $G = 8$ as the strongest non-telescoping adaptation configuration. We next investigate whether varying channel-mixing granularity across Transformer depth in a *telescoping* manner further improves this adapter. We compare a uniform grouped PWConv configuration against the proposed telescoping configuration while keeping the backbone, training protocol, adapter placement, and remaining hyperparameters fixed.

In the non-telescoping configuration, every adapter uses eight groups. In the telescoping configuration, adapters in the first, middle, and final thirds of the Transformer use 32, 16, and 8 groups, respectively. Fewer groups allow for more mixing, providing more restricted, cheaper channel interaction in earlier layers and progressively denser interaction in deeper layers, where the representations are expected to encode more task-specific semantic information.

Table 3: Ablation of TeleScopeMix and ScopeMix adapters on the HAM10000 dataset. Precision, recall, F1, and AUROC are macro-averaged.

Configuration	Group Schedule	Accuracy	Bal. Acc.	Macro AUROC	Macro Prec.	Macro F1	Trainable Params. (M)
Non-telescoping	8/8/8	**0.9262 ± 0.0050**	0.8685 ± 0.0133	**0.9863 ± 0.0028**	0.8816 ± 0.0108	0.8706 ± 0.0130	0.93
Telescoping	32/16/8	0.9260 ± 0.0035	**0.8820 ± 0.0115**	0.9860 ± 0.0024	**0.8823 ± 0.0106**	**0.8800 ± 0.0093**	**0.56**

Effect of telescoping. As shown in Table 3, TeleScopeMix achieves comparable performance relative to the uniform grouped configuration. It increases balanced accuracy from 86.85% to 88.20%, macro precision from 88.16% to 88.23%, and macro F1 from 87.06% to 88.00%, while reducing the number of trainable parameters by an additional 39.80%. These results support the proposed depth-dependent allocation strategy: fewer adaptation parameters are assigned to earlier Transformer blocks, whose representations change less during task adaptation, while greater channel-mixing capacity is retained in later blocks, where more substantial task-specific adaptation is required. Representative TeleScopeMix attention maps are shown in Figure 3. Further comparison of TeleScopeMix and ScopeMix is found in C.

4.6 Depthwise Adaptation Across Supervised and Self-Supervised ViTs

We evaluate the generality of ScopeMix adaptation across four ViT families with different architectural and pretraining characteristics: supervised DeiT and SwinViT models, and self-supervised

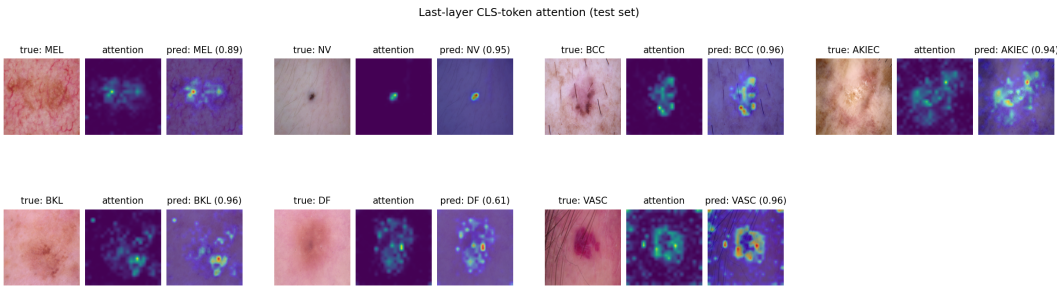


Figure 3: Representative attention maps produced by the proposed TeleScopeMix adapter.

iBOT and DINOv3 models. For each backbone, we compare full fine-tuning, ScopeMix adaptation, and TeleScopeMix. This comparison examines both the effectiveness of spatially aware parameter-efficient adaptation across model families and whether the optimal placement of adaptation capacity depends on the backbone's pretraining objective.

Table 4: Performance of full fine-tuning, ScopeMix, and TeleScopeMix across supervised and self-supervised Transformer backbones. Results are reported as the mean $\pm$ standard deviation over five runs. AUROC, precision, and F1 are macro-averaged. The best predictive performance for each backbone and metric is shown in bold, while bold parameter counts indicate the most parameter-efficient configuration.

Backbone	Configuration	Group Schedule	Accuracy	Bal. Acc.	Macro AUROC	Macro Prec.	Macro F1	Trainable Params. (M)
DeiT	Full fine-tuning	—	$\mathbf{0.8910 \pm 0.0054}$	0.7952 ± 0.0165	0.9477 ± 0.0051	$\mathbf{0.8705 \pm 0.0162}$	$\mathbf{0.8270 \pm 0.0163}$	85.80
	ScopeMix	8/8/8	0.8742 ± 0.0019	$\mathbf{0.8142 \pm 0.0145}$	$\mathbf{0.9720 \pm 0.0030}$	0.7971 ± 0.0119	0.8026 ± 0.0111	0.93
	TeleScopeMix	32/16/8	0.8669 ± 0.0034	0.7939 ± 0.0086	0.9715 ± 0.0029	0.7834 ± 0.0131	0.7835 ± 0.0059	**0.56**
SwinViT	Full fine-tuning	—	$\mathbf{0.9217 \pm 0.0048}$	$\mathbf{0.8407 \pm 0.0089}$	0.9650 ± 0.0039	$\mathbf{0.9019 \pm 0.0104}$	$\mathbf{0.8669 \pm 0.0035}$	86.90
	ScopeMix	8/8/8	0.9019 ± 0.0026	0.8391 ± 0.0079	0.9712 ± 0.0042	0.8479 ± 0.0105	0.8397 ± 0.0074	1.04
	TeleScopeMix	32/16/8	0.8997 ± 0.0034	0.8357 ± 0.0072	$\mathbf{0.9753 \pm 0.0025}$	0.8440 ± 0.0123	0.8369 ± 0.0088	**0.79**
iBOT ViT	Full fine-tuning	—	$\mathbf{0.9014 \pm 0.0043}$	0.7973 ± 0.0105	0.9575 ± 0.0066	$\mathbf{0.8854 \pm 0.0110}$	$\mathbf{0.8317 \pm 0.0059}$	85.80
	ScopeMix	8/8/8	0.8941 ± 0.0040	$\mathbf{0.8336 \pm 0.0113}$	$\mathbf{0.9818 \pm 0.0027}$	0.8320 ± 0.0152	0.8290 ± 0.0096	0.93
	TeleScopeMix	32/16/8	0.8877 ± 0.0041	0.8155 ± 0.0061	0.9808 ± 0.0011	0.8257 ± 0.0033	0.8147 ± 0.0058	**0.56**
DINOv3	Full fine-tuning	—	$\mathbf{0.9348 \pm 0.0028}$	0.8514 ± 0.0160	0.9704 ± 0.0072	$\mathbf{0.9235 \pm 0.0097}$	$\mathbf{0.8828 \pm 0.0078}$	85.67
	ScopeMix	8/8/8	0.9262 ± 0.0050	0.8685 ± 0.0133	$\mathbf{0.9863 \pm 0.0028}$	0.8816 ± 0.0108	0.8706 ± 0.0130	0.93
	TeleScopeMix	32/16/8	0.9260 ± 0.0035	$\mathbf{0.8820 \pm 0.0115}$	0.9860 ± 0.0024	0.8823 ± 0.0106	0.8800 ± 0.0093	**0.56**

Table 4 compare full fine-tuning, ScopeMix, and TeleScopeMix across the four evaluated backbones. A complete evaluation table is found in Appendix D.

ScopeMix performs particularly well with the iBOT ViT and DINOv3 backbones, approaching full fine-tuning in accuracy while achieving comparable or higher balanced accuracy and macro AUROC. For SwinViT and DeiT, ScopeMix performs slightly below full fine-tuning in accuracy, macro precision, and macro F1. Nevertheless, ScopeMix and TeleScopeMix consistently achieve higher macro AUROC than full fine-tuning across all evaluated backbones, suggesting that the local spatial inductive bias improves class-wise discrimination and generalization.

Across every backbone, TeleScopeMix performs comparably to the non-telescoping ScopeMix configuration while using fewer trainable parameters. Although minor metric-level differences remain, the telescoping schedule largely preserves ScopeMix's predictive performance across both supervised and self-supervised pretrained models. Finally, based on the full evaluation table in Appendix D, both ScopeMix and TeleScopeMix outperform LoRA based finetuning across every backbone.

Together with the kernel CKA results in Figure 2, these findings demonstrate that allocating channel-mixing capacity according to network depth substantially improves parameter efficiency while preserving predictive performance across different architectures and pretraining regimes.

4.7 Cross-Dataset Medical Imaging Evaluation

We further evaluate TeleScopeMix on additional medical imaging datasets spanning a range of modalities and sizes (Table 1). Across the cross-dataset benchmarks, TeleScopeMix demonstrates strong generalization across diverse medical imaging modalities and dataset sizes (Table 5). On

the two large histopathology datasets PathMNIST and PatchCamelyon, TeleScopeMix outperforms full fine-tuning across every reported metric while updating fewer than 1% of the model parameters. Furthermore, it matches performance on the smaller BloodMNIST dataset. The strong results on PathMNIST, PatchCamelyon, and BloodMNIST suggest that TeleScopeMix's DWConv-based adaptation effectively captures local spatial relationships in stain-specific medical images.

On Kvasir, a smaller endoscopy dataset, TeleScopeMix still performs extremely well, achieving higher accuracy, precision, and F1 score compared to full fine-tuning, with marginally lower AUROC and balanced accuracy.

This performance extends beyond color imagery. On the BRISC 2025 brain MRI dataset, TeleScopeMix remains within one percentage point of full fine-tuning across all reported metrics. It further outperforms full fine-tuning on both the large OCTMNIST dataset and the smaller PneumoniaMNIST X-ray dataset. These results demonstrate that TeleScopeMix remains effective under substantial modality shifts from natural-image pretraining to grayscale MRI, OCT, and X-ray imagery. Finally, its strong performance across datasets of varying sizes suggests that TeleScopeMix remains effective in both limited-data and data-rich settings.

Taken together, the results show that TeleScopeMix is a competitive and substantially more parameter-efficient alternative to full fine-tuning across dataset sizes and imaging modalities. Its strongest gains emerge on large histopathology datasets, where the target signal is highly spatial and localized, while its competitive performance on smaller datasets suggests that it does not require abundant labeled data to adapt effectively. Overall, parameter-efficient spatial adaptation can successfully bridge the domain gap between natural-image pretraining and medical imaging, in some cases surpassing full fine-tuning while updating roughly two orders of magnitude fewer parameters.

Table 5: Classification performance of full fine-tuning and TeleScopeMix across the medical imaging datasets. Despite training only approximately 1% as many parameters as full fine-tuning, TeleScopeMix achieves comparable or superior performance. Results are reported as the mean $\pm$ standard deviation over five runs, with the best mean result for each dataset and metric shown in bold.

Dataset	Configuration	Accuracy	Bal. Acc.	Macro AUROC	Macro Prec.	Macro F1
RetinaMNIST	Full finetuning	$\mathbf{0.5260 \pm 0.0202}$	0.4219 ± 0.0143	0.7514 ± 0.0040	$\mathbf{0.4551 \pm 0.0777}$	0.4186 ± 0.0202
	TeleScopeMix	0.5225 ± 0.0112	$\mathbf{0.4424 \pm 0.0219}$	$\mathbf{0.7696 \pm 0.0045}$	0.4484 ± 0.0299	$\mathbf{0.4255 \pm 0.0259}$
Kvasir	Full fine-tuning	0.9447 ± 0.0058	$\mathbf{0.9463 \pm 0.0059}$	$\mathbf{0.9970 \pm 0.0006}$	0.9465 ± 0.0056	$\mathbf{0.9461 \pm 0.0059}$
	TeleScopeMix	$\mathbf{0.9450 \pm 0.0074}$	0.9461 ± 0.0070	0.9968 ± 0.0005	$\mathbf{0.9470 \pm 0.0069}$	$\mathbf{0.9461 \pm 0.0071}$
PneumoniaMNIST	Full finetuning	0.9365 ± 0.0083	0.9193 ± 0.0114	$\mathbf{0.9865 \pm 0.0014}$	0.9472 ± 0.0055	0.9303 ± 0.0095
	TeleScopeMix	$\mathbf{0.9401 \pm 0.0029}$	$\mathbf{0.9249 \pm 0.0056}$	0.9864 ± 0.0024	$\mathbf{0.9485 \pm 0.0017}$	$\mathbf{0.9345 \pm 0.0036}$
BRISC2025	Full fine-tuning	$\mathbf{0.9946 \pm 0.0033}$	$\mathbf{0.9950 \pm 0.0029}$	$\mathbf{0.9997 \pm 0.0005}$	$\mathbf{0.9952 \pm 0.0033}$	$\mathbf{0.9951 \pm 0.0031}$
	TeleScopeMix	0.9936 ± 0.0026	0.9943 ± 0.0023	0.9994 ± 0.0005	0.9942 ± 0.0024	0.9943 ± 0.0023
BloodMNIST	Full finetuning	$\mathbf{0.9783 \pm 0.0015}$	$\mathbf{0.9761 \pm 0.0021}$	$\mathbf{0.9987 \pm 0.0002}$	$\mathbf{0.9767 \pm 0.0020}$	$\mathbf{0.9762 \pm 0.0018}$
	TeleScopeMix	0.9719 ± 0.0013	0.9706 ± 0.0028	0.9986 ± 0.0003	0.9698 ± 0.0023	0.9699 ± 0.0016
PathMNIST	Full fine-tuning	0.9386 ± 0.0031	0.9203 ± 0.0044	0.9874 ± 0.0052	0.9199 ± 0.0046	0.9174 ± 0.0047
	TeleScopeMix	$\mathbf{0.9428 \pm 0.0032}$	$\mathbf{0.9256 \pm 0.0033}$	$\mathbf{0.9934 \pm 0.0018}$	$\mathbf{0.9256 \pm 0.0033}$	$\mathbf{0.9229 \pm 0.0040}$
OCTMNIST	Full finetuning	0.8164 ± 0.0068	0.8164 ± 0.0068	0.9483 ± 0.0061	0.8444 ± 0.0086	0.7990 ± 0.0083
	TeleScopeMix	$\mathbf{0.8488 \pm 0.0148}$	$\mathbf{0.8488 \pm 0.0148}$	$\mathbf{0.9767 \pm 0.0019}$	$\mathbf{0.8621 \pm 0.0126}$	$\mathbf{0.8400 \pm 0.0181}$
PatchCamelyon	Full fine-tuning	0.9098 ± 0.0058	0.9097 ± 0.0058	0.9715 ± 0.0029	0.9173 ± 0.0031	0.9093 ± 0.0060
	TeleScopeMix	$\mathbf{0.9156 \pm 0.0048}$	$\mathbf{0.9155 \pm 0.0048}$	$\mathbf{0.9729 \pm 0.0018}$	$\mathbf{0.9204 \pm 0.0034}$	$\mathbf{0.9153 \pm 0.0049}$

5 Conclusion

We introduced ScopeMix and TeleScopeMix, convolutional adapters for parameter-efficient adaptation of pretrained Vision Transformers to medical-image classification. ScopeMix combines depthwise spatial filtering with efficient grouped channel mixing, introducing both local spatial bias and cross-channel interaction while leaving the pretrained backbone frozen. TeleScopeMix further improves parameter efficiency by varying this channel-mixing capacity across network depth.

Experiments on various medical imaging benchmarks show that TeleScopeMix remains competitive with full fine-tuning across diverse dataset sizes and imaging modalities while optimizing fewer than 1% of the backbone parameters. We hope that our work inspires more work for lightweight adaptation for real world applications and tasks.

AI DISCLOSURE

In this work, we used generative AI tools for aiding and polishing writing. We have not used generative AI tools for discovery, research ideation, drafting sections, or generating synthetic datasets. The rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for identifying gaps in experiments as well as editing the research paper to improve readability. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Ibrahim Aruk, Ishak Pacal, and Ahmet Nusret Toprak. A comprehensive comparison of convolutional neural network and visual transformer models on skin cancer classification. *Computational Biology and Chemistry*, 120:108713, 2026. doi: 10.1016/j.compbiolchem.2025.108713.

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1251–1258, 2017.

Ekin D. Cubuk, Barret Zoph, Jon Shlens, and Quoc V. Le. RandAugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 702–703, 2020.

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. In *International Conference on Learning Representations (ICLR)*, 2021.

Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *International conference on machine learning*, pp. 2286–2296. PMLR, 2021.

Babak Ehteshami Bejnordi et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *JAMA*, 318(22):2199–2210, 2017. doi: 10.1001/jama.2017.14585. URL https://doi.org/10.1001/jama.2017.14585.

Amirreza Fateh, Yasin Rezvani, Sara Moayedi, Sadjad Rezvani, Fatemeh Fateh, Mansoor Fateh, and Vahid Abolghasemi. Brisc: Annotated dataset for brain tumor segmentation and classification. *Scientific Data*, 13(1), February 2026. ISSN 2052-4463. doi: 10.1038/s41597-026-06753-y. URL http://dx.doi.org/10.1038/s41597-026-06753-y.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16000–16009, 2022.

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

Shibo Jie, Zhi-Hong Deng, Shixuan Chen, and Zhijuan Jin. Convolutional bypasses are better vision transformer adapters. In *ECAI 2024: 27th European Conference on Artificial Intelligence, 19–24 October 2024, Santiago de Compostela, Spain–Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024)*, pp. 202–209. SAGE Publications Pvt. Ltd 1 Oliver's Yard, 55 City Road, London, EC1Y 1SP, 2024.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 3519–3529, 2019.

Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, Michele Magno, Luca Benini, and Luc Van Goo. Localvit: Analyzing locality in vision transformers. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9598–9605. IEEE, 2023.

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 (ICCV)*, pp. 10012–10022, 2021.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

Christos Matsoukas, Johan Fredin Haslum, Moein Sorkhei, Magnus Söderberg, and Kevin Smith. What makes transfer learning work for medical images: Feature reuse & other factors. *arXiv preprint arXiv:2203.01825*, 2022.

Konstantin Pogorelov, Kristin Ranheim Randel, Carsten Griwodz, Sigrun Losada Eskeland, Thomas de Lange, Dag Johansen, Concetto Spampinato, Duc-Tien Dang-Nguyen, Mathias Lux, Peter Thelin Schmidt, Michael Riegler, and Pål Halvorsen. Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, MMSys'17, pp. 164–169, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5002-0. doi: 10.1145/3083187.3083212.

Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pp. 3342–3352, 2019.

Smriti Regmi, Aliza Subedi, Ulas Bagci, and Debesh Jha. Vision transformer for efficient chest x-ray and gastrointestinal image classification. *arXiv preprint arXiv:2304.11529*, 2023. doi: 10.48550/arXiv.2304.11529.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning (ICML)*, pp. 10347–10357, 2021.

Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5(1): 180161, 2018.

Bastiaan S. Veeling, Jasper Linmans, Jim Winkens, Taco S. Cohen, and Max Welling. Rotation equivariant CNNs for digital pathology. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, pp. 210–218. Springer, 2018. doi: 10.1007/978-3-030-00934-2_24. URL https://doi.org/10.1007/978-3-030-00934-2_24.

Xiyue Wang, Sen Yang, Jun Zhang, Minghui Wang, Jing Zhang, Junzhou Huang, Wei Yang, and Xiao Han. Transformer-based unsupervised contrastive learning for histopathological image classification. *Medical Image Analysis*, 81:102559, 2022. doi: 10.1016/j.media.2022.102559.

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22–31. IEEE, 2021.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018.

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6023–6032, 2019.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018a.

Tianxiao Zhang, Bo Luo, and Guanghui Wang. Improving vision transformers by overlapping heads in multi-head self-attention. *arXiv preprint arXiv:2410.14874*, 2024.

Tianxiao Zhang, Wenju Xu, Bo Luo, and Guanghui Wang. Depth-wise convolutions in vision transformers for efficient training on small datasets. *Neurocomputing*, 617:128998, 2025.

Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6848–6856, 2018b.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. iBOT: Image BERT pre-training with online tokenizer. In *International Conference on Learning Representations*, 2022.

A Additional Diagrams

A.1 Grouped PWConv Diagram

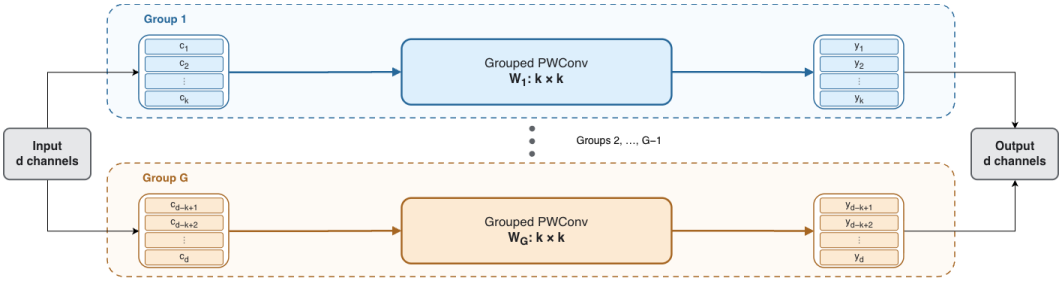


Figure 4: Grouped pointwise convolution (PWConv) divides d input channels into G groups of $k = d/G$ channels. A separate 1×1 convolution mixes channels within each group, producing d output channels without mixing across groups.

A.2 Low Rank PWConv Diagram



Figure 5: Low-rank pointwise convolution projects d input channels into an r-channel bottleneck, then expands them back to d output channels using two 1×1 convolutions, where $r \ll d$.

A.3 TeleScopeMix Diagram

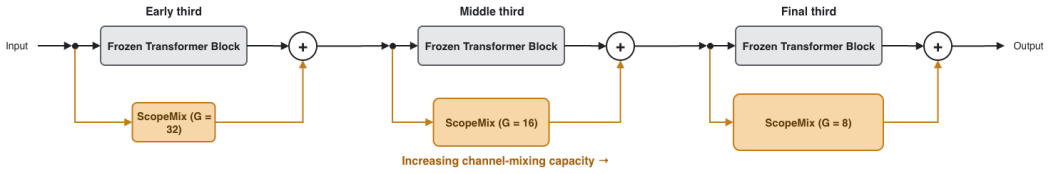
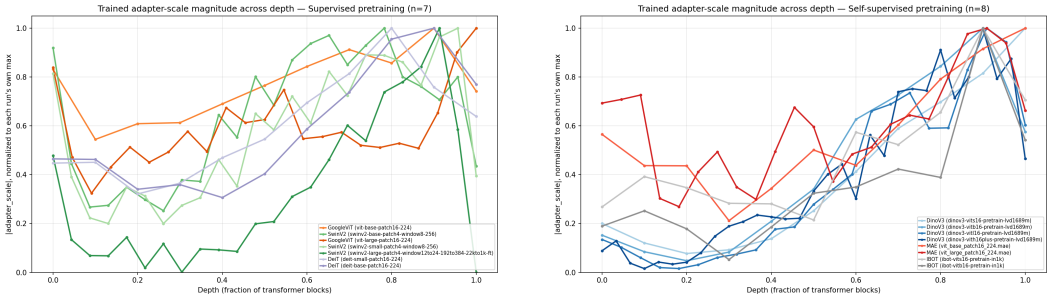


Figure 6: TeleScopeMix allocates greater channel-mixing capacity to later Transformer blocks by reducing the number of ScopeMix groups from 32 to 16 to 8 across network depth.

B Adapter Scale Analysis

As defined in Equation 4, the output of the adapter at transformer block n is incorporated into the frozen encoder representation through the learnable residual scale α_n. To analyze depth-wise utilization, we measure the adapter scale value at every transformer block, and use its magnitude as a measure for how much influence the adapter block has at each transformer block.

Adapter-scale analysis. Figure 7 presents the learned adapter-scale magnitudes across transformer depth for the supervised and self-supervised backbones. Across the self-supervised models evaluated in this work, including DINO, MAE, and iBOT, the learned scales generally increase toward the later transformer blocks. For iBOT and DINO, this is especially pronounced, with a clear linear trend until around 90% of the overal depth. This trend indicates that the optimization process assigns greater weight to adapter outputs at greater depths. Compared to the self-supervised backbones, the supervised ViT variants are much more varied. Compared with the self-supervised models, the GoogleViTs show a much smaller upward trend throughout the depth. The SwinV2



(a) Supervised pretraining.

(b) Self-supervised pretraining.

Figure 7: Learned adapter-scale magnitudes across transformer depth for supervised and self-supervised pretrained backbones. Larger magnitudes indicate that the corresponding adapter output receives greater weight when combined with the frozen transformer-block output. Depth is normalized by the total number of blocks to facilitate comparison across architectures.

and DeiT follow a trend closer to the self-supervised models, tending to increase throughout depth. Although the precise magnitude and shape of the profiles vary by backbone, the repeated late-layer concentration across independently pretrained model backbones suggests that a uniform parameter allocation may devote unnecessary capacity to their earlier blocks.

Table 6: Full Evaluation table of supervised and self-supervised Transformer backbones. Results are reported as the mean ± standard deviation over five runs. AUROC, precision, and F1 are macro-averaged. The best predictive performance for each backbone and metric is shown in bold.

Backbone	Configuration	Group Schedule	Accuracy	Bal. Acc.	Macro AUROC	Macro Prec.	Macro F1	Trainable Params. (M)
DeiT	Full fine-tuning	—	**0.8910 ± 0.0054**	0.7952 ± 0.0165	0.9477 ± 0.0051	**0.8705 ± 0.0162**	**0.8270 ± 0.0163**	85.80
	LoRA Rank 16	—	0.7487 ± 0.0108	0.7354 ± 0.0189	0.9534 ± 0.0041	0.5624 ± 0.0115	0.6059 ± 0.0131	0.60
	LoRA Rank 32	—	0.7678 ± 0.0147	0.7628 ± 0.0225	0.9589 ± 0.0036	0.5867 ± 0.0268	0.6397 ± 0.0233	1.19
	ScopeMix	8/8/8	0.8742 ± 0.0019	**0.8142 ± 0.0145**	**0.9720 ± 0.0030**	0.7971 ± 0.0119	0.8026 ± 0.0111	0.93
	TeleScopeMix	32/16/8	0.8669 ± 0.0034	0.7939 ± 0.0086	0.9715 ± 0.0029	0.7834 ± 0.0131	0.7835 ± 0.0059	0.56
SwinViT	Full fine-tuning	—	**0.9217 ± 0.0048**	**0.8407 ± 0.0089**	0.9650 ± 0.0039	**0.9019 ± 0.0104**	**0.8669 ± 0.0035**	86.90
	LoRA Rank 16	—	0.8266 ± 0.0063	0.8067 ± 0.0143	0.9689 ± 0.0008	0.6927 ± 0.0152	0.7386 ± 0.0155	0.78
	LoRA Rank 32	—	0.8361 ± 0.0059	0.8207 ± 0.0153	0.9718 ± 0.0009	0.7177 ± 0.0170	0.7586 ± 0.0161	1.55
	ScopeMix	8/8/8	0.9019 ± 0.0026	0.8391 ± 0.0079	0.9712 ± 0.0042	0.8479 ± 0.0105	0.8397 ± 0.0074	1.04
	TeleScopeMix	32/16/8	0.8997 ± 0.0034	0.8357 ± 0.0072	**0.9753 ± 0.0025**	0.8440 ± 0.0123	0.8369 ± 0.0088	0.79
iBOT ViT	Full fine-tuning	—	**0.9014 ± 0.0043**	0.7973 ± 0.0105	0.9575 ± 0.0066	**0.8854 ± 0.0110**	**0.8317 ± 0.0059**	85.80
	LoRA Rank 16	—	0.8116 ± 0.0104	0.7854 ± 0.0078	0.9686 ± 0.0020	0.6956 ± 0.0219	0.7303 ± 0.0160	0.60
	LoRA Rank 32	—	0.8261 ± 0.0087	0.8025 ± 0.0128	0.9732 ± 0.0020	0.7259 ± 0.0209	0.7543 ± 0.0163	1.19
	ScopeMix	8/8/8	0.8941 ± 0.0040	**0.8336 ± 0.0113**	**0.9818 ± 0.0027**	0.8320 ± 0.0152	0.8290 ± 0.0096	0.93
	TeleScopeMix	32/16/8	0.8877 ± 0.0041	0.8155 ± 0.0061	0.9808 ± 0.0011	0.8257 ± 0.0033	0.8147 ± 0.0058	0.56
DINOv3	Full fine-tuning	—	**0.9348 ± 0.0028**	0.8514 ± 0.0160	0.9704 ± 0.0072	**0.9235 ± 0.0097**	**0.8828 ± 0.0078**	85.67
	LoRA Rank 16	—	0.8701 ± 0.0119	0.8522 ± 0.0212	0.9812 ± 0.0033	0.7692 ± 0.0208	0.8025 ± 0.0219	0.60
	LoRA Rank 32	—	0.8774 ± 0.0072	0.8633 ± 0.0136	0.9831 ± 0.0016	0.7769 ± 0.0116	0.8114 ± 0.0117	1.19
	ScopeMix	8/8/8	0.9262 ± 0.0050	0.8685 ± 0.0133	**0.9863 ± 0.0028**	0.8816 ± 0.0108	0.8706 ± 0.0130	0.93
	TeleScopeMix	32/16/8	0.9260 ± 0.0035	**0.8820 ± 0.0115**	0.9860 ± 0.0024	0.8823 ± 0.0106	0.8800 ± 0.0093	0.56

C HAM10000 Evaluation Figures

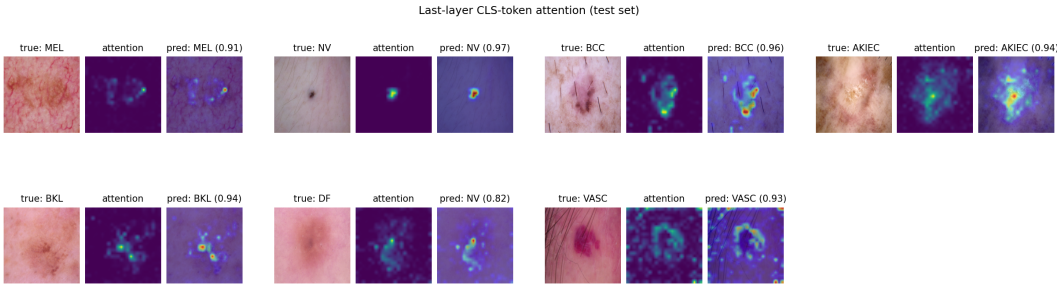
C.1 Attention Map Comparison

Figure 8 presents the attention map for the best ScopeMix configuration based on classification results in Table 2 and the attention map for the proposed TeleScopeMix adapter.

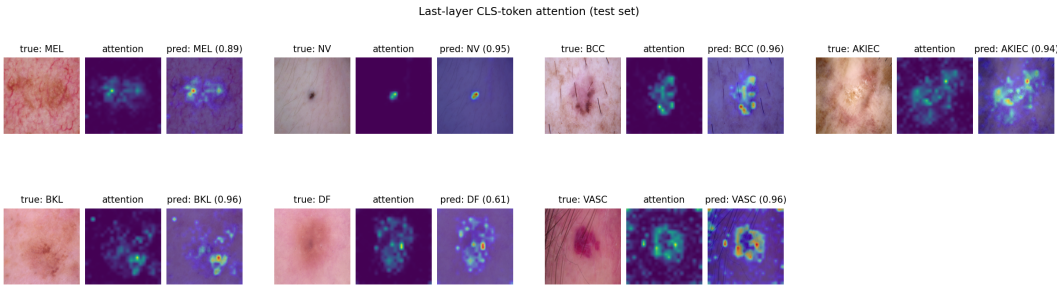
Compared with ScopeMix, TeleScopeMix produces more spatially concentrated attention patterns across the visualized examples. When comparing the outputs on the NV, BCC, and VASC classes, TeleScopeMix exhibits smaller, more localized attention activations. Furthermore, for BCC and VASC examples, these activations also appear to align closer to the lesion itself, wheras ScopeMix distributes attention more broadly across the images. These observations suggest that telescoping may encourage more localized use of lesion-specific features.

C.2 Confusion Matrix Comparison

Figure 9 presents the confusion matrices for Grouped ScopeMix and TeleScopeMix on the test set.

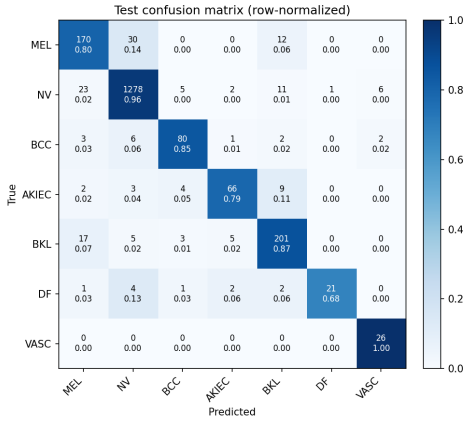


(a) Grouped ScopeMix.

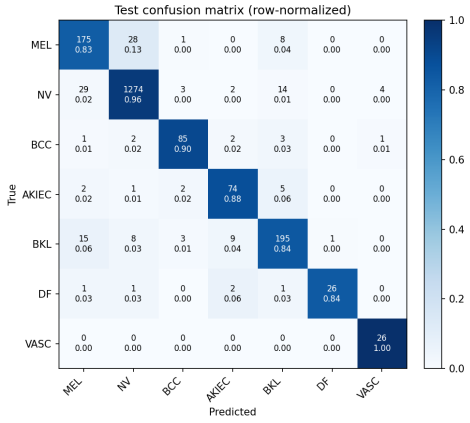


(b) TeleScopeMix.

Figure 8: Last-layer CLS token attention maps upscaled back to original resolution for skin-lesion samples across all HAM10000 classes. Each triplet contains the original image, the final transformer layer's upscaled CLS-token attention over image patches, and the attention heatmap overlaid with the original image itself. Prediction class and confidence is also displayed.



(a) Grouped ScopeMix.



(b) TeleScopeMix.

Figure 9: Confusion matrices of results on HAM10000 test set.

D Full Cross Model Evaluation

Table 6 compares LoRA, a widely used PEFT method, with the proposed ScopeMix and TeleScopeMix architectures across supervised and self-supervised backbones. ScopeMix and TeleScopeMix consistently outperform rank-16 LoRA. This advantage persists even when the LoRA rank is increased to 32, despite the resulting model using more trainable parameters than either of our methods.