RST: Residual Side Tuning with Cross-Layer Correlation for Parameter Efficient Transfer Learning

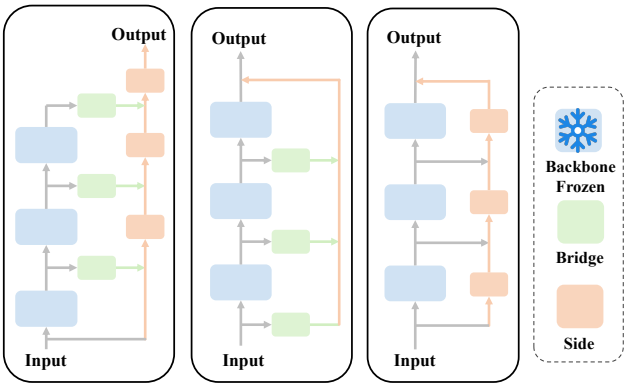
Anonymous Authors[1]

Abstract

Existing fine-tuning methods for pre-trained models, including parameter-efficient transfer learning (PETL) approaches, suffer from inefficient information extraction and substantial resource consumption. To address these issues, we present Residual Side Tuning (RST), a novel PETL framework designed to enhance information extraction efficiency while maintaining minimal additional parameters. Specifically, RST extracts aggregated features, i.e., residuals, and employs a dual-block side tuning structure: Collect Blocks extract inter-layer information into residuals while Feed Blocks strategically reintegrate them back into the backbone. This parallel processing framework effectively models cross-layer relationships and significantly improves the efficiency of hierarchical feature extraction. Furthermore, RST reinforces these relationships by leveraging an element-wise feature enhancement strategy that integrates residuals with the current layer's outputs, thereby augmenting information extraction capabilities. This enhanced extraction efficiency enables a parameter sharing strategy within the Collect Blocks, significantly reducing the number of trainable parameters through shared adaptations across multiple layers. Extensive experiments on several benchmark datasets, particularly in low-shot learning scenarios, demonstrate that RST not only outperforms existing PETL methods in accuracy but also achieves substantial reductions in both parameter and memory usage.

1. Introduction

The paradigm of large-scale pre-training followed by fine-tuning has become the cornerstone of modern machine learning, driving significant advancements across various domains such as natural language processing , computer vision , and beyond. As the scale of these pre-trained models continues to expand, fine-tuning the entire parameter set has become increasingly impractical due to prohibitive computational and memory demands. This challenge is particularly pronounced in scenarios with limited computational resources or when deploying models on edge devices.

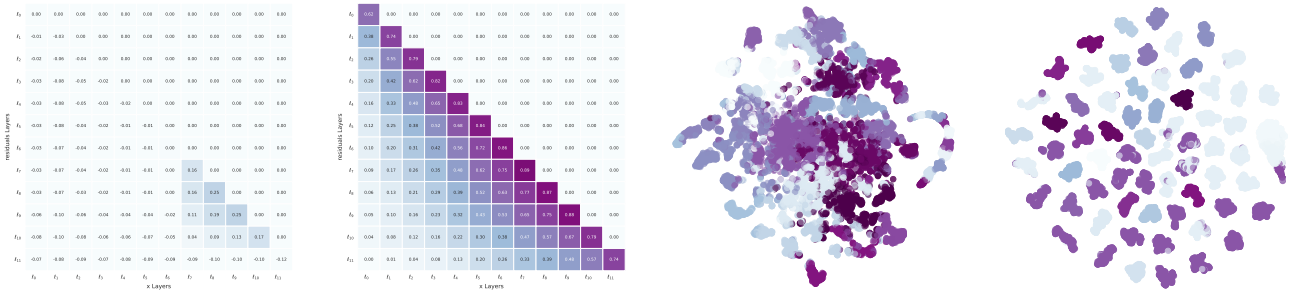


(a) LST (b) DTL (c) RST (Ours)

Figure 1. Comparative Architectures of Side Tuning Methods. In all subfigures, blue, green, orange elements respectively represent the frozen backbone network, bridge blocks that directly process backbone information, and side paths responsible for processing aggregated features. (a) LST: Combines bridge blocks and side paths. (b) DTL: Utilizes only bridge blocks. (c) RST: Employs only side paths, focusing on aggregated feature processing.

In response to these challenges, Parameter-Efficient Transfer Learning (PETL) methods have emerged as a promising solution. PETL approaches aim to adapt large pre-trained models to new tasks by updating only a small subset of parameters, thereby reducing the computational overhead and minimizing the risk of overfitting. To mitigate the memory challenges inherent in fine-tuning large pre-trained models, side tuning strategies have been proposed. These strategies decouple the trainable modules from the backbone network by introducing parallel side networks or lightweight modules, thereby effectively reducing GPU memory usage. By eliminating the need to store extensive intermediate gradients within the backbone network, side tuning not only maintains parameter efficiency but also enhances the feasibility of fine-tuning large-scale models in resource-limited settings.

Although side tuning effectively addresses memory consumption by decoupling the trainable components, existing side tuning methods often suffer from inadequate model-



(a) Feature Similarity of DTL. (b) Feature Similarity of RST. (c) Feature richness of DTL. (d) Feature richness of RST.

Figure 2. Feature Similarity and Richness in DTL and RST. (a)-(b): Layer-wise correlation analysis measuring similarity (cosine similarity) between aggregated features and corresponding backbone layer inputs across all preceding blocks, where RST (b) demonstrates stronger inter-layer correlations compared to DTL (a), indicating enhanced cross-layer relationship modeling. (c)-(d): t-SNE visualizations of aggregated features indicating feature richness, where RST (d) displays a significantly higher degree of linear separability than DTL (c), indicating enhanced richness and discriminative power of the aggregated features.

ing of inter-layer relationships and insufficient emphasis on information pertinent to the current layer. These limitations result in suboptimal information extraction capabilities, which can impede the overall performance and adaptability of fine-tuned models, especially in tasks requiring nuanced feature representations or operating under low-data regimes.

To address these fundamental drawbacks, we introduce Residual Side Tuning (RST), a novel PETL framework designed to enhance information extraction efficiency while maintaining minimal additional parameters. RST leverages a position-aware dual-block architecture where Collect/Feed Blocks are strategically aligned with specific backbone segments. By employing low-rank linear mappings on residuals, RST effectively models inter-layer relationships and focuses on task-specific feature extraction. Furthermore, we incorporate an Element-Wise Feature Enhancement strategy to dynamically integrate residual information with the current layer's outputs, thereby augmenting the model's ability to emphasize pertinent features and improve information extraction capabilities. Additionally, we implement a Parameter Sharing Strategy that enables efficient utilization of model parameters by sharing weights across Collect Blocks, which reduces the overall number of trainable parameters without compromising the richness and diversity of the extracted features. As shown in Fig. 2, compared to DTL, RST maintains stronger associations between aggregated features and the current layer while still relating to previous layers, and is capable of modeling richer feature representations. Our contributions can be summarized as follows:
1) We introduce RST, a novel PETL framework that employs low-rank linear mappings for residuals within a dual-block architecture. This approach facilitates efficient modeling of inter-layer relationships and enhances task-specific information extraction, supported by theoretical analysis of its advantageous properties.
2) We develop an Element-Wise Feature Enhancement strategy that integrates residual information with the current layer's outputs through element-wise operations, enhancing information extraction and enabling parameter reduction.

To comprehensively evaluate the effectiveness of RST, we conduct extensive experiments across multiple benchmarks, including VTAB-1K (Zhai et al., 2019), VTAB-100 built on VTAB-1K, few-shot learning, and domain generalization. Our experiments demonstrate that RST consistently outperforms existing PETL methods in accuracy, particularly in low-shot learning scenarios. Additionally, RST exhibits favorable scaling properties as model size increases. To further validate the strengths of RST, we perform ablation studies that confirm the contributions of its key components.

2. Related Work

Challenges in Fine-Tuning Large Pre-Trained Models Large pre-trained models have significantly advanced fields such as natural language processing (NLP), computer vision (CV), and vision-language (VL) tasks by leveraging vast datasets to develop comprehensive and generalizable representations. However, fine-tuning (Devlin et al., 2019; Howard & Ruder, 2018) these massive models for specific downstream tasks is computationally expensive and memory-intensive. Additionally, fully fine-tuning all parameters can lead to catastrophic forgetting, where the model loses its pre-trained knowledge when adapting to new tasks. Traditional fine-tuning approaches like linear probing (Chen et al., 2020; Yosinski et al., 2014) , which involve training only a linear classifier on frozen features, often underperform compared to full fine-tuning, highlighting the need for methods that balance parameter efficiency and training resource requirements.

Parameter-Efficient Transfer Learning (PETL) Methods Recent advances in parameter-efficient transfer learning have produced diverse adaptation strategies. Adapters (Houlsby et al., 2019; Chen et al., 2022) introduce trainable bottleneck layers between transformer blocks for task-specific feature transformation, while LoRA (Hu et al., 2021) achieves parameter reduction through low-rank decomposition of weight update matrices. BitFit (Ben Zaken et al., 2022) demonstrates surprising effectiveness by selectively updating bias terms, establishing a minimalistic tuning paradigm. In vision domains, VPT (Jia et al., 2022) pioneers learnable prompt injection at transformer inputs, whereas SSF (Lian et al., 2022) enables feature adaptation through element-wise scaling and shifting operations. Fact (Jie & Deng, 2023) enhances low-rank tuning efficiency via tensor decomposition techniques, and ConvPass (Jie & Deng, 2022) incorporates convolutional layers for localized spatial adaptation. NOAH (Zhang et al., 2022) further advances the field by automating architecture selection across multiple PETL components through neural architecture search.

Side-Tuning Methods Side Tuning (Zhang et al., 2020) enhances pre-trained backbone networks by integrating auxiliary side networks without modifying the backbone's original parameters, reducing fine-tuning memory overhead and enabling efficient knowledge transfer. Ladder Side-Tuning (LST) (Sung et al., 2022) separates trainable parameters from the backbone with a lightweight side network, effectively reducing memory consumption but potentially degrading performance on challenging tasks. Disentangled Transfer Learning (DTL) (Fu et al., 2024) builds on LST by introducing a Compact Side Network (CSN) with low-rank linear mappings, reducing memory footprint and improving performance on difficult tasks. Fig. 1 shows the main structure of them.

These existing side-tuning-based PETL methods demonstrate the potential for enhancing backbone networks efficiently. However, they often struggle to effectively extract complex and task-specific features, limiting their performance on more challenging tasks. To address these shortcomings, we propose a novel Residual Side Tuning (RST) approach, which enhances feature extraction capabilities while maintaining parameter efficiency. RST is introduced in detail in the following sections.

3. Method

We introduce Residual Side Tuning (RST), a novel parameter-efficient transfer learning framework, as shown in Fig. 3. First, we detail RST's structural design, featuring Collect and Feed Blocks alongside a residual-based LoRA approach. We then present the element-wise feature enhancement strategy that models cross-layer correlations. Finally, we describe our parameter sharing strategy within Collect Blocks, which optimizes parameter efficiency by sharing LoRA_A matrices.

3.1. Dual-Block Architecture with Low-Rank Mapping for Residuals

Dual-Block Framework RST employs a dual-block framework comprising Collect Blocks and Feed Blocks, which operate in parallel to specific sections of the frozen backbone network, thereby preserving its pre-trained knowledge. Only the Collect and Feed Blocks are learnable and updated during training. The architecture is shown in Fig. 3.

Collect Blocks are aligned with the first six blocks of the Vision Transformer (ViT) backbone, while Feed Blocks correspond to the last six blocks. These blocks extract inter-layer residuals that capture task-specific features through low-rank linear mappings, efficiently aggregating side information. During forward propagation, Collect Blocks gather residuals from the initial backbone layers, which Feed Blocks then reintegrate back into the backbone. This reintegration allows the backbone to adapt its feature representations based on the aggregated and refined task-specific information from the side path.

In backward propagation, gradients flow exclusively through the side path and the last six backbone blocks parallel to the Feed Blocks, limiting gradient backpropagation to the middle of the backbone and thus reducing memory usage. Unlike Ladder Side-Tuning (LST), which does not reintegrate information before the output layer, RST enables gradient flow through the latter backbone layers. This design achieves a balanced trade-off between performance and memory efficiency, enhancing the model's ability to adapt to new tasks while maintaining lower memory consumption and preserving the integrity of the backbone's pre-trained knowledge.

Low-Rank Linear Mapping for Residuals To further enhance RST's capability to capture complex and task-specific features, we implement low-rank linear mapping on the residuals extracted by both Collect Blocks and Feed Blocks, instead of on the inputs of backbone blocks like DTL.

Proposition 1 (Feature Aggregation Dynamics)**.** *For ViT blocks with feature dimension m and low-rank adaptation rank r, let $\boldsymbol{A}^{(i)} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{B}^{(i)} \in \mathbb{R}^{r\times m}$ denote the LoRA matrices at layer i. The aggregated features under DTL and RST architectures respectively satisfy:*

DTL:

$$s_1^{(i)}|_k = \boldsymbol{B}^{(k)\top}\boldsymbol{A}^{(k)\top}x^{(k)} \tag{1}$$

exhibiting uniform attention over historical features.

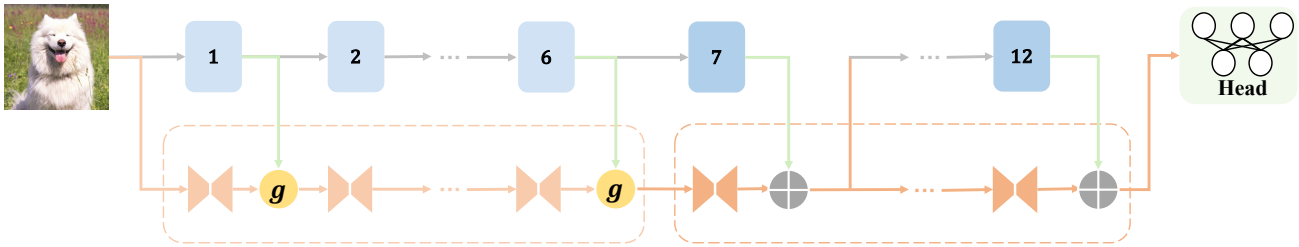


Figure 3. Architecture of RST. The RST model features a dual-block structure consisting of two distinct types of blocks. The first six blocks are Collect Blocks (shaded in light color) responsible for aggregating features from the backbone network. The subsequent six blocks are Feed Blocks (shaded in dark color) that process the aggregated features to enhance task-specific representations. An Element-Wise Feature Enhancement Gate denoted by g is integrated into the architecture, defined as $g(s, x) = \sigma(x) \odot s + x$, allowing for adaptive feature refinement and improved information extraction.

RST:

$$s_2^{(i)}|_k = \boldsymbol{B}^{(i)\top} D_k \boldsymbol{A}^{(k+1)\top} \cdot \boldsymbol{B}^{(k)\top} \boldsymbol{A}^{(k)\top} x^{(k)} \quad (2)$$

establishing layer-adaptive feature composition through matrix chain multiplication.

Proposition 2 (Gradient Sensitivity Characterization)**.** *The sensitivity of aggregated features to backbone activations reveals fundamental architectural differences:*

DTL:

$$\frac{\partial s_1^{(i)}}{\partial x^{(k)\top}}|_l = \boldsymbol{B}^{(l)\top} \boldsymbol{A}^{(l)\top} \frac{\partial x^{(l)}}{\partial x^{(k)\top}} \quad (3)$$

RST:

$$\begin{aligned} \frac{\partial s_2^{(i)}}{\partial x^{(k)\top}}|_l &= \prod_{j=0}^{i-l} \boldsymbol{B}^{(i-j)\top} \boldsymbol{A}^{(i-j)\top} \frac{\partial x^{(l)}}{\partial x^{(k)\top}} \\ &= \boldsymbol{B}^{(i)\top} D_l \boldsymbol{A}^{(l+1)\top} \boldsymbol{B}^{(l)\top} \boldsymbol{A}^{(l)\top} \frac{\partial x^{(l)}}{\partial x^{(k)\top}} \end{aligned} \quad (4)$$

where $D_l \in \mathbb{R}^{r \times r}$ is an implicit scaling matrix. The additional term $\boldsymbol{B}^{(i)\top} D_l \boldsymbol{A}^{(l+1)\top}$ in RST enables adaptive noise filtering through layer-wise decoding.

Comparing Eq. (1) and Eq. (2), $s_2^{(k)}$ has an additional item $\boldsymbol{B}^{(i)T} D_k \boldsymbol{A}^{(k+1)T}$, which implies that the information extracted from the preceding layers will be decoded by the decoder of the current layer. This reveals that applying a low-rank linear mapping to the aggregated information can make the features of the historical layers more compatible with the current features, thereby enhance the model's overall information extraction capabilities.

Comparing Eq. (3) and Eq. (4), $s_2^{(k)}$ has an additional item $\boldsymbol{B}^{(i)T} D_l \boldsymbol{A}^{(k+1)T}$, which implies that the sensitivity will be decoded by the decoder of the current layer. This reveals that our architecture can reduce the sensitivity to noise and irrelevant information introduced by the backbone network by filtering and refining residuals before reintegration. This refinement ensures that only the most pertinent and clean task-specific information is incorporated into the backbone's feature maps, thereby improving robustness and reducing susceptibility to noisy or confounding signals from the backbone, also endowing the model with the potential to enhance its generalization capabilities.

3.2. Element-Wise Feature Enhancement Strategy

To further enhance the model's ability to focus on the most relevant features within each module, we introduce an Element-Wise Feature Enhancement strategy. The primary motivation behind this strategy is to increase the model's attention to the current module's information while mitigating the challenges arising from the misalignment between the backbone's inherent features and the aggregated side path information. In complex models such as Vision Transformers (ViT), ensuring that the aggregated information aligns seamlessly with the backbone's feature representations is crucial for optimal performance. However, discrepancies between these information streams can lead to suboptimal feature integration and diminished overall model efficacy.

To address this, the Element-Wise Feature Enhancement strategy employs a two-fold approach. First, the backbone's feature map, denoted as x, undergoes a transformation using a sigmoid activation function. The choice of the sigmoid function is deliberate; it maintains the dimensional integrity of x by keeping the output within the same range as the input (i.e., between 0 and 1). This bounded transformation ensures that the scaling applied to x does not distort its original dimensional characteristics, thereby preserving the structural coherence necessary for effective feature integration.

Subsequently, the transformed feature map is combined with the aggregated residual information through an element-wise operation, specifically the Hadamard product. By multiplying corresponding elements of the transformed x and

the residuals, the model can selectively emphasize or suppress specific feature dimensions based on their relevance to the current feature. This selective enhancement ensures that the most pertinent features are accentuated, while less relevant or potentially noisy features are attenuated, thereby improving the overall quality and robustness of the feature representations.

The above process can be represented by Eq. (5) as:

$$s^{(i)} = (\boldsymbol{B}^{(i)T}\boldsymbol{A}^{(i)T}x^{(i-1)}) \odot \sigma(x^{(i+1)}) + x^{(i+1)} \quad (5)$$

This means that the aggregated features are "normalized" by the output of the current layer, which evidently heightens the focus on the current layer's information and further bridges the gap between the aggregated information and the backbone information. It is worth mentioning that despite the aforementioned advantages of this strategy, we only apply it within the collect block to prevent reducing the richness of the features.

3.3. Encoder Parameter Sharing Strategy

To enhance model efficiency, RST employs a Parameter Sharing Strategy by sharing the encoder (LoRA_A) across all Collect Blocks while keeping decoders (LoRA_B) layer-specific. This approach builds on the low-rank mapping structure and Element-Wise Feature Enhancement to reduce the number of trainable parameters and computational overhead without compromising performance.

The motivation for parameter sharing in RST arises from the distinct roles of the encoder and decoder within the low-rank mapping modules of Collect Blocks. The encoder compresses backbone information into a low-dimensional subspace, facilitating consistent residual processing across layers. In contrast, the decoder reconstructs and adapts this compressed information to enrich feature representations, capturing layer-specific nuances essential for maintaining feature richness and high performance. Sharing decoders across multiple layers would undermine their ability to refine residuals adaptively, leading to potential degradation in feature quality.

Conversely, sharing the encoder is advantageous as its role in compressing information is inherently compatible with parameter sharing. Standardizing the encoding process across different layers ensures uniform compression and seamless integration of information from diverse backbone layers, thereby enhancing overall information extraction and representation capabilities. While sharing the encoder may slightly reduce the diversity and richness of the encoded information, empirical evidence indicates that this impact is minimal and does not significantly affect performance. This is especially true since Collect Blocks primarily extract consistent and original information from the frozen backbone.

By universally sharing the encoder across all Collect Blocks, RST ensures uniform processing of residual information. Since Collect Blocks enhance feature representations through Feed Blocks and Element-Wise Feature Enhancement without directly feeding information back into the backbone, the shared encoder does not lead to significant performance losses. This architectural decision effectively balances parameter efficiency with feature quality, enabling RST to achieve superior performance with reduced resource requirements.

To implement this strategy, we universally share the encoder (LoRA_A) across all Collect Blocks within the RST framework. This sharing approach ensures that all Collect Blocks utilize the same low-dimensional compression mechanism, promoting uniformity in how residual information is processed and integrated. Importantly, since Collect Blocks do not directly feed information back into the backbone but instead serve to enhance feature representations through the Feed Blocks and Element-Wise Feature Enhancement, the impact of sharing the encoder is further mitigated. This architectural decision ensures that the consistency introduced by parameter sharing does not translate into significant performance losses, as the critical decoding and reintegration processes remain layer-specific and unaffected by the shared encoder.

We demonstrate that this strategy does not compromise the conclusions we reached in Section 3.1.

Based on Eq. (1) and Eq. (2), we can get:

$$\begin{aligned} s_1^{(i)}|_k &= \boldsymbol{B}^{(k)T}\boldsymbol{A}^{(k)T}x^{(k)} = \boldsymbol{B}^{(k)T}\boldsymbol{A}x^{(k)} \\ s_2^{(i)}|_k &= \boldsymbol{B}^{(i)T}D_k\boldsymbol{A}^{(k+1)T} \cdot \boldsymbol{B}^{(k)T}\boldsymbol{A}^{(k)T}x^{(k)} \\ &= \boldsymbol{B}^{(i)T}D'_k\boldsymbol{A}x^{(k)} \end{aligned}$$

Moreover, the sensitivity deducted in Eq. (3) and Eq. (4):

$$\begin{aligned} \frac{\partial s_1^{(i)}}{\partial x^{(k)T}}|_l &= \boldsymbol{B}^{(l)^T}\boldsymbol{A}^{(l)^T}\frac{\partial x^{(l)}}{\partial x^{(k)T}} = \boldsymbol{B}^{(l)^T}\boldsymbol{A}\frac{\partial x^{(l)}}{\partial x^{(k)T}} \\ \frac{\partial s_2^{(i)}}{\partial x^{(k)T}}|_l &= \boldsymbol{B}^{(i)^T}D_l\boldsymbol{A}^{(l+1)T}\boldsymbol{B}^{(l)^T}\boldsymbol{A}^{(l)^T}\frac{\partial x^{(l)}}{\partial x^{(k)T}} \\ &= \boldsymbol{B}^{(i)^T}D'_l\boldsymbol{A}\frac{\partial x^{(l)}}{\partial x^{(k)T}} \end{aligned}$$

where $D'_k, D'_l \in \mathbb{R}^{r\times r}$, also indicating a scaling matrix.

Therefore, the conclusions we derived in Section 3.1 remain applicable and are even presented in a clearer form.

Based on the methodologies described in this chapter, we construct two variants of the Residual Side Tuning framework: RST-L and RST-H. RST-L configures both the Collect Blocks and Feed Blocks with a rank of 2, introducing an

additional 0.029M trainable parameters. And RST-H maintains a rank of 2 for the Collect Blocks while employing a higher rank of 4 for the Feed Blocks, resulting in an additional 0.048M parameters.

4. Experiments

To comprehensively evaluate the effectiveness of the proposed Residual Side Tuning (RST) framework, we conduct extensive experiments across multiple benchmarks, including VTAB-1K (Zhai et al., 2019), VTAB-100 built on VTAB-1K, few-shot learning, and domain generalization. Besides, we conduct ablation studies to verify the properties of RST applied in Appendix B.

4.1. Experimental Settings

This section outlines our experimental settings, including the selection of pre-trained backbones, baseline methods for comparison, and implementation details.

Pre-trained Backbone For our experiments, we exclusively utilize the Vision Transformer Base/16 (ViT-B/16) (Dosovitskiy et al., 2020) model, which consists of approximately 86 million parameters and is pre-trained on the ImageNet-21K dataset (Deng et al., 2009). The ViT-B/16 backbone is chosen due to its strong scalability and adherence to the scaling laws, which facilitate efficient adaptation across various tasks. Its widespread adoption in prior works underscores its robustness and versatility, making it an ideal foundation for evaluating the performance and scalability of the RST framework.

The baseline methods are mentioned in Section 2.

Implementation Details We adhere to the implementation protocols established in prior works (Lian et al., 2022; Jie & Deng, 2023; Fu et al., 2024) to ensure consistency and reproducibility in our experiments. Specifically, we employ the AdamW optimizer (Loshchilov & Hutter, 2017) with a cosine learning rate schedule. All models are fine-tuned for 100 epochs with a batch size of 32. For the RST framework, the rank r of the low-rank linear mappings within the Collect Blocks is set to 2. We configure the number of Collect Blocks to 6, indicating that half of the later blocks' outputs are calibrated by integrating residual information. This configuration ensures a balance between adaptation capacity and parameter efficiency.

Unlike some previous methods (Zhang et al., 2022; Lian et al., 2022), we restrict our approach to standard data augmentation techniques and do not incorporate additional strategies such as mixup (Zhang et al., 2017), cutmix (Yun et al., 2019), or label smoothing (Szegedy et al., 2016). This decision streamlines the training process and highlights the intrinsic effectiveness of the RST framework. Comprehensive details of our training hyperparameters and configurations are provided in the supplementary material.

4.2. Experiments on VTAB-1K

The VTAB-1K benchmark (Zhai et al., 2019) is designed to evaluate the generalization ability of transfer learning approaches across diverse image domains. It comprises 19 distinct datasets categorized into three groups: 1) Natural images captured by standard cameras, including everyday objects and scenes, reflecting common visual recognition tasks. 2) Specialized images captured by specialist equipment, often involving medical imaging, satellite imagery, and other domains requiring specialized knowledge. 3) Structured images generated in simulated environments, including synthetic data for tasks like depth prediction and object counting. Each dataset contains exactly 1,000 training examples, making it a stringent test for Parameter-Efficient Transfer Learning (PETL) methods. The diversity of VTAB-1K spans various task-specific objectives, including classic visual recognition, object counting, and depth prediction, among others. This variety ensures that any proposed method must demonstrate robust adaptability across different visual tasks and domains.

In comparison to our previous work, the RST framework demonstrates both competitive and enhanced performance. Specifically, the RST-L variant attains an average accuracy of 76.2%, which is marginally below that of DTL. However, RST-L compensates for this slight decrease by offering significant parameter efficiency, introducing less than 0.03M trainable parameters. Furthermore, the RST-H variant surpasses DTL by achieving an average accuracy of 77.0%, thereby improving the average accuracy by 0.3%, despite a need for less than 0.01M parameters than DTL.

4.3. Experiments on VTAB-100

While VTAB-1K offers a comprehensive evaluation across various domains, we further investigate the model's information extraction capabilities using the proposed VTAB-100 subset. The motivation behind this experiment stems from the observation that low-shot datasets are more indicative of a model's ability to effectively extract and generalize information from limited data. By selecting a subset of VTAB-1K to construct VTAB-100, we aim to create a challenging benchmark that emulates an approximately 1-shot learning scenario across diverse tasks.

Specifically, VTAB-100 is meticulously constructed based on the VTAB-1K benchmark, retaining the original categorization into Natural, Specialized, and Structured images across the 19 datasets. For datasets within VTAB-1K that contain 100 classes or fewer, we ensure that each class is represented by at least one sample, maintaining a total of

Table 1. Per-task fine-tuning results on VTAB-1k benchmark. The backbone is ViT-B/16, and we ignore the linear layer when calculating the number of learnable parameters.

	#Params (M)	Natural							Specialized				Structured								Average
		CIFAR100	Caltech101	DTD	Flower102	Pets	SVHN	SUN397	Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DMLab	KITTI-Dist	dSpr-Loc	dSpr-Ori	sNORB-Azim	sNORB-Ele	
Traditional methods																					
Full	85.8	68.9	87.7	64.3	97.2	86.9	87.4	38.8	79.7	95.7	84.2	73.9	56.3	58.6	41.7	65.5	57.5	46.7	25.7	29.1	68.9
Linear	0	63.4	85.0	63.2	97.0	86.3	36.6	51.0	78.5	87.5	68.5	74.0	34.3	30.6	33.2	55.4	12.5	20.0	9.6	19.2	57.6
PETL methods																					
VPT-deep	0.60	**78.8**	90.8	65.8	98.0	88.3	78.1	49.6	81.8	96.1	83.4	68.4	68.5	60.0	46.5	72.8	73.6	47.9	32.9	37.8	72.0
BitFit	0.10	72.8	87.0	59.2	97.5	85.3	59.9	51.4	78.7	91.6	72.9	69.8	61.5	55.6	32.4	55.9	66.6	40.0	15.7	25.1	65.2
Adapter	0.16	69.2	90.1	68.0	98.8	89.9	82.8	54.3	84.0	94.9	81.9	75.5	80.9	65.3	48.6	78.3	74.8	48.5	29.9	41.6	73.9
LoRA	0.25	67.1	91.4	69.4	98.8	90.4	85.3	54.0	84.9	95.3	84.4	73.6	**82.9**	**69.2**	49.8	78.5	75.7	47.1	31.0	44.0	74.5
AdaptFormer	0.16	70.8	91.2	70.5	99.1	90.9	86.6	54.8	83.0	95.8	84.4	**76.3**	81.9	64.3	49.3	80.3	76.3	45.7	31.7	41.1	74.7
Compacter	0.15	71.9	89.0	69.7	99.1	90.7	82.7	56.1	86.0	93.5	82.4	75.3	80.2	63.4	47.4	77.2	78.1	53.5	27.3	39.8	74.2
SSF	0.21	69.0	92.6	**75.1**	**99.4**	91.8	90.2	52.9	**87.4**	95.9	**87.4**	75.5	75.9	62.3	**53.3**	80.6	77.3	54.9	29.5	37.9	75.7
NOAH	0.39	69.6	92.7	70.2	99.1	90.4	86.1	53.7	84.4	95.4	83.9	75.8	82.8	68.9	49.9	81.7	81.8	48.3	32.8	44.2	75.5
Convpass	0.33	72.3	91.2	72.2	99.2	90.9	**91.3**	54.9	84.2	96.1	85.3	75.6	82.3	67.9	51.3	80.0	85.9	53.1	**36.4**	44.4	76.6
FacT-TK	0.07	70.6	90.6	70.8	99.1	90.7	88.6	54.1	84.8	96.2	84.5	75.7	82.6	68.2	49.8	80.7	80.8	47.4	33.2	43.0	75.6
LST	2.38	59.5	91.5	69.0	99.2	89.9	79.5	54.6	86.9	95.9	85.3	74.1	81.8	61.8	52.2	81.0	71.7	49.5	33.7	45.2	74.3
DTL	0.04	69.6	**94.8**	71.3	99.3	91.3	83.3	56.2	87.1	96.2	86.1	75.0	82.8	64.2	48.8	**81.9**	**93.9**	53.9	34.2	**47.1**	76.7
Proposed methods																					
RST-L	**0.029**	72.3	94.0	73.1	**99.4**	91.7	78.8	**57.8**	86.8	96.0	86.6	73.3	82.7	64.3	49.6	80.3	85.4	54.0	32.2	45.6	76.2
RST-H	0.048	72.1	94.7	71.7	**99.4**	**91.9**	81.8	57.4	87.1	**96.5**	87.1	75.2	82.8	65.1	50.5	81.7	88.7	**57.3**	33.1	45.7	**77.0**

Table 2. Per-task fine-tuning results on VTAB-100 benchmark. The backbone is ViT-B/16, and we ignore the linear layer when calculating the number of learnable parameters.

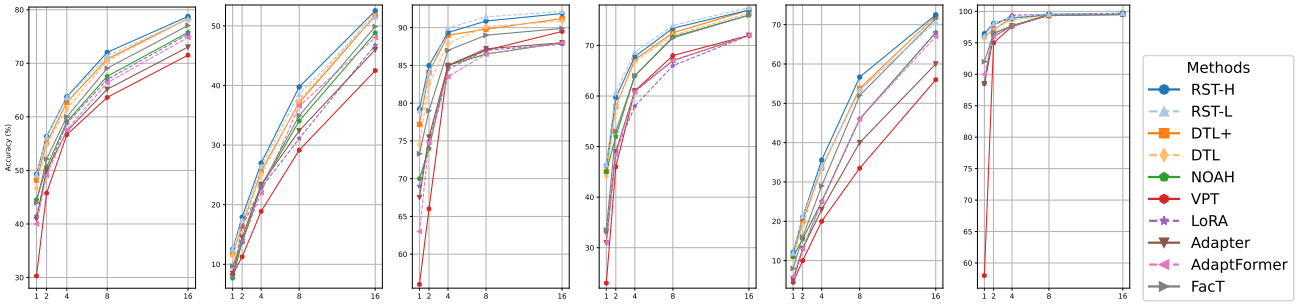
	#Params (M)	Natural							Specialized				Structured								Average
		CIFAR100	Caltech101	DTD	Flower102	Pets	SVHN	SUN397	Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DMLab	KITTI-Dist	dSpr-Loc	dSpr-Ori	sNORB-Azim	sNORB-Ele	
PETL methods																					
DTL	0.04	32.3	81.5	42.9	96.9	70.0	38.1	39.5	78.4	84.1	51.5	73.6	46.5	37.4	32.5	62.9	**24.7**	22.3	**13.4**	**22.9**	54.0
DTL+	0.05	34.5	82.1	43.2	97.0	67.7	**38.5**	41.6	77.3	85.1	50.8	**73.7**	**47.7**	**39.8**	32.8	63.6	22.9	24.5	13.4	22.9	54.3
Proposed methods																					
RST-L	**0.029**	34.4	82.5	**45.1**	97.4	**71.3**	35.2	**42.1**	**79.2**	83.5	51.3	73.6	46.8	38.2	**35.1**	**70.3**	18.0	**25.4**	11.2	22.1	54.5
RST-H	0.048	**35.8**	**82.6**	44.3	**97.5**	70.9	36.0	41.9	78.9	**85.3**	**52.4**	73.6	47.6	38.6	34.7	69.6	17.6	24.5	11.5	22.4	**54.8**

100 training samples. Conversely, for datasets with more than 100 classes, we adopt a strict 1-shot approach, selecting one representative sample per class. This methodology guarantees that VTAB-100 uniformly tests the model's ability to generalize from minimal data across a wide array of tasks and domains.

4.4. Experiments on Few-Shot Learning

To evaluate the few-shot learning capabilities of RST, we conduct experiments on five fine-grained benchmarks: Aircraft (Maji et al., 2013), Pets (Parkhi et al., 2012), Food-101 (Bossard et al., 2014), Cars (Krause et al., 2013), and Flowers102 (Nilsback & Zisserman, 2008). Following most of the previous work (Jia et al., 2022; Hu et al., 2021; Fu et al., 2024), we fine-tune the pre-trained backbone models using training sets with 1, 2, 4, 8, and 16 shots per class and report the average accuracy on the test sets over three random seeds.

As illustrated in Fig. 4, the proposed RST-L and RST-H variants consistently outperform all baseline Parameter-Efficient Transfer Learning (PETL) methods across various few-shot



(a) Average (b) Aircraft (c) OxfordPets (d) Food101 (e) Cars (f) Flowers102

Figure 4. Top-1 accuracy on fine-grained few-shot benchmark with ViT-B/16 as the backbone. Note that our approach outperforms all baseline methods.

Table 3. Top-1 accuracy on domain generalization experiments with ViT-B/16 as the backbone. Our method shows significant gains w.r.t baseline methods.

Method	Source	Target				
	ImageNet	-Sketch	-V2	-A	-R	Avg
Adapter	70.5	16.4	59.1	5.5	22.1	25.8
VPT	70.5	18.3	58.0	4.6	23.2	26.0
LoRA	70.8	20.0	59.3	6.9	23.3	27.4
NOAH	71.5	24.8	66.1	11.9	28.5	32.8
DTL	78.3	35.4	67.8	14.0	34.4	37.9
DTL+	**78.7**	35.7	67.8	14.2	34.4	38.0
RST-L	77.2	**36.8**	68.4	**15.6**	**36.4**	**39.3**
RST-H	77.0	36.6	**68.5**	15.3	36.3	39.2

scenarios. Furthermore, we observe that RST-L and RST-H exhibit average improvements of over 1% compared to the previous state-of-the-art method DTL+. These comparisons underscore the exceptional performance of the RST variants in few-shot learning tasks, validating the effectiveness of our proposed approach in extracting and generalizing information from scarce training samples.

4.5. Experiments on Domain Generalization

To assess the robustness of RST under domain shifts, we conduct domain generalization experiments following the setup of noah (Zhang et al., 2022) and dtl (Fu et al., 2024). The training set consists of samples from the original ImageNet-1K training set, with each class containing 16 training images. The model is evaluated on four distinct datasets: ImageNet-Sketch (Wang et al., 2019) composed of sketch images sharing the same label space with ImageNet-1K, ImageNet-V2 (Recht et al., 2019) collected from different sources compared with ImageNet-1K, ImageNet-A (Hendrycks et al., 2019) consisting of adversarial examples, and ImageNet-R (Hendrycks et al., 2021) containing various artistic renditions of ImageNet-1K. The paper reports the average accuracy on the train sets over three random seeds.

The results of the domain adaptation experiments are presented in Table. 3. We observe that, compared to the previous state-of-the-art methods, both RST-L and RST-H achieve impressive gains in evaluation accuracy across all target domains, with average improvements reaching up to approximately 1.3%. These comparisons highlight the exceptional robustness of the RST variants in addressing domain shift challenges and effectively demonstrate the superiority of the proposed method. Together with our previous theoretical analysis, these results validate the effectiveness and adaptability of the RST framework in diverse domain adaptation scenarios.

5. Conclusion

We introduced Residual Side Tuning (RST), a novel parameter-efficient transfer learning framework that utilizes a dual-block architecture and low-rank mappings to enhance feature extraction while minimizing parameter updates. Experimental results across multiple benchmarks demonstrated that RST outperforms existing PETL methods in accuracy and robustness, particularly in low-shot learning scenarios. Additionally, RST achieves improved memory efficiency and scalability, making it suitable for deploying large-scale pre-trained models in resource-constrained environments. These findings position RST as a significant advancement in transfer learning, with potential for further optimization and application to diverse model architectures. Extensive experiments are performed across multiple benchmarks, including VTAB-1K, VTAB-100 built on VTAB-1K, few-shot learning, and domain generalization. These experiments demonstrate that RST consistently outperforms existing PETL methods in accuracy, particularly in low-shot learning scenarios. Additionally, RST exhibits favorable scaling properties as model size increases. To further validate the strengths of RST, we perform ablation studies that confirm the contributions of its key components.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Ben Zaken, E., Goldberg, Y., and Ravfogel, S. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–9, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.1. URL https://aclanthology.org/2022.acl-short.1/.

Bossard, L., Guillaumin, M., and Van Gool, L. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 1597 – 1607, 2020.

Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., and Qiao, Y. Vision Transformer Adapter for Dense Predictions. *arXiv e-prints*, art. arXiv:2205.08534, May 2022. doi: 10.48550/arXiv.2205.08534.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019. URL https://api.semanticscholar.org/CorpusID:52967399.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv e-prints*, art. arXiv:2010.11929, October 2020. doi: 10.48550/arXiv.2010.11929.

Fu, M., Zhu, K., and Wu, J. Dtl: Disentangled transfer learning for visual recognition, 2024. URL https://arxiv.org/abs/2312.07856.

Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., and Song, D. Natural Adversarial Examples. *arXiv e-prints*, art. arXiv:1907.07174, July 2019. doi: 10.48550/arXiv.1907.07174.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., and Gilmer, J. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8320–8329, 2021. doi: 10.1109/ICCV48922.2021.00823.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-Efficient Transfer Learning for NLP. *arXiv e-prints*, art. arXiv:1902.00751, February 2019. doi: 10.48550/arXiv.1902.00751.

Howard, J. and Ruder, S. Universal language model fine-tuning for text classification. In Gurevych, I. and Miyao, Y. (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1031. URL https://aclanthology.org/P18-1031/.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv e-prints*, art. arXiv:2106.09685, June 2021. doi: 10.48550/arXiv.2106.09685.

Jia, M., Tang, L., Chen, B.-C., Cardie, C., Belongie, S., Hariharan, B., and Lim, S.-N. Visual Prompt Tuning. *arXiv e-prints*, art. arXiv:2203.12119, March 2022. doi: 10.48550/arXiv.2203.12119.

Jie, S. and Deng, Z.-H. Convolutional Bypasses Are Better Vision Transformer Adapters. *arXiv e-prints*, art. arXiv:2207.07039, July 2022. doi: 10.48550/arXiv.2207.07039.

Jie, S. and Deng, Z.-H. Fact: factor-tuning for lightweight adaptation on vision transformer. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i1.25187. URL https://doi.org/10.1609/aaai.v37i1.25187.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. In *2013 IEEE*

International Conference on Computer Vision Workshops, pp. 554–561, 2013. doi: 10.1109/ICCVW.2013.77.

Lian, D., Zhou, D., Feng, J., and Wang, X. Scaling & shifting your features: a new baseline for efficient model tuning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Loshchilov, I. and Hutter, F. Decoupled Weight Decay Regularization. *arXiv e-prints*, art. arXiv:1711.05101, November 2017. doi: 10.48550/arXiv.1711.05101.

Maji, S., Rahtu, E., Kannala, J., Blaschko, M., and Vedaldi, A. Fine-Grained Visual Classification of Aircraft. *arXiv e-prints*, art. arXiv:1306.5151, June 2013. doi: 10.48550/arXiv.1306.5151.

Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729, 2008. doi: 10.1109/ICVGIP.2008.47.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. V. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505, 2012. doi: 10.1109/CVPR.2012.6248092.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do ImageNet Classifiers Generalize to ImageNet? *arXiv e-prints*, art. arXiv:1902.10811, February 2019. doi: 10.48550/arXiv.1902.10811.

Sung, Y.-L., Cho, J., and Bansal, M. Lst: Ladder side-tuning for parameter and memory efficient transfer learning, 2022. URL https://arxiv.org/abs/2206.06522.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016. doi: 10.1109/CVPR.2016.308.

Wang, H., Ge, S., Lipton, Z., and Xing, E. P. Learning robust global representations by penalizing local predictive power. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/3eefceb8087e964f89c2d59e8a249915-Paper.pdf.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pp. 3320–3328, Cambridge, MA, USA, 2014. MIT Press.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. *arXiv e-prints*, art. arXiv:1905.04899, May 2019. doi: 10.48550/arXiv.1905.04899.

Zhai, X., Puigcerver, J., Kolesnikov, A., Ruyssen, P., Riquelme, C., Lucic, M., Djolonga, J., Susano Pinto, A., Neumann, M., Dosovitskiy, A., Beyer, L., Bachem, O., Tschannen, M., Michalski, M., Bousquet, O., Gelly, S., and Houlsby, N. A Large-scale Study of Representation Learning with the Visual Task Adaptation Benchmark. *arXiv e-prints*, art. arXiv:1910.04867, October 2019. doi: 10.48550/arXiv.1910.04867.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond Empirical Risk Minimization. *arXiv e-prints*, art. arXiv:1710.09412, October 2017. doi: 10.48550/arXiv.1710.09412.

Zhang, J. O., Sax, A., Zamir, A., Guibas, L., and Malik, J. Side-tuning: A baseline for network adaptation via additive side networks. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III*, pp. 698–714, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58579-2. doi: 10.1007/978-3-030-58580-8_41. URL https://doi.org/10.1007/978-3-030-58580-8_41.

Zhang, Y., Zhou, K., and Liu, Z. Neural Prompt Search. *arXiv e-prints*, art. arXiv:2206.04673, June 2022. doi: 10.48550/arXiv.2206.04673.

A. Proofs and Derivations

A.1. Proof of Proposition 1

For the structure of DTL that the inputs of each ViT block are processed by the low-rank linear mapping, the recursive formula for the aggregate features s is:

$$s^{(i)} = s^{(i-1)} + \boldsymbol{B}^{(i)^T}\boldsymbol{A}^{(i)^T}x^{(i)}$$

where $x^{(i)} = f^{(i)}(x^{(i-1)})$.

Obviously, we get:

$$s^{(1)} = \boldsymbol{B}^{(1)^T}\boldsymbol{A}^{(1)^T}x^{(1)}$$

Thus we can get the expression of aggregated features, as shown in Eq. (6):

$$s_1^{(i)} = \sum_{j=1}^{i} \boldsymbol{B}^{(j)^T}\boldsymbol{A}^{(j)^T}x^{(j)} \tag{6}$$

The observation indicates that DTL exhibits an equal level of attention to information across all previous layers.

For the structure of RST that the aggregated features parallel to each ViT block are processed by the low-rank linear mapping, the recursive formula for the aggregate information s is:

$$s^{(i)} = \boldsymbol{B}^{(i)^T}\boldsymbol{A}^{(i)^T}s^{(i)} + x^{(i+1)}$$

where $x^{(i+1)} = f^{(i+1)}(x^{i})$.

Obviously, we get:

$$s^{(1)} = \boldsymbol{B}^{(1)^T}\boldsymbol{A}^{(1)^T}x^{(1)} + x^{(2)}$$

Thus we can get the expression of aggregated features, as shown in Eq. (7):

$$s_2^{(i)} = \sum_{k=1}^{i}\prod_{j=0}^{i-k} \boldsymbol{B}^{(i-j)T}\boldsymbol{A}^{(i-j)T}x^{(k)} + x^{(i+1)} \tag{7}$$

For a fixed k, the aggregated feature of $s_1^{(i)}|_k$ and $s_2^{(i)}|_k$ related to x_k can be expressed in Eq. (1) and Eq. (2), respectively.

$$\begin{aligned}
s_1^{(i)}|_k &= \boldsymbol{B}^{(k)T}\boldsymbol{A}^{(k)T}x^{(k)} \\
s_2^{(i)}|_k &= \prod_{j=0}^{i-k} \boldsymbol{B}^{(i-j)T}\boldsymbol{A}^{(i-j)T}x^{(k)} \\
&= \boldsymbol{B}^{(i)T}\boldsymbol{A}^{(i)T}\cdots\boldsymbol{B}^{(k+1)T}\boldsymbol{A}^{(k+1)T}\boldsymbol{B}^{(k)T}\boldsymbol{A}^{(k)T}x^{(k)} \\
&= \boldsymbol{B}^{(i)T}D_k\boldsymbol{A}^{(k+1)T}\cdot\boldsymbol{B}^{(k)T}\boldsymbol{A}^{(k)T}x^{(k)}
\end{aligned}$$

where $D_k \in \mathbb{R}^{r\times r}$, indicating a scaling matrix.

A.2. Proof of Proposition 2

Similarly, we can get the sensitivity of the aggregated features to the backbone information of the two structures, as shown in Eq. (8) and Eq. (9):

$$\frac{\partial s_1^{(i)}}{\partial x^{(k)T}} = \sum_{l=k}^{i} \boldsymbol{B}^{(l)^T}\boldsymbol{A}^{(l)^T}\frac{\partial x^{(l)}}{\partial x^{(k)T}} \tag{8}$$

$$\frac{\partial s_2^{(i)}}{\partial x^{(k)T}} = \sum_{l=k}^{i}\Big(\prod_{j=0}^{i-l} \boldsymbol{B}^{(i-j)^T}\boldsymbol{A}^{(i-j)^T}\Big)\frac{\partial x^{(l)}}{\partial x^{(k)T}} + \frac{\partial x^{(i+1)}}{\partial x^{(k)T}} \tag{9}$$

Table 4. Ablation study of r: Per-task fine-tuning results on VTAB-100 benchmark. The intensity of the colors represents the degree of monotonic increase, with darker shades indicating a higher degree of monotonicity.

	#Params (M)	Natural							Specialized				Structured								Average
		CIFAR100	Caltech101	DTD	Flower102	Pets	SVHN	SUN397	Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DMLab	KITTI-Dist	dSpr-Loc	dSpr-Ori	sNORB-Azim	sNORB-Ele	
RST-L																					
$r=2$	0.029	34.4	82.5	45.1	97.4	71.3	35.2	42.1	79.2	83.5	51.3	73.6	46.8	38.2	35.1	70.3	18.0	25.4	11.2	22.1	54.5
$r=4$	0.058	39.0	82.0	44.9	97.5	71.4	34.1	41.6	79.2	83.4	51.9	73.6	46.4	39.6	34.5	67.2	17.7	25.3	11.6	22.3	54.6
$r=6$	0.088	41.7	82.3	45.6	97.5	73.1	35.5	41.2	81.5	83.0	53.5	73.6	46.5	39.8	34.5	71.0	19.4	26.7	11.6	23.1	55.5
$r=8$	0.117	41.5	82.4	44.5	97.4	73.3	37.2	41.0	81.8	85.9	52.7	73.6	48.6	38.8	33.9	69.1	17.4	26.6	12.0	23.7	55.6
RST-H																					
$r=2$	0.048	35.8	82.6	44.3	97.5	70.9	36.0	41.9	78.9	85.3	52.4	73.6	47.6	38.6	34.7	69.6	17.6	24.5	11.5	22.4	54.8
$r=4$	0.095	35.5	82.0	45.4	97.4	70.4	35.6	40.5	79.3	83.6	52.6	73.6	46.9	39.7	34.1	70.8	16.5	25.6	12.0	22.9	54.7
$r=6$	0.143	38.8	82.6	45.1	97.1	72.4	36.6	39.9	79.9	85.2	52.5	73.6	47.1	40.2	33.8	69.7	16.7	25.9	11.7	22.8	55.1
$r=8$	0.190	40.1	82.5	44.5	96.9	72.8	38.1	41.0	80.4	84.7	52.9	73.6	47.4	39.5	34.2	70.9	17.5	26.1	11.7	21.9	55.3

For a fixed l, the sensitivity of $s_1^{(i)}$ and $s_2^{(i)}$ to the backbone information of previous layers can be expressed as:

$$\frac{\partial s_1^{(i)}}{\partial x^{(k)T}}|_l = \boldsymbol{B}^{(l)^T} \boldsymbol{A}^{(l)^T} \frac{\partial x^{(l)}}{\partial x^{(k)T}}$$

$$\frac{\partial s_2^{(i)}}{\partial x^{(k)T}}|_l = \prod_{j=0}^{i-l} \boldsymbol{B}^{(i-j)^T} \boldsymbol{A}^{(i-j)^T} \frac{\partial x^{(l)}}{\partial x^{(k)T}} = \boldsymbol{B}^{(i)^T} D_l \boldsymbol{A}^{(l+1)T} \boldsymbol{B}^{(l)^T} \boldsymbol{A}^{(l)^T} \frac{\partial x^{(l)}}{\partial x^{(k)T}}$$

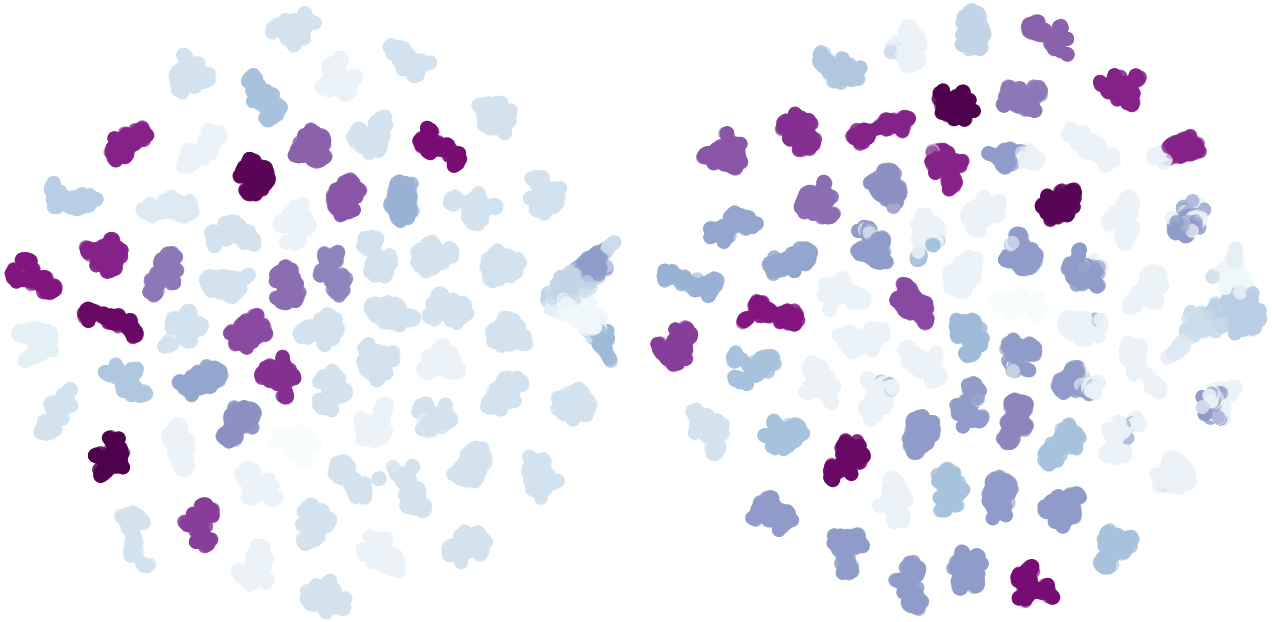
B. Ablation Study

Scaling Property To evaluate the scaling properties of RST, we investigate whether increasing the number of parameters leads to performance improvements. Specifically, we conduct experiments on the VTAB-100 benchmark to assess how different ranks r affect the performance of both RST-L and RST-H variants. For both RST-L and RST-H, we experiment with ranks $r = 2, 4, 6, 8$, of which $r = 2$ represents the original configuration. The experimental results are presented in Table 4, where results demonstrating strong scaling properties are highlighted with varying shades of yellow, where deeper shades indicate better scaling behavior.

As illustrated in Table 4, both RST-L and RST-H demonstrate consistent performance improvements as the rank r increases. Notably, RST-L shows a more pronounced scaling trend compared to RST-H, indicating its superior ability to leverage additional parameters for enhanced feature extraction. The highlighted results corroborate that RST exhibits strong scaling properties, making it adaptable and effective across different parameter configurations.

Effect of Parameter Sharing on Feature Richness To assess whether the parameter sharing strategy affects the richness of aggregated features, we conducted ablation experiments exclusively on the RST-L variant. Feature richness was evaluated using t-distributed Stochastic Neighbor Embedding (t-SNE) to visualize the separability of aggregated features. The degree of separability in the t-SNE plots serves as an indicator of feature richness, with higher separability implying more distinctive and informative feature representations.

Figure 5 presents t-SNE visualizations comparing RST-L models with and without parameter sharing. The results reveal that the introduction of parameter sharing does not degrade the separability of features. In fact, the feature distributions remain similarly distinct, indicating that feature richness is preserved despite parameter sharing. This finding aligns with our theoretical predictions, demonstrating that parameter sharing effectively reduces the number of trainable parameters without compromising the expressiveness or diversity of the extracted features. Consequently, the parameter sharing strategy employed in the RST framework is both reasonable and effective, ensuring efficient parameter utilization while maintaining high-quality feature extraction.



(a) Feature richness of RST w/o parameter sharing.

(b) Feature richness of RST w/ parameter sharing.

Figure 5. Feature Richness in RST with/without parameter sharing. T-SNE visualizations on CIFAR100 of aggregated features indicating feature richness, where parameter sharing has little effect on feature richness.