

GIST: Improving Parameter Efficient Fine Tuning via Knowledge Interaction

Jiacheng Ruan, Jingsheng Gao, Mingye Xie, Suncheng Xiang, Zefang Yu, Ting Liu, Yuzhuo Fu
 Shanghai Jiao Tong University
 jackchenruan@sjtu.edu.cn

Abstract

The Parameter-Efficient Fine-Tuning (PEFT) method, which adjusts or introduces fewer trainable parameters to calibrate pre-trained models on downstream tasks, has become a recent research interest. However, existing PEFT methods within the traditional fine-tuning framework have two main shortcomings: 1) They overlook the explicit association between trainable parameters and downstream task knowledge. 2) They neglect the interaction between the intrinsic task-agnostic knowledge of pre-trained models and the task-specific knowledge in downstream tasks. To address this gap, we propose a novel fine-tuning framework, named **GIST**, in a plug-and-play manner. Specifically, our framework first introduces a trainable token, called the *Gist token*, when applying PEFT methods on downstream tasks. This token serves as an aggregator of the task-specific knowledge learned by the PEFT methods and forms an explicit association with downstream knowledge. Furthermore, to facilitate explicit interaction between task-agnostic and task-specific knowledge, we introduce the concept of **Knowledge Interaction** via a Bidirectional Kullback-Leibler Divergence objective. As a result, PEFT methods within our framework can make the pre-trained model understand downstream tasks more comprehensively by leveraging the knowledge interaction. Extensive experiments demonstrate the universality and scalability of our framework. Notably, on the VTAB-1K benchmark, we employ the Adapter (a prevalent PEFT method) within our **GIST** framework and achieve a performance boost of 2.25%, with an increase of only 0.8K parameters (0.01% of ViT-B/16). The Code will be released.

1. Introduction

The advent of large-scale datasets and the pre-training fine-tuning paradigm has empowered pre-trained models to achieve remarkable performances [63]. By leveraging task-agnostic knowledge (**TAK**) from the pre-training phase and learning task-specific knowledge (**TSK**) during the fine-tuning process [48, 68, 71], pre-trained models, particularly

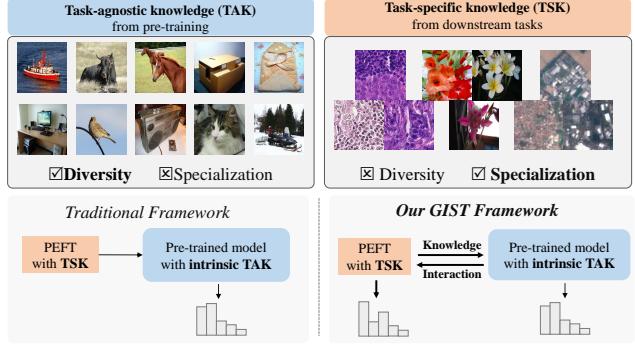


Figure 1. The issue and our motivation: Models can acquire task-agnostic knowledge (TAK) via pre-training, which is often broad and diverse but lacks specialization. Conversely, they can obtain task-specific knowledge (TSK) through fine-tuning on downstream tasks, which is typically specialized. Unlike the traditional fine-tuning framework, our goal is to establish an explicit connection between the learnable parameters and the downstream tasks, thereby comprehensively learning TSK. In addition, we introduce the concept of knowledge interaction, establishing interactions between the TAK represented by the frozen parameters and the TSK represented by the learnable parameters, thus enabling the model to better adapt to the downstream tasks.

Transformer-based models [16, 38], have exhibited exemplary performance across fields such as computer vision (CV) and natural language processing (NLP). However, the burgeoning parameters in Transformer-based models have made the Full parameter fine-Tuning (FT) method less practical for downstream tasks. The FT method demands training and storing different full parameters for each task, an approach that is not only storage-unfriendly but also prone to overfitting due to the often limited data volume in downstream tasks.

To enhance fine-tuning efficiency, the research community has shown growing interest in Parameter-Efficient Fine-Tuning (PEFT) methods [12]. PEFT methods predominantly freeze the bulk of pre-trained model parameters, adjusting or introducing a small set of trainable parameters to assimilate TSK. However, as shown in Figure 1, PEFT methods within traditional fine-tuning framework do not ex-

plicitly establish a connection between the learnable parameters and TSK, and also overlook the interaction with TAK.

To address this issue, we initiate our investigation from the perspective of TSK acquisition, based on VPT [27], a classic PEFT method. VPT achieves commendable fine-tuning performance by freezing the model’s backbone parameters and introducing learnable prompt tokens. However, in VPT, prompt tokens are not explicitly used for the final loss calculation, which may hinder the learning of TSK by trainable parameters. Consequently, we naturally utilize prompt tokens as an additional dependency for computing the loss, observing an improvement in performance. Subsequently, to demonstrate the scalability of this discovery, we attempt to employ this loss as a plug-and-play design. We apply another PEFT method (*e.g.*, Adapter) with VPT during the fine-tuning process, and find that using prompt tokens as an extra criterion also enhances performance. This discovery implies that naturally using learnable parameters as the basis for computing loss can lead to more effective downstream TSK learning.

Inspired by the above observations, we propose **GIST**, a concise and efficient framework, in a plug-and-play manner. To reduce the additional parameter burden, we directly decrease the length of prompt tokens introduced by VPT to 1, referred to as the Gist token ([GIST]), which acts as an aggregator to integrate the TSK learned by the PEFT parameters. This concept of an aggregator is derived from the pre-training stage of ViT [14], where a Class token ([CLS]) is introduced to aggregate global information for the final loss calculation.¹ Thus, by simply using the GIST token as an additional basis for loss calculation, it can help the learnable parameters of PEFT obtain more comprehensive TSK. Additionally, to address the lack of knowledge interaction, we introduce a Bidirectional Kullback-Leibler Divergence (BKLD) objective between [CLS] and [GIST], bridging the gap between the TAK and TSK. In short, under our GIST framework, PEFT methods can improve fine-tuning performance while introducing only an additional 0.8K parameters (0.01% of ViT-B/16).

We conduct extensive experiments to validate the effectiveness of our framework on 19 image classification, 5 fine-grained few-shot and 8 language understanding datasets. The results demonstrate that, when existing PEFT methods are fine-tuned within our GIST framework, they consistently deliver improved performance across a diverse range of scenarios without necessitating a substantial increase in parameter count. The contributions are as follows:

- We introduce a learnable Gist token, serving as an aggregator for task-specific knowledge acquisition. This establishes an explicit connection between learnable parame-

¹Note that [GIST] is introduced during downstream fine-tuning, is trainable, and aggregates TSK, while [CLS] is frozen during fine-tuning and can be considered to retain TAK.

ters and downstream tasks.

- We pioneer the concept of knowledge interaction during the fine-tuning phase by employing a Bidirectional Kullback-Leibler Divergence objective for explicit interaction between task-agnostic and task-specific knowledge. This objective more effectively utilizes the task-agnostic knowledge of the pre-trained models.
- We propose an innovative fine-tuning framework, dubbed as GIST. Extensive experimental results demonstrate that our framework enhances the performance of existing PEFT methods, with almost no increase in trainable parameters count.

2. Related Works

2.1. Parameter Efficient Fine-tuning

Parameter Efficient Fine-tuning (PEFT) methods enhance the performance of pre-trained models on downstream tasks in a power-saving and efficient manner. Essentially, PEFT techniques modify a select subset or introduce new trainable parameters during fine-tuning to assimilate TSK, thereby calibrating the model’s predictions on downstream tasks. Initial explorations into PEFT were predominantly within NLP tasks, with notable methodologies including Adapter [25], Prompt [35], Prefix [36], and LoRA [26], etc. Subsequently, VPT [27] migrates the Prompt technique from NLP to CV, demonstrating the potential of PEFT in visual tasks. For instance, VPT achieves impressive results by fine-tuning with only 0.1% of the total model parameters. AdaptFormer [7] introduces Adapter in parallel into the ViT’s FFN layer, achieving performance comparable to the FT method in image recognition and video understanding tasks. SSF [37] adjusts the model’s features by scaling and shifting, achieving superior results in image recognition. This success catalyzes further research into PEFT for the CV tasks, with methodologies like Conypass [28], FacT [29], and ReAdapter [42] further advancing the state-of-the-art. However, under the traditional fine-tuning framework, existing PEFT methods do not fully realize their potential because they overlook the explicit connection with TSK and the knowledge interaction with TAK. Therefore, with hardly any increase in parameters, we propose the GIST fine-tuning framework to establish explicit connections and interactions, thereby maximizing the capabilities of existing PEFT methods.

2.2. Self-Knowledge Distillation

A concept resonating with our proposed framework is self-knowledge distillation. Knowledge distillation paradigms [24] focus on enhancing the performance of student models by assimilating knowledge from a larger teacher model. In contrast, self-knowledge distillation posits the student model as its own teacher. This is achieved by deriving soft

Tag	Method	Loss	Params. (M)	Mean
(a)	VPT	\mathcal{L}_{ce}	0.05	62.41
(b)	VPT	$\mathcal{L}_{ce} + \mathcal{L}_{vpt}$	0.05	62.91
-	Adapter	\mathcal{L}_{ce}	0.13	71.46
(c)	Adapter + VPT	\mathcal{L}_{ce}	0.15	71.70
(d)	Adapter + VPT	$\mathcal{L}_{ce} + \mathcal{L}_{vpt}$	0.15	72.19

Table 1. Top-1 average accuracy on VTAB-1K. We are progressively experimenting with various structural combinations to enhance performance on downstream tasks. The hidden dimension of the Adapter is 4, and the prompt tokens’ length introduced by VPT is 20. It is better viewed in conjunction with Figure 2.

labels through specially crafted branches or distinct distributions, subsequently computing the distillation loss against its own predictions. For instance, in the BYOT approach [69], the deepest classifier is regarded as the teacher, and it imparts its knowledge to shallower networks. CS-KD [66] uses two different samples from the same category to normalize the consistency between two different views of the predicted distribution. USKD [64] utilizes the student model’s logits as soft target labels and employs the ranking of intermediate features along with Zipf’s law to generate soft non-target labels. Subsequently, USKD performs knowledge distillation using both soft labels from target and non-target classes, making it an advanced approach. In our work, we extrapolate the concept of knowledge distillation. By introducing a learnable token to derive soft labels and employing the BKLD loss as the metric between these soft labels and the model’s predictions, our fine-tuning framework aims to augment the efficacy of extant PEFT techniques with negligible parameter overhead.

3. Methods

This section delineates our GIST framework. Initially, in Section 3.1, we reassess the PEFT methods from a knowledge perspective, offering a potential application at the framework level. Subsequently, Section 3.2 delves into our GIST framework, elucidating the integration of the Gist token within the Transformer architecture, and the employment of the Bidirectional Kullback-Leibler Divergence (BKLD) loss for knowledge interaction.

3.1. Rethinking PEFT via knowledge acquisition

In this section, we first explore existing PEFT methods from the perspective of knowledge acquisition. Experiments are conducted on the VTAB-1K benchmark, with settings identical to those in Section 4.2. Initially, as shown in Figure 2(a), we start our exploration with a classic PEFT method, VPT-shallow [27]. We fix the length of the learnable prompt tokens introduced by VPT to 20, achieving an accuracy of 62.41% in fine-tuning (Table 1(a)). However, in the original VPT, only the Class token is utilized to calculate cross-entropy loss with true labels, and the learnable prompt tokens

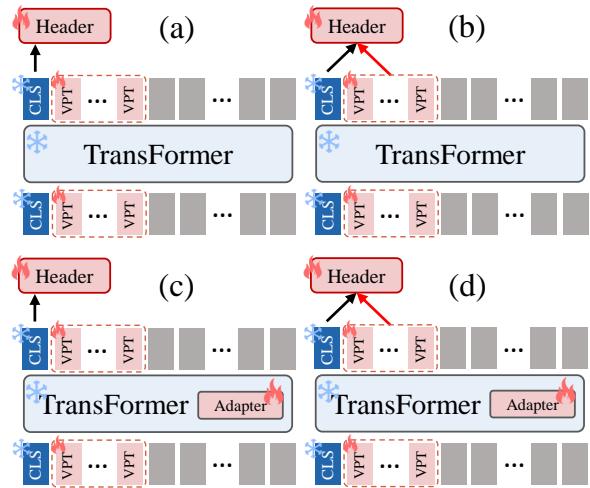


Figure 2. The different fine-tuning structures in Table 1. (a) original VPT in [27]. (b) on the top of (a), additionally using VPT prompt tokens as the basis for calculating loss. (c) combining two classic PEFT methods (VPT and Adapter) for fine-tuning. (d) on the top of (c), additionally using VPT prompt tokens as the basis for calculating loss.

tokens are not directly involved in the loss computation. We believe this form is suboptimal for fine-tuning. Therefore, as shown in Figure 2(b) and Equation 1, we naturally attempt to incorporate the prompt tokens for calculating the cross-entropy loss with the true labels. This simple modification results in a 0.5% increase in fine-tuning performance (Table 1(b)). A possible reason is that explicitly including learnable parameters in the loss calculation can lead to more comprehensive task-specific knowledge (**TSK**) acquisition.

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{ce}(S_{cls}, y) + \mathcal{L}_{vpt} \\ \mathcal{L}_{vpt} &= \mathcal{L}_{ce}(S_{vpt}, y)\end{aligned}\quad (1)$$

where \mathcal{L}_{ce} denotes the cross-entropy loss, and y represents true labels. S_{cls} and S_{vpt} represent the logits obtained from the Class token and VPT prompt tokens after passing through the linear classification head, respectively.

Subsequently, we investigate the feasibility of integrating the loss calculation approach derived from VPT with other PEFT methods. As shown in Figures 2(c, d), VPT is implemented alongside the Adapter for the fine-tuning process. Performance is evaluated and compared before and after the incorporation of the additional loss \mathcal{L}_{vpt} . The results indicate that this supplementary loss, detailed in Tables 1(c, d), further enhances the Adapter’s fine-tuning efficacy.

Therefore, we pose a question: *Can this method serve as a free lunch-style framework to enhance the fine-tuning performance of existing PEFT methods?* The answer is affirmative. In the next section, we introduce our GIST fine-tuning framework, which can enhance the performance of

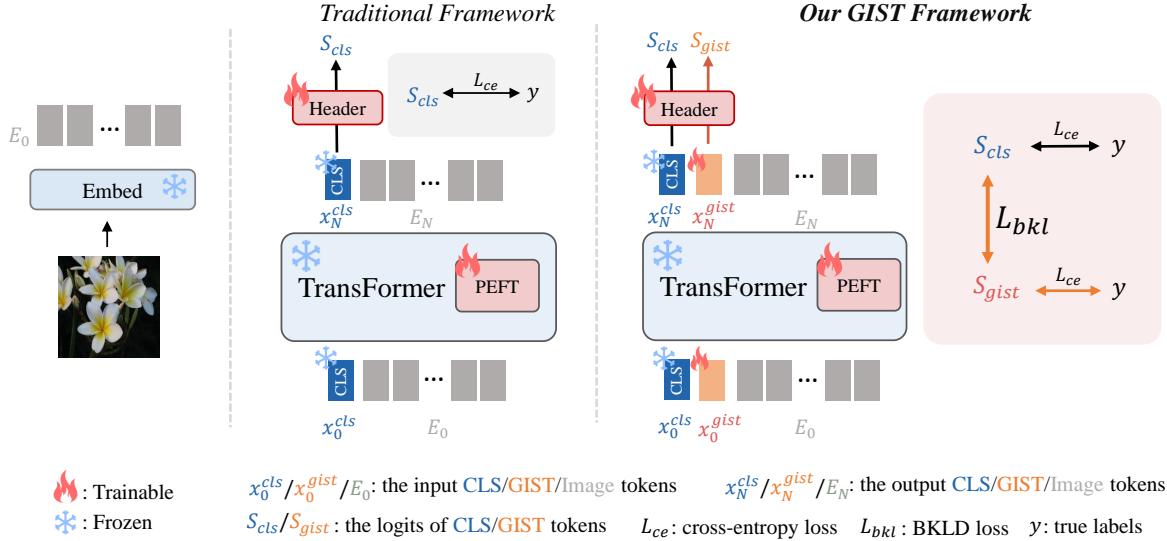


Figure 3. Overview of our GIST fine-tuning framework. Unlike the traditional fine-tuning framework, we introduce a learnable token of length one, called the Gist token, which collaboratively learns task-specific knowledge with the PEFT method on downstream tasks. Subsequently, we introduce a Bidirectional Kullback-Leibler Divergence loss to facilitate knowledge interaction.

PEFT methods in a plug-and-play manner without adding extra burden.

3.2. GIST Framework

As discussed in Section 3.1, incorporating VPT’s prompt tokens for additional loss calculation can enhance fine-tuning performance. However, this approach also introduces an increased parameter burden. Additionally, relying solely on \mathcal{L}_{vpt} as an extra loss component does not fully utilize the task-agnostic knowledge (**TAK**) from the pre-training phase. Therefore, as shown in Figure 3, our GIST framework introduces a special token called the Gist token, which is only 1 in length and designed to be an aggregator for learning TSK. Furthermore, we introduce the BKLD loss for knowledge interaction, thereby maximizing the potential of PEFT methods for more effective fine-tuning.

Gist token ([GIST]) For a TransFormer model², the input data undergoes an embedding transformation, yielding a sequence $x \in \mathbb{R}^{L \times D}$. Afterwards, the Class token $x_0^{cls} \in \mathbb{R}^{1 \times D}$ is concatenated with this sequence, augmented by a positional embedding $P \in \mathbb{R}^{(L+1) \times D}$, resulting in the input sequence $X_0 \in \mathbb{R}^{(L+1) \times D}$, as formulated in Equation 2.

$$X_0 = [x_0^{cls}; x] + P \quad (2)$$

where $[.;.]$ represents the concatenation operation. Subsequent processing of X_0 ensues through a series of Trans-

²In this section, for the sake of simplicity in expression, we omit the processing procedure of the PEFT method.

former layers, as depicted in Equation 3.

$$\begin{aligned} X'_l &= \text{MHSA}(\text{LN}(X_{l-1})) + X_{l-1} \\ X_l &= \text{FFN}(\text{LN}(X'_l)) + X'_l, l = 1, 2, \dots, N \end{aligned} \quad (3)$$

where MHSA stands for multi-head self-attention block, FFN represents the feed-forward network, and LN stands for LayerNorm [2]. After applying all Transformer layers, we can derive x_N^{cls} from X_N . The logits S_{cls} can then be obtained using the linear classification head (HEAD), as shown in Equation 4.

$$S_{cls} = \text{HEAD}(x_N^{cls}) \quad (4)$$

Ultimately, the cross entropy loss function computes the loss between S_{cls} and the true labels. Notably, during the fine-tuning phase, x_0^{cls} is frozen, preserving the model’s TAK from the pre-training phase. Different from the traditional fine-tuning framework as shown in Figure 3, our framework introduces an additional learnable token ([GIST]), denoted as $x_0^{gist} \in \mathbb{R}^{1 \times D}$, to aggregate the TSK learned by the PEFT method during fine-tuning. Thus, on the basis of Equation 2, we concatenate [GIST] to obtain our input sequence $X_0 \in \mathbb{R}^{(L+2) \times D}$, as Equation 5.

$$X_0 = [[x_0^{cls}; x] + P; x_0^{gist}] \quad (5)$$

After processing the input sequence through all Transformer layers, x_N^{cls} and x_N^{gist} are derived from X_N . We subsequently send both x_N^{cls} and x_N^{gist} through the linear classification head, resulting in S_{cls} and S_{gist} , respectively. The

loss \mathcal{L}_{cls} is computed by contrasting S_{cls} with the truth labels. Similarly, the loss \mathcal{L}_{gist} is determined by comparing S_{gist} with the true labels. These two loss terms can be expressed as follows:

$$\begin{aligned}\mathcal{L}_{cls} &= \mathcal{L}_{ce}(S_{cls}, y) \\ \mathcal{L}_{gist} &= \mathcal{L}_{ce}(S_{gist}, y)\end{aligned}\quad (6)$$

where \mathcal{L}_{ce} is the cross entropy loss, y is the true labels.

Bidirectional Kullback-Leibler Divergence (BKLD) Loss Only utilizing \mathcal{L}_{cls} and \mathcal{L}_{gist} does not facilitate explicit interaction between the TAK represented by S_{cls} and the TSK represented by S_{gist} . Therefore, we introduce the BKLD loss function, as shown in Equation 7.

$$\begin{aligned}\mathcal{L}_{bkl} &= \mathcal{L}_{fkl} + \mathcal{L}_{rkl} \\ &= \text{KL}(S_{cls} || S_{gist}; T) + \text{KL}(S_{gist} || S_{cls}; T)\end{aligned}\quad (7)$$

where \mathcal{L}_{bkl} represents our BKLD loss. \mathcal{L}_{fkl} is the forward KLD loss. \mathcal{L}_{rkl} is the reverse KLD loss. $\text{KL}(\cdot || \cdot; T)$ means computing the KL divergence between two distributions with a temperature T . The parameter T , is introduced to soften the outputs before they are processed through softmax, adjusting the sharpness of the distribution. Higher values of T produce softer probabilities [19].

For most knowledge distillation methods, the forward KLD is generally utilized as the loss function. It can be represented as $\mathcal{L}_{fkl} = \text{KL}(p || q; T)$, where p and q represent two different distributions. With p taken as the reference, \mathcal{L}_{fkl} quantifies how much the distribution q diverges from p . Conversely, the reverse KLD, denoted as $\mathcal{L}_{rkl} = \text{KL}(q || p; T)$, uses q as the reference and measures the divergence of distribution p from q . In this paper, we leverage both forward and reverse KLD as loss functions to facilitate explicit interaction between TAK and TSK. On one hand, we employ the forward KLD loss to enhance the learning of TSK, guided by TAK. On the other hand, by utilizing the reverse KLD loss, we ensure that the pre-trained model is more effectively tailored to downstream tasks, following the directives of TSK.

Overall Loss The overall loss function is derived by amalgamating \mathcal{L}_{cls} , \mathcal{L}_{gist} , and \mathcal{L}_{bkl} , as depicted in Equation 8. This loss function guides the model during the fine-tuning phase, allowing [GIST] to co-learn with other PEFT parameters and aggregate TSK, while fully leveraging TAK to ensure an explicit interaction between the two types of knowledge.

$$\mathcal{L}_{all} = \mathcal{L}_{cls} + \mu \mathcal{L}_{gist} + \lambda \mathcal{L}_{bkl} \quad (8)$$

where μ and λ is the hyperparameter that controls the trade-off among the three loss terms. It is noteworthy that the

aforementioned use of S_{gist} is limited only to the training process. For the inference process, we still solely rely on S_{cls} as the exclusive basis for prediction.

4. Experiments

4.1. Datasets and metrics

Image classification tasks We utilize the VTAB-1K benchmark [67] to validate our GIST framework for image classification tasks. Specifically, VTAB-1K includes 19 different datasets, which can be categorized into three groups: Natural, Specialized, and Structured. Each dataset consists of 1,000 samples for training, with an average of 20,000 samples for testing, making it a highly challenging benchmark. Following previous works [37], for each dataset, we report the Top-1 accuracy on the test set. For the entire benchmark, we present the arithmetic mean of the Top-1 accuracy.

Fine-grained few-shot tasks In a few-shot setting, we validate the performance of our framework in the low-data regime using Food-101 [5], OxfordPets [49], Stanford Cars [32], Oxford-Flowers102 [46], and FGVC-Aircraft [43] datasets. Similar to previous work [29, 70], we conduct validation under $\{1, 2, 4, 8, 16\}$ -shot settings and report the Top-1 accuracy on the test set.

Language understanding tasks To validate the universality of our framework, we also conduct verification for the PEFT methods in NLP. GLUE benchmark [58] is utilized to verify the effectiveness of GIST framework. Specifically, we train and test on a total of 8 tasks: MNLI, QQP, QNLI, SST-2, STS-B, MRPC, RTE, and CoLA. Following previous works [1], we use Pearson Correlation for STS-B and accuracy for other tasks as metrics.

4.2. Implementation details

For the VTAB-1K benchmark and FGVC datasets, we employ the ViT-B/16 [14] model, pre-trained on the ImageNet-21K dataset [11], as the backbone. In terms of training configurations, we follow the work of predecessors [29, 37, 42], to ensure fairness and reproducibility. Turning to the GLUE benchmark, we harness the T5-base [52] model as the backbone. Similar to the setting of the previous work [1] by configuring a batch size of 32, imposing a maximum token length of 256, setting the learning rate to 3e-4, and conducting training for 20 epochs on each task.

Regarding our GIST framework, to avoid redundancy brought about by further hyperparameter adjustment, we fix temperature T at 3, μ to 0.5, and only allow λ to be searched from $\{0.25, 0.5, 0.75\}$. Pytorch [51] and Transformers [61] are utilized to implement experiments on NVIDIA RTX 3090 GPUs, and more detailed settings are in the Appendix.

Method	Natural					Specialized			Structured					Mean	Δ	Params. (M)						
	CIFAR-100	Caltech101	DTD	Flowers102	Pets	SVHN	Sun397	PatchCamelyon	EuroSAT	Resisc45	Retinopathy	Clevr/count	Clevr/distance	DMLab	KITTI/distance	dSprites/loc	dSprites/ori	SmallNORB/azimuth	SmallNORB/elevation			
FT	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	65.57	-	85.84
LP	63.4	85.0	63.2	97.0	86.3	36.6	51.0	78.5	87.5	68.6	74.0	34.3	30.6	33.2	55.4	12.5	20.0	9.6	19.2	52.94	-	0.04
Adapter	70.2	92.6	74.6	99.4	91.2	80.4	51.4	84.1	96.3	88.0	75.6	84.2	59.6	53.2	76.3	60.7	51.9	27.8	40.2	71.46	2.25↑	0.13
Adapter*	74.5	92.3	76.9	99.5	92.3	85.7	54.6	88.2	96.5	87.9	77.4	83.6	61.2	54.0	81.2	72.3	52.1	29.3	41.0	73.71	2.25↑	0.13
VPT	60.5	90.6	70.6	99.1	89.3	50.1	50.8	82.2	93.8	82.5	74.9	50.6	58.9	41.0	68.1	39.0	32.4	22.3	29.1	62.41	1.22↑	0.05
VPT*	64.6	90.9	72.3	99.3	90.4	56.4	52.6	82.8	93.9	83.6	75.1	49.0	60.5	41.1	66.9	43.0	34.8	22.7	29.1	63.63	1.22↑	0.05
SSF	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	73.10	0.91↑	0.24
SSF*	74.2	93.1	74.4	99.5	91.8	91.2	53.7	87.5	96.1	87.3	76.2	79.1	61.6	54.5	81.2	81.7	53.9	30.9	38.2	74.01	0.91↑	0.24
FacT	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	73.23	0.32↑	0.11
FacT*	71.0	91.8	70.2	99.0	90.8	89.3	54.1	85.7	95.5	84.3	75.6	83.2	69.2	50.3	80.2	81.4	47.6	35.2	43.1	73.55	0.32↑	0.11
ReAdapter	72.4	91.6	71.0	99.2	91.4	90.7	55.1	85.3	95.9	84.6	75.9	82.3	68.0	50.4	79.9	80.4	49.2	38.6	41.0	73.83	0.43↑	0.22
ReAdapter*	73.4	92.7	71.5	99.2	91.5	91.4	55.4	84.9	96.3	85.2	75.6	82.6	70.2	51.2	80.9	82.0	47.3	36.9	42.8	74.26	0.43↑	0.22

Table 2. **The comparative results on VTAB-1K.** The symbol * indicates employing the PEFT method within our GIST framework. FT represents the full parameter fine-tuning method, and LP stands for the Linear Probing method. Params. stands for trainable parameters.

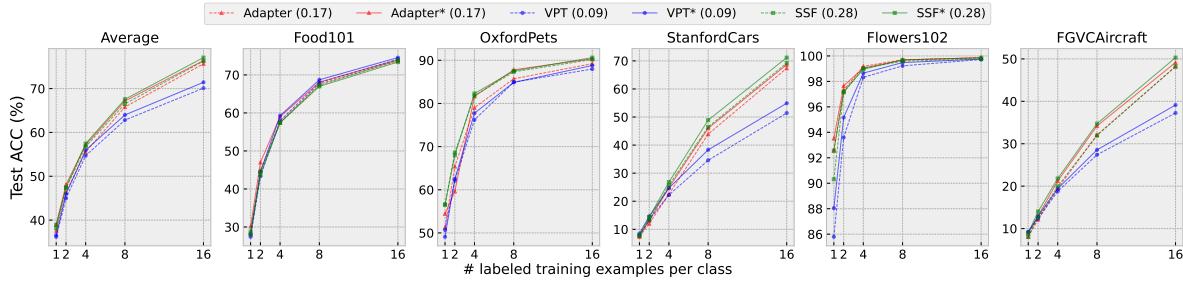


Figure 4. **Top-1 accuracy of few-shot learning on FGVC datasets.** The trainable parameters (M) is shown in parentheses.

4.3. Main results

Comparative Results on VTAB-1K We have thoroughly validated GIST framework on the benchmark for visual tasks, and the experimental results are shown in Table 2. For the three types of PEFT methods in [65], namely Adapter Tuning (Adapter and ReAdapter [42]), Prompt Tuning (VPT [27]), and Parameter Tuning (SSF [37] and FacT [29]), we apply these methods within our framework for fine-tuning on downstream tasks. This further improves the performance of the existing PEFT methods with an average increase of 1.03%, without significantly increasing the number of parameters. In the best case, our GIST can improve Adapter’s performance by 2.25%, and in the worst case, it can still enhance FacT’s performance by 0.32%. The results indicate that our framework facilitates a more comprehensive knowledge interaction and enhances the performance of PEFT methods by leveraging task-agnostic knowledge.

Comparative Results on FGVC We conduct thorough validation in a few-shot scenario. The PEFT methods

used are Adapter, VPT, and SSF, which are fine-tuned under both traditional frameworks and the GIST framework, with results shown in Figure 4. Overall, even in the low-regime few-shot scenario, fine-tuning different types of PEFT methods under the GIST framework can improve performance without significantly increasing the number of trainable parameters.

Comparative Results on GLUE We conduct validation on the GLUE benchmark for NLP tasks, and the results are shown in Table 3. When applying the FT method, 220M parameters are used to achieve a performance of 84.95%. When applying Adapter [25] within the traditional fine-tuning framework, 1.9M parameters are needed, but the performance is still 0.5% lower than that of the FT method. Notably, when utilizing Adapter within our fine-tuning framework, the performance improves by 1.45%, even exceeding the FT method by 0.95%.

4.4. Ablation studies

We conduct extensive ablation experiments on the VTAB-1K benchmark. Unless otherwise specified, we employ the

Method	MNLI	QQP	QNLI	SST-2	STS-B	MRPC	RTE	CoLA	Mean	Params. (M)
FT	86.8	91.6	93.0	94.6	89.7	90.2	71.9	61.8	84.95	220
Adapter	86.5	90.2	93.2	93.8	90.7	85.3	71.9	64.0	84.45	1.9
Adapter*	86.9	90.6	93.2	94.0	90.8	88.7	77.7	65.3	85.90	1.9

Table 3. **The comparative results on the GLUE benchmark.** We use Pearson Correlation for STS-B, and accuracy for other tasks as metrics. The symbol * indicates employing the PEFT method within our GIST framework.

λ	Mean	token len.	Mean	Params. (M)
-	71.46	1	73.71	0.13
0.25	73.31	10	73.42	0.14
0.5	73.18	50	71.96	0.16
0.75	73.44	100	71.21	0.21

Table 4. Ablation studies for different λ .

\mathcal{L}_{cls}	\mathcal{L}_{gist}	\mathcal{L}_{bkl}	Mean
✓			71.46
✓	✓		72.71
✓		✓	73.29
✓	✓	✓	73.71

Table 6. Ablations on our loss function.

Table 5. Ablation studies for different Gist token length.

Loss function	Mean
\mathcal{L}_{cls}	71.46
$\mathcal{L}_{cls} + \mathcal{L}_{gist} + \mathcal{L}_{mse}$	73.03
$\mathcal{L}_{cls} + \mathcal{L}_{gist} + \mathcal{L}_{cos}$	72.88
$\mathcal{L}_{cls} + \mathcal{L}_{gist} + \mathcal{L}_{bkl}$	73.71

Table 7. Results on different loss functions for knowledge interaction.

ViT-B/16 model, pre-trained on the ImageNet-21K dataset, as the backbone, and use Adapter as the PEFT method. Furthermore, the symbol * indicates employing the PEFT method within our GIST framework, and we display the arithmetic mean of the Top-1 accuracy. More detailed results can be found in the Appendix.

The impact of λ In our GIST framework, we only search for λ from the set $\{0.25, 0.5, 0.75\}$ to control the interaction strength between task-agnostic and task-specific knowledge. Therefore, we first conduct ablation experiments for different interaction strengths, and the results are shown in Table 4. The results indicate that regardless of the interaction strength, our fine-tuning framework can further enhance the performance of existing methods. Even in the worst case with $\lambda = 0.5$, there's still an improvement of nearly 2%.

The impact of token length As depicted in Table 5, we assess the GIST framework's performance across varying Gist token lengths. The result suggests a clear trend: as token length increases, the efficacy of our fine-tuning framework diminishes. Similar to the Class token's role during the pre-training phase, where it accumulates task-agnostic

Method	Params	Mean
S+Adapter	0.07	71.39
S+Adapter*	0.07	72.47
L+Adapter	0.30	71.81
L+Adapter*	0.30	73.89

Table 8. Results on ViT-S/16 (S) and ViT-L/16 (L).

Method	Params	Mean
FT	86.7	72.46
Linear probing	0.1	58.19
Adapter	0.21	73.19
Adapter*	0.21	74.15

Table 9. Results on Swin-B.

knowledge from the diverse training data, our Gist token's purpose is to aggregate downstream task knowledge during the fine-tuning phase. However, it is note that the length of the Class token is fixed at one. Increasing the Gist token's length may cause disproportionate knowledge interaction, leading to a decline in performance.

The impact of loss function In this study, we employ loss functions that extend beyond traditional classification loss, encompassing two components: \mathcal{L}_{gist} and \mathcal{L}_{bkl} . To evaluate the individual contributions of these components, we execute ablation studies, the results are presented in Table 6. Evidently, the efficacy of the GIST framework diminishes with a reduction in the number of loss terms. We first demonstrate the importance of establishing a direct connection between learnable parameters and task-specific knowledge during the fine-tuning process. When we introduce \mathcal{L}_{gist} into the basic loss function \mathcal{L}_{cls} , the accuracy improved by 1.25%. Alternatively, by adding \mathcal{L}_{bkl} to \mathcal{L}_{cls} , it achieves a performance gain of 1.83%, underscoring the effectiveness of knowledge interaction during downstream fine-tuning. Finally, when we introduce both types of losses simultaneously, the overall performance improves by 2.25%. This not only proves the compatibility of these two loss functions but also indicates that more comprehensive downstream knowledge acquisition can enhance the effects of knowledge interaction.

Furthermore, we assess the performance of our framework by substituting the BKLD loss with the Mean Squared Error loss \mathcal{L}_{mse} and the Cosine Similarity loss \mathcal{L}_{cos} . The comparative results are depicted in Table 7. Intriguingly, within the confines of our GIST framework, replacing our BKLD loss by common loss functions for knowledge interaction still yields a performance enhancement ranging from 1% to 2%. This attests to the scalability of our fine-tuning framework. Namely, when more advanced loss functions are proposed in subsequent research, our GIST framework can also be utilized directly to enhance the performance of the existing PEFT methods.

The impact of different networks First, to illustrate the versatility of our GIST across models of varying sizes, we substitute ViT-B/16 with ViT-S/16 and ViT-L/16, as detailed in Table 8. Next, to highlight our framework's adaptability

Method	Mean	Natural	Specialized	Structured
Adapter	71.46	79.96	86.02	56.73
Adapter+BYOT	69.70	77.86	86.24	54.29
Adapter+CS-KD	71.24	82.63	86.22	53.78
Adapter+USKD	71.40	80.14	86.78	56.06
Adapter*	73.71	82.26	87.50	59.24

Table 10. The comparative results with different self-knowledge distillation methods.



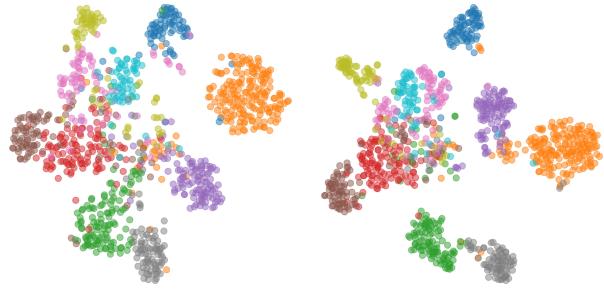
Figure 5. The attention map visualization on Sun397 dataset.

to different network structures, we conduct experiments using Swin-B [39] as the backbone, as presented in Table 9. As evident from Tables 8 and 9, regardless of whether we modify the model size or transition to an alternate backbone, our GIST consistently bolsters performance without a significant increase in parameters.

Comparisons with self-knowledge distillation methods
Our work is inspired by self-knowledge distillation (SKD) techniques. Consequently, we compare our approach with two classical methods (BYOT [69] and CS-KD [66]) as well as a state-of-the-art method (USKD [64]). The results are presented in Table 10, which reveal that even the most advanced SKD techniques can lead to a performance degradation of PEFT methods. A potential reason is that the existing SKD methods do not specifically acquire soft labels tailored for fine-tuning phase. In contrast to them, the Gist token we introduced serves as an aggregator, effectively capturing task-specific knowledge, thereby providing superior soft labels for knowledge interaction.

4.5. Visualization

We conduct attention map and t-SNE [55] visualization analysis, as depicted in Figure 5 and 6. For this, we extract the [CLS] following the final Transformer layer and preceding the linear classification head. This analysis is performed on the Sun397 [62] and SVHN [45] dataset. Notably, upon integrating GIST, the attention is more focused on the target object, and the classification clusters appear more condensed. This suggests that our framework en-



(a) Adapter, 80.4% Top-1 Acc. (b) **Adapter***, 85.7% Top-1 Acc.

Figure 6. The t-SNE visualization on SVHN dataset.

hances the ability of existing PEFT methods to assimilate more thorough task-specific knowledge via knowledge interaction. More results of visualization are in the Appendix.

4.6. Discussion

Our GIST framework possesses the following two preeminent characteristics:

- **Universality:** In the experimental section, we conduct experiments for PEFT methods on the image classification, fine-grained few-shot and language understanding tasks. The results demonstrate that our framework is versatile and can be applied to PEFT methods across various scenarios, not just confined to computer vision fields.
- **Scalability:** At the core of GIST framework lies the principle of knowledge interaction, which can be realized in multiple ways, not merely limited to the approach presented in this paper. A simple illustration, as shown in Table 7, reveals that by substituting the BKLD loss with other common losses for knowledge interaction, performance can still be augmented. This means that advanced loss functions in future research can be seamlessly integrated into our GIST framework to improve the fine-tuning performance.

5. Conclusions

In this paper, we propose GIST, an efficient and straightforward fine-tuning framework, tailored specifically for PEFT methods. This framework incorporates a learnable Gist token to explicitly establish a connection between trainable parameters and downstream tasks, thereby aiming to acquire a more comprehensive task-specific knowledge. In addition, it employs a Bidirectional Kullback-Leibler Divergence loss to enhance the interaction between task-specific and intrinsic task-agnostic knowledge of pre-trained models. Extensive experiments demonstrate that integrating existing PEFT methods with our GIST framework leads to improved performance without significantly increasing the parameter count.

GIST: Improving Parameter Efficient Fine Tuning via Knowledge Interaction

Supplementary Material

6. More detailed motivations

In this paper, our primary motivation stems from the disparities between different types of knowledge. Initially, during the pre-training phase, the datasets employed are often large-scale and diverse, yet lacking specialized information. As a result, the pre-training phase mainly endows the model with task-agnostic knowledge. In contrast, the datasets used during the fine-tuning phase tend to be small-scale and specialized, embodying primarily task-specific knowledge. At this juncture, the pre-trained model is tasked with adjusting its intrinsic task-agnostic knowledge to bridge the gap with the task-specific requirements, thereby adapting to downstream tasks. However, the volume of samples in downstream datasets is significantly smaller than that of the pre-training datasets. This means that the information content of task-specific knowledge is also less than that of task-agnostic knowledge. Consequently, the Parameter-Efficient Fine-Tuning (PEFT) approach posits that it isn't necessary to update all model parameters. Instead, making adjustments to or introducing a small number of trainable parameters can suffice for acquiring task-specific knowledge during the fine-tuning phase.

However, during the fine-tuning phase, the PEFT method under the traditional framework introduces learnable parameters that lack an explicit connection with downstream targets, leading to an inadequate acquisition of downstream knowledge. To address this, we have introduced the 'Gist token', creating a bridge between the learnable parameters and downstream objectives for more effective learning of task-specific knowledge. Further enhancing this approach, we utilize knowledge interaction through a Bidirectional Kullback-Leibler Divergence loss. This method calculates the KL divergence between Class logits, representing task-agnostic knowledge, and Gist logits, which embody task-specific knowledge. Such an interaction allows for mutual guidance between these knowledge types, significantly improving the model's adaptability to downstream tasks.

7. Datasets

In this section, we present detailed information about the VTAB-1K benchmark [67], FGVC datasets, GLUE benchmark [58] used in this paper, as shown in Tables 11, 12, and 13. Notably, following the previous work [1], we utilize 8 tasks on the GLUE benchmark, including MNLI [60], QQP³, QNLI [53], SST-2 [54], STS-B [6], MRPC [13], RTE [4, 10, 18, 21], and CoLA [59].

³data.quora.com/First-Quora-Dataset-Release-Question-Pairs

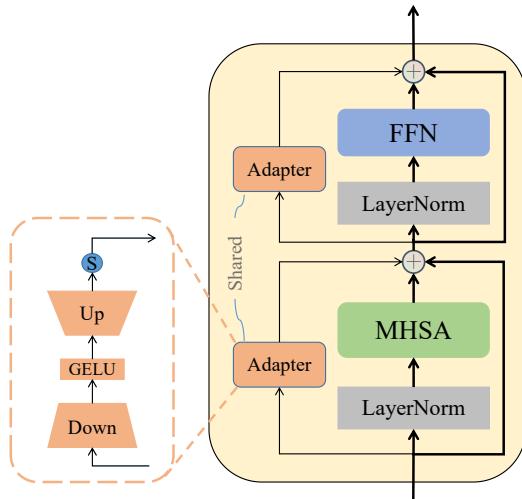


Figure 7. The Shared Adapter utilized on the VTAB-1K benchmark in this paper.

8. Implementation details on VATB-1K

Our GIST framework is compared against the traditional fine-tuning framework. Therefore, for different PEFT methods, we keep the implementation settings the same with the traditional framework.

8.1. Adapter, VPT and SSF

8.1.1 Training settings

For Adapter, VPT and SSF methods, we follow the training settings of SSF [37]. Namely, we directly resize the image to 224×224 . We employ AdamW [41] as the optimizer, set the batch size to 32, and designate 100 epochs with a provision for a 10-epoch warm-up at a warmup learning rate of 1e-7. Regarding the initial learning rate (lr), previous study [37] have established different values for various datasets, as detailed in Table 14.

8.1.2 Adapter method settings

In this study, we opt for the parallel model Adapter instead of the sequential one, offering a stronger baseline (with fewer parameters and improved performance) when compared under the traditional fine-tuning framework, as illustrated in Table 15. Specifically, as depicted in Figure 7, within the Adapter, we set the dimension of the intermediate layer to 4 and designate a scaling factor $s = 0.1$ for all experiments.

Group	Dataset	Train	Val	Test	# Class
Natural	CIFAR100 [33]			10,000	100
	Caltech101 [15]			6,084	102
	DTD [9]			1,880	47
	Oxford-Flowers102 [47]	800/1,000	200	6,149	102
	Oxford-Pets [50]			3,669	37
	SVHN [45]			26,032	10
Specialized	Sun397 [62]			21,750	397
	Patch Camelyon [57]			32,768	2
	EuroSAT [22]	800/1,000	200	5,400	10
	Resisc45 [8]			6,300	45
Structured	Retinopathy [20]			42,670	5
	Clevr/count [30]			15,000	8
	Clevr/distance [30]			15,000	6
	DMLab [3]			22,735	6
	KITTI-Dist [17]	800/1,000	200	711	4
	dSprites/location [44]			73,728	16
	dSprites/orientation [44]			73,728	16
	SmallNORB/azimuth [34]			12,150	18
	SmallNORB/elevation [34]			12,150	18

Table 11. The details of the VTAB-1K benchmark.

Dataset	Train	Val	Test	# Class
Food-101 [5]		20,200	30,300	101
Oxford-Pets [49]		736	3,669	37
Stanford Cars [32]	(1/2/4/8/16)*(#Class)	1,635	8,041	196
Oxford-Flowers102 [46]		1,633	2,463	102
FGVC-Aircraft [43]		3,333	3,333	100

Table 12. The details of the FGVC datasets.

Dataset	Task	Domain	Metric	Train	Test
MNLI	natural language inference	various	accuracy	393k	20k
QQP	paraphrase detection	social QA questions (Quora)	accuracy & F1	364k	391k
QNLI	natural language inference	Wikipedia	accuracy	105k	5.4k
SST-2	sentiment analysis	Movie Reviews	accuracy	67k	1.8k
STS-B	sentence similarity	various	Pearson & Spearman corr.	7k	1.4k
MRPC	paraphrase detection	news	accuracy & F1	3.7k	1.7k
RTE	natural language inference	News, Wikipedia	accuracy	2.5k	3k
CoLA	acceptability	various	Matthews corr.	8.5k	1k

Table 13. The details of 8 tasks we utilized on the GLUE benchmark.

Dataset	CIFAR-100	Caltech101	DTD	Flowers102	Pets	SVHN	Sun397	Patch Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr/count	Clevr/distance	DMLab	KITTI/distance	dSprites/loc	dSprites/ori	SmallNORB/azi	SmallNORB/ele
Initial lr	5e-3	1e-3	5e-3	5e-3	5e-3	1e-2	5e-3	3e-3	2e-3	5e-3	2e-3	2e-3	5e-2	5e-3	1e-2	1e-2	5e-3	2e-2	5e-3

Table 14. The detailed initial learning rate of SSF [37] method on the VTAB-1K benchmark.

Method	Natural						Specialized			Structured						Mean Params. (M)					
	CIFAR-100	Caltech101	DTD	Flowers102	Pets	SVHN	Sun397	Patch Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr/count	Clevr/distance	DMLab	KITTI/distance	dSprites/loc	dSprites/ori	SmallNORB/azi			
sequential Adapter	74.1	86.1	63.2	97.7	87.0	34.6	50.8	76.3	88.0	73.1	70.5	45.7	37.4	31.2	53.2	30.3	25.4	13.8	22.1	55.82	0.27
Shared Adapter	70.2	92.6	74.6	99.4	91.2	80.4	51.4	84.1	96.3	88.0	75.6	84.2	59.6	53.2	76.3	60.7	51.9	27.8	40.2	71.46	0.13

Table 15. The comparative results on VTAB-1K, using the backbone of ViT-B/16 pre-trained on ImageNet-21K. The results of the sequential adapter are from [37].

8.1.3 VPT method settings

In the original paper of the VPT method [27], the authors conducted an extensive search on the VTAB-1K benchmark for various tasks, experimenting with the lengths of the newly introduced prompt token in the set $\{1, 5, 10, 50, 100, 200\}$. This search process is both tedious and intricate. However, the primary aim of our study is to contrast the advantages of the GIST framework against the traditional fine-tuning framework. Consequently, for the VPT method, we consistently set the prompt token length to 20 across all experiments.

8.1.4 SSF method settings

SSF [37] eliminates the need for a tedious hyperparameter search process. Therefore, we conduct experiments under our GIST framework using the default settings from the SSF source code directly.

8.2. FacT, ReAdapter

8.2.1 Training settings

Following [29, 42], we resize the images to 224×224 and then normalize them using ImageNet’s mean and standard deviation. We employ AdamW as our optimizer with a batch size set to 64. The initial learning rate is set at 1e-3, with a weight decay of 1e-4. Training is conducted over 100 epochs, inclusive of 10 warm-up epochs, and we utilize the CosineAnnealingLR [40] for the learning rate scheduler.

8.2.2 FacT method settings

In the original paper for FacT [29], the authors undertook an extensive hyperparameter search. Specifically, for different tasks within the VTAB-1K benchmark, the authors searched for the scaling factor s across $\{0.01, 0.1, 1, 10, 100\}$. Moreover, for the rank r , they searched within the set $\{2, 4, 8, 16, 32\}$. In our study, we directly utilized the default settings from the official FacT code to conduct experiments under the GIST framework. However, it’s important to note that for the FacT method under the traditional fine-tuning framework, we directly report the results that

the authors provided in the original paper, which came after their elaborate hyperparameter search. Therefore, our GIST framework operates from a potentially disadvantaged baseline. Still, the results of Table 1 demonstrate that our GIST framework manages to surpass the traditional framework by 0.32%.

8.2.3 ReAdapter method settings

In the original ReAdapter [42] paper, the authors conducted a relatively straightforward hyperparameter search. Specifically, they only searched for the scaling factor s within the set $\{0.1, 0.5, 1, 5, 10\}$. Consequently, for ReAdapter under our GIST framework, we carried out the same hyperparameter search. The results indicate that using ReAdapter within our GIST framework outperforms its utilization within the traditional fine-tuning framework by 0.43%.

9. Implementation details on FGVC datasets

For the FGVC datasets, we use Adapter, VPT [27], and SSF [37] as representatives of three different PEFT methods, and have verified them in $\{1, 2, 4, 8, 16\}$ -shot scenarios respectively.

9.1. Training settings

For the three different PEFT methods, we consistently use AdamW [41] as the optimizer, with a batch size set to 64, a learning rate of 1e-3, weight decay of 1e-3, training for 100 epochs with 10 warmup-epochs.

9.2. Methods settings

For the three PEFT methods, the setup is the same as in Sec. 8.1.2, 8.1.3, 8.1.4.

10. Implementation details on GLUE benchmark

10.1. Training settings

For the eight tasks within the GLUE benchmark, we employ consistent training configurations [1]. Specifically, we

λ	CIFAR-100	Caltech101	DTD	Flowers102	Pets	SVHN	Sun397	Patch Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr/count	Clevr/distance	DMLab	KITTI/distance	dSprites/loc	dSprites/ori	SmallNORB/azi	SmallNORB/ele	Mean
-	70.2 92.6 74.6 99.4 91.2 80.4 51.4	84.1 96.3 88.0 75.6	84.2 59.6 53.2 76.3 60.7 51.9 27.8 40.2	71.46																
0.25	74.3 92.3 76.9 99.5 92.0 85.7 54.5	87.1 96.5 87.5 77.2	83.3 61.2 53.4 80.2 70.8 52.0 29.3 39.3	73.31																
0.5	74.5 92.2 76.3 99.5 92.1 85.0 54.4	88.1 96.3 87.6 77.4	83.1 59.8 54.0 78.7 69.5 51.9 29.1 41.0	73.18																
0.75	74.4 92.3 76.6 99.5 92.3 85.3 54.6	88.2 96.4 87.9 76.5	83.6 60.1 53.0 81.2 72.3 52.1 29.2 39.8	73.44																

Table 16. Detailed results of the ablation studies for different λ on the VTAB-1K benchmark.

Token length	CIFAR-100	Caltech101	DTD	Flowers102	Pets	SVHN	Sun397	Patch Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr/count	Clevr/distance	DMLab	KITTI/distance	dSprites/loc	dSprites/ori	SmallNORB/azi	SmallNORB/ele	Mean
1	74.5 92.3 76.9 99.5 92.3 85.7 54.6	88.2 96.5 87.9 77.4	83.6 61.2 54.0 81.2 72.3 52.1 29.3 41.0	73.71 0.13																
10	74.0 92.3 76.9 99.6 92.3 86.3 54.2	87.2 96.1 87.4 75.8	83.6 61.5 53.3 79.5 71.3 52.9 28.2 42.6	73.42 0.14																
50	73.0 92.9 74.9 99.5 91.9 87.8 52.9	85.8 96.3 87.7 75.6	83.3 54.8 52.6 79.4 65.3 53.4 19.2 40.8	71.96 0.16																
100	72.9 92.9 73.1 99.4 91.1 88.1 52.2	85.3 96.3 86.9 76.8	81.7 37.2 52.0 79.9 66.1 52.2 28.8 40.2	71.21 0.21																

Table 17. Detailed results of the ablation studies for different Gist token length on the VTAB-1K benchmark.

\mathcal{L}_{cls}	\mathcal{L}_{gist}	\mathcal{L}_{fkl}	\mathcal{L}_{rkl}	CIFAR-100	Caltech101	DTD	Flowers102	Pets	SVHN	Sun397	Patch Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr/count	Clevr/distance	DMLab	KITTI/distance	dSprites/loc	dSprites/ori	SmallNORB/azi	SmallNORB/ele	Mean
✓				70.2 92.6 74.6 99.4 91.2 80.4 51.4	84.1 96.3 88.0 75.6	84.2 59.6 53.2 76.3 60.7 51.9 27.8 40.2	71.46																
✓	✓	✓	✓	74.5 92.3 76.9 99.5 92.3 85.7 54.6	88.2 96.5 87.9 77.4	83.6 61.2 54.0 81.2 72.3 52.1 29.3 41.0	73.71																
✓		✓	✓	73.9 91.9 76.6 99.5 92.2 85.4 54.2	87.6 96.4 88.2 76.6	83.7 60.2 53.6 79.9 70.4 53.2 28.4 40.3	73.29																
✓	✓		✓	74.3 92.3 76.6 99.5 92.1 85.0 54.6	87.8 96.7 87.7 77.1	83.5 61.5 53.5 80.5 70.8 52.9 29.6 40.2	73.48																
✓	✓	✓		74.3 92.2 76.8 99.5 92.1 85.4 54.6	87.5 96.6 87.8 76.2	83.5 60.5 53.7 79.8 68.2 51.8 27.0 41.0	73.07																
✓			✓	73.4 91.9 76.5 99.5 92.1 84.8 54.0	87.4 96.7 87.5 76.8	83.7 60.5 53.2 79.2 66.4 52.7 26.5 39.7	72.76																
✓	✓			72.5 92.0 75.9 99.5 92.2 83.0 53.8	84.2 96.4 87.6 76.6	84.0 59.8 53.6 78.2 69.0 52.4 27.0 39.3	72.46																
✓	✓	✓		72.2 92.3 75.3 99.4 91.8 83.0 53.7	86.4 96.5 87.6 76.7	83.6 61.0 53.1 79.1 67.0 52.1 28.9 41.9	72.71																

Table 18. Detailed results of the ablation studies of our loss function on the VTAB-1K benchmark.

Loss function	CIFAR-100	Caltech101	DTD	Flowers102	Pets	SVHN	Sun397	Patch Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr/count	Clevr/distance	DMLab	KITTI/distance	dSprites/loc	dSprites/ori	SmallNORB/azi	SmallNORB/ele	Mean
\mathcal{L}_{cls}	70.2 92.6 74.6 99.4 91.2 80.4 51.4	84.1 96.3 88.0 75.6	84.2 59.6 53.2 76.3 60.7 51.9 27.8 40.2	71.46																
$\mathcal{L}_{cls} + \mathcal{L}_{gist} + \mathcal{L}_{mse}$	74.6 92.6 76.5 99.6 92.2 87.1 53.2	88.1 96.6 87.6 75.9	81.7 61.3 52.0 82.6 65.1 52.9 28.8 39.5	73.03																
$\mathcal{L}_{cls} + \mathcal{L}_{gist} + \mathcal{L}_{cos}$	73.0 92.0 75.8 99.5 92.2 82.9 53.6	86.2 96.4 87.5 76.5	83.6 61.3 53.7 80.1 69.1 52.6 28.0 40.7	72.88																
$\mathcal{L}_{cls} + \mathcal{L}_{gist} + \mathcal{L}_{bkl}$	74.5 92.3 76.9 99.5 92.3 85.7 54.6	88.2 96.5 87.9 77.4	83.6 61.2 54.0 81.2 72.3 52.1 29.3 41.0	73.71																

Table 19. Detailed results on different loss functions for knowledge interaction on the VTAB-1K benchmark.

set the batch size to 32, the max token length to 256, and the learning rate to 3e-4. Training was conducted over 20 epochs, incorporating 500 warm-up iterations.

10.2. Method settings

For NLP tasks on the GLUE benchmark, we carry out relatively straightforward experiments to further demonstrate

the universality of our GIST framework. We employ the default parameters from Adapter[1, 31] for our experiments. Specifically, we use GELU [23] as the activation function and set the reduction factor to 32.

Method	CIFAR-100	Caltech101	DTD	Flowers102	Pets	SVHN	Sun397	Patch Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr/count	Clevr/distance	DMLab	KITTI/distance	dSprites/loc	dSprites/ori	SmallNORB/azi	SmallNORB/ele	Mean	Params. (M)
S+Adapter	68.1	92.3	73.1	99.4	90.3	81.9	52.2	87.3	96.1	85.8	77.0	81.9	60.3	49.7	75.6	67.2	50.4	25.5	42.2	71.39	0.07
S+Adapter*	70.4	92.4	74.2	99.4	90.9	85.2	52.2	86.4	96.2	85.6	76.5	83.1	61.2	52.3	77.7	70.4	50.8	28.2	43.8	72.47	0.07
L+Adapter	72.8	91.8	74.4	99.5	92.2	84.0	54.0	87.1	96.2	89.1	75.6	78.6	57.0	52.6	77.5	68.2	53.4	25.7	34.8	71.81	0.30
L+Adapter*	77.3	91.7	77.5	99.6	92.9	88.2	58.5	87.5	96.6	89.8	76.4	81.5	55.7	54.8	81.9	73.7	54.1	27.6	38.5	73.89	0.30

Table 20. Detailed results for ViT-S/16 (S) and ViT-L/16 (L) [56] on the VTAB-1K benchmark. The symbol * indicates employing the PEFT method within our GIST framework.

Method	CIFAR-100	Caltech101	DTD	Flowers102	Pets	SVHN	Sun397	Patch Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr/count	Clevr/distance	DMLab	KITTI/distance	dSprites/loc	dSprites/ori	SmallNORB/azi	SmallNORB/ele	Mean	Params. (M)
Adapter	70.2	93.3	77.3	99.6	92.4	82.1	54.9	87.9	96.1	88.3	76.8	84.6	56.2	52.8	83.6	78.2	54.2	24.4	37.9	73.19	0.21
Adapter*	71.6	93.5	77.9	99.6	92.6	85.4	55.6	88.9	96.7	88.7	77.0	84.6	60.4	54.3	85.3	78.9	53.1	26.8	38.0	74.15	0.21

Table 21. Detailed results for Swin-B [39] on the VTAB-1K benchmark. The symbol * indicates employing the PEFT method within our GIST framework.

Method	CIFAR-100	Caltech101	DTD	Flowers102	Pets	SVHN	Sun397	Patch Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr/count	Clevr/distance	DMLab	KITTI/distance	dSprites/loc	dSprites/ori	SmallNORB/azi	SmallNORB/ele	Mean	Params. (M)
Adapter	70.2	92.6	74.6	99.4	91.2	80.4	51.4	84.1	96.3	88.0	75.6	84.2	59.6	53.2	76.3	60.7	51.9	27.8	40.2	71.46	
Adapter+BYOT	62.8	92.2	71.3	99.2	89.2	83.4	46.8	85.2	96.3	86.9	76.5	81.8	49.7	52.9	59.0	53.8	76.5	23.2	37.4	69.70	
Adapter+CS-KD	77.6	94.0	75.2	99.7	91.7	88.3	51.8	83.5	96.7	88.3	76.4	80.1	28.1	51.7	74.4	52.9	79.5	24.9	38.7	71.24	
Adapter+USKD	73.0	92.2	72.2	99.5	91.4	80.5	52.1	85.9	96.7	88.0	76.6	83.4	58.9	53.6	57.6	53.5	77.9	26.2	37.4	71.40	
Adapter*	74.5	92.3	76.9	99.5	92.3	85.7	54.6	88.2	96.5	87.9	77.4	83.6	61.2	54.0	81.2	72.3	52.1	29.3	41.0	73.71	

Table 22. Detailed results with different self-knowledge distillation methods on the VTAB-1K benchmark. The symbol * indicates employing the PEFT method within our GIST framework.

11. More experimental results

Due to space constraints, when conducting ablation experiments on the VTAB-1K benchmark, we only present the arithmetic mean of the Top-1 accuracy. Therefore, we display the complete results of all experiments, as shown in Tables 16, 19, 17, 18, 20, 21, and 22.

12. Visualization

Due to space constraints in the main paper. Thus, we present the more visualization results for the VTAB-1K benchmark, as depicted in Figure 8.

References

- [1] Akari Asai, Mohammadreza Salehi, Matthew E Peters, and Hannaneh Hajishirzi. Attempt: Parameter-efficient multi-task tuning via attentional mixtures of soft prompts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6655–6672, 2022. 5, 1, 3, 4
- [2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4
- [3] Charles Beattie, Joel Z Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, et al. Deepmind lab. *arXiv preprint arXiv:1612.03801*, 2016. 2
- [4] Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. The fifth pascal recognizing textual entailment challenge. *TAC*, 7:8, 2009. 1
- [5] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101-mining discriminative components with random forests. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part VI 13*, pages 446–461. Springer, 2014. 5, 2
- [6] Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual

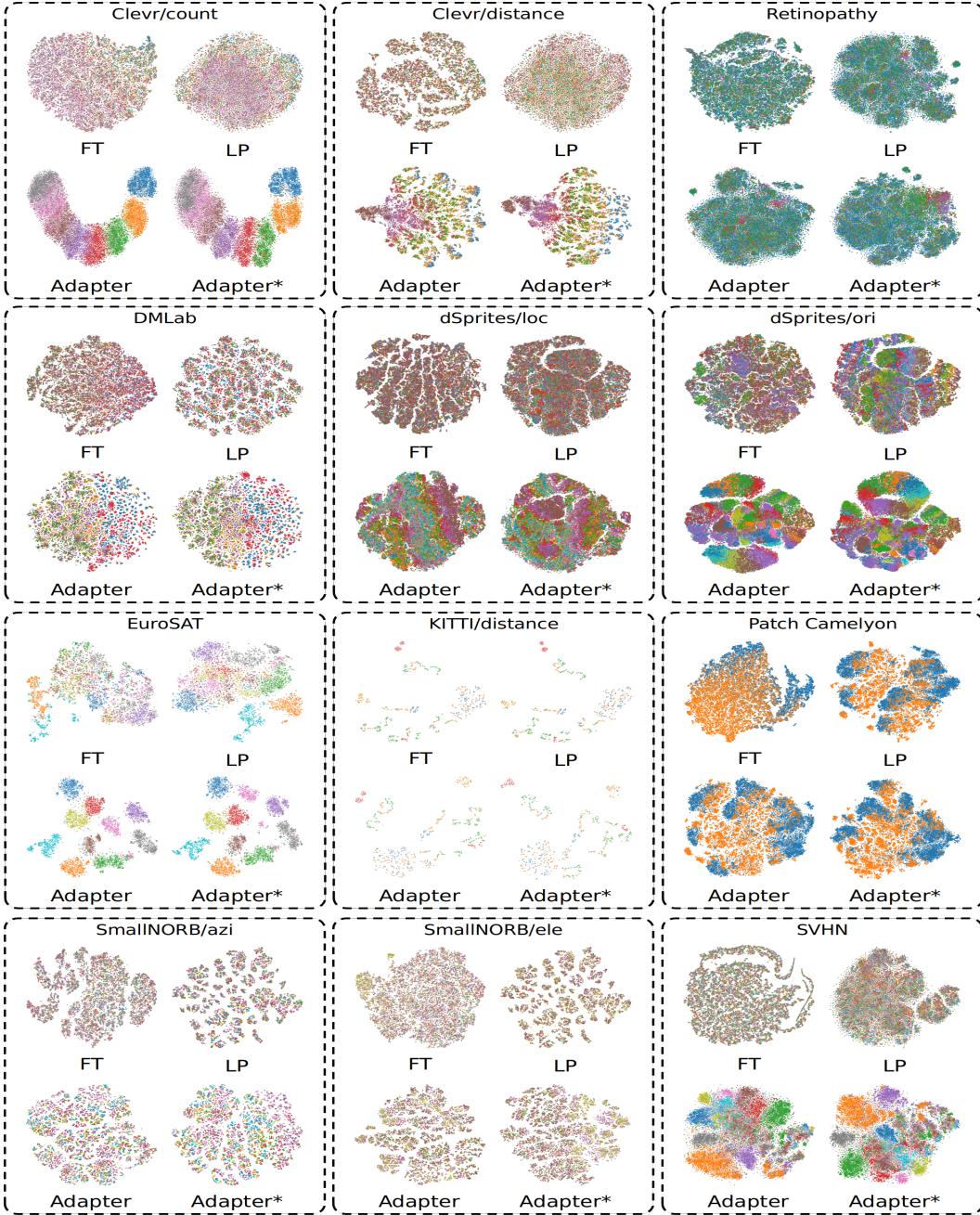


Figure 8. The results of visualization. We selected the datasets from the VTAB-1K benchmark with fewer than 20 categories for visualization. FT stands for full parameter fine-tuning, and LP stands for Linear Probing. The symbol * indicates employing the PEFT method within our GIST framework.

similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*, 2017. 1

[7] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022. 2

[8] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017. 2

[9] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014. 2

- [10] Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer, 2005. 1
- [11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5
- [12] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023. 1
- [13] Bill Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*, 2005. 1
- [14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 5
- [15] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004. 2
- [16] Zhe Gan, Linjie Li, Chunyuan Li, Lijuan Wang, Zicheng Liu, Jianfeng Gao, et al. Vision-language pre-training: Basics, recent advances, and future trends. *Foundations and Trends® in Computer Graphics and Vision*, 14(3–4):163–352, 2022. 1
- [17] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2
- [18] Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9, 2007. 1
- [19] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021. 5
- [20] Ben Graham. Kaggle diabetic retinopathy detection competition report. *University of Warwick*, 22, 2015. 2
- [21] R Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 785–794, 2006. 1
- [22] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. 2
- [23] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 4
- [24] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2
- [25] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. 2, 6
- [26] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021. 2
- [27] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer, 2022. 2, 3, 6
- [28] Shibo Jie and Zhi-Hong Deng. Convolutional bypasses are better vision transformer adapters. *arXiv preprint arXiv:2207.07039*, 2022. 2
- [29] Shibo Jie and Zhi-Hong Deng. Fact: Factor-tuning for lightweight adaptation on vision transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1060–1068, 2023. 2, 5, 6, 3
- [30] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017. 2
- [31] Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34:1022–1035, 2021. 4
- [32] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 5, 2
- [33] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2
- [34] Yann LeCun, Fu Jie Huang, and Leon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, pages II–104. IEEE, 2004. 2
- [35] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. 2
- [36] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Pa-*

- pers), pages 4582–4597, Online, 2021. Association for Computational Linguistics. 2
- [37] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems*, 35:109–123, 2022. 2, 5, 6, 1, 3
- [38] Yang Liu, Yao Zhang, Yixin Wang, Feng Hou, Jin Yuan, Jiang Tian, Yang Zhang, Zhongchao Shi, Jianping Fan, and Zhiqiang He. A survey of visual transformers. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. 1
- [39] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 8, 5
- [40] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 3
- [41] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1, 3
- [42] Gen Luo, Minglang Huang, Yiyi Zhou, Xiaoshuai Sun, Guannan Jiang, Zhiyu Wang, and Rongrong Ji. Towards efficient visual adaption via structural re-parameterization. *arXiv preprint arXiv:2302.08106*, 2023. 2, 5, 6, 3
- [43] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 5, 2
- [44] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset, 2017. 2
- [45] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 8, 2
- [46] M-E Nilsback and Andrew Zisserman. A visual vocabulary for flower classification. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, pages 1447–1454. IEEE, 2006. 5, 2
- [47] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. 2
- [48] Shuteng Niu, Yongxin Liu, Jian Wang, and Houbing Song. A decade survey of transfer learning (2010–2020). *IEEE Transactions on Artificial Intelligence*, 1(2):151–166, 2020. 1
- [49] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 5, 2
- [50] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 2
- [51] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 5
- [52] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 5
- [53] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016. 1
- [54] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013. 1
- [55] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 8
- [56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 5
- [57] Bastiaan S Veling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16–20, 2018, Proceedings, Part II 11*, pages 210–218. Springer, 2018. 2
- [58] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018. 5, 1
- [59] Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019. 1
- [60] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017. 1
- [61] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierrick Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020. 5
- [62] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010. 8, 2
- [63] Qiang Yang, Yu Zhang, Wenyuan Dai, and Sinno Jialin Pan. *Transfer learning*. Cambridge University Press, 2020. 1
- [64] Zhendong Yang, Ailing Zeng, Zhe Li, Tianke Zhang, Chun Yuan, and Yu Li. From knowledge distillation to

- self-knowledge distillation: A unified approach with normalized loss and customized soft labels. *arXiv preprint arXiv:2303.13005*, 2023. 3, 8
- [65] Bruce XB Yu, Jianlong Chang, Haixin Wang, Lingbo Liu, Shijie Wang, Zhiyu Wang, Junfan Lin, Lingxi Xie, Haojie Li, Zhouchen Lin, et al. Visual tuning. *arXiv preprint arXiv:2305.06061*, 2023. 6
- [66] Sukmin Yun, Jongjin Park, Kimin Lee, and Jinwoo Shin. Regularizing class-wise predictions via self-knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13876–13885, 2020. 3, 8
- [67] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019. 5, 1
- [68] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *arXiv preprint arXiv:2304.00685*, 2023. 1
- [69] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3713–3722, 2019. 3, 8
- [70] Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. Neural prompt search. *arXiv preprint arXiv:2206.04673*, 2022. 5
- [71] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020. 1