Combining Pre-trained LoRA Modules Improves Few-shot Adaptation of Foundation Models to New Tasks

Nader Asadi [*] [1] **Mahdi Beitollahi** [*] [1] **Yasser Khalil** [1] **Yinchuan Li** [1] **Guojun Zhang** [1] **Xi Chen** [1]

Abstract

The efficiency of low-rank adaptation (LoRA) has facilitated the creation and sharing of hundreds of custom LoRA modules for various downstream tasks. In this paper, we explore the composability of LoRA modules, examining if combining these pre-trained modules enhances the generalization of foundation models to unseen downstream tasks. Our investigation involves evaluating two approaches: (a) *uniform composition*, involving averaging upstream LoRA modules with equal weights, and (b) *learned composition*, where we learn the weights for each upstream module and perform weighted averaging. Our experimental results on both vision and language models reveal that in few-shot settings, where only a limited number of samples are available for the downstream task, both uniform and learned composition methods result in better transfer accuracy; outperforming full fine-tuning and training a LoRA from scratch. Our research unveils the potential of composition strategies for enhancing the transferability of foundation models in low-shot settings.

1. Introduction

In recent years, foundation models have demonstrated their effectiveness across a diverse set of tasks in natural language understanding, computer vision, and other fields (Bommasani et al., 2021). The widespread adoption of these models, coupled with their zero-shot capability, has spurred a trend toward standardization in training models for new tasks. Both the training methodology, often involving transfer learning from popular foundational models, and the model architecture itself have conformed to established norms, typically following a few influential foundation models (Dosovitskiy et al., 2020; Chung et al., 2022; Radford et al., 2021; Touvron et al., 2023). This standardization has given rise to numerous publicly available fine-tuned models, all sharing the same architecture. With the availability of numerous fine-tuned models derived from the same foundation model, recent studies have focused on merging multiple fine-tuned models originating from a set of upstream tasks (Matena & Raffel; Choshen et al., 2022; Ramé et al., 2023; Davari & Belilovsky, 2023).

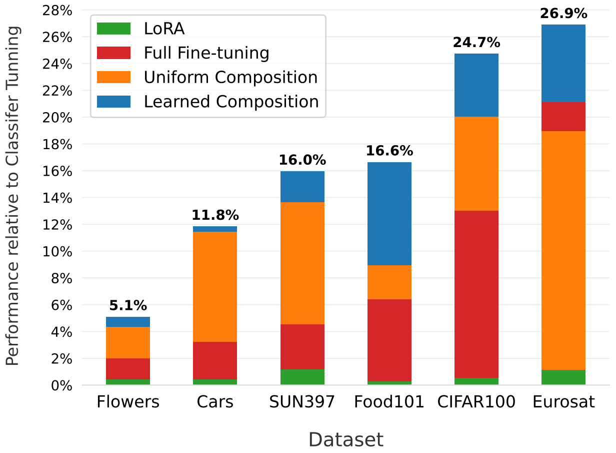


Figure 1. Performance of fine-tuning strategies relative to classifier tuning in the *one-shot* transfer learning setting. Both learned (blue) and uniform (orange) composition methods mostly outperform regular LoRA (green) and full fine-tuning (red) baselines, suggesting that the linear interpolation of pre-trained LoRA modules helps few-shot transfer to an unseen downstream task. For each dataset, we use each of the rest of the dataset as an upstream task. Refer to Section 4.1 for experiment details.

Simultaneously, due to the substantial computational cost of fine-tuning foundation models, there has been a surge in proposals for efficient *adapter* modules, enabling *parameter-efficient fine-tuning* of these models (Lester et al., 2021; Hu et al., 2021; Liu et al., 2022). Notably, low-rank adaptation (LoRA) (Hu et al., 2021) has emerged as an efficient fine-tuning technique. LoRA involves adding and training lightweight modules to a frozen pre-trained model, achieving good performance on the downstream task. By alleviating high memory demands and computational costs, LoRA has become the standard for fine-tuning Large Language Models (LLMs), diffusion models, and vision transform-

[*]Equal contribution [1]Noah's Ark Lab, Montreal, Canada. Correspondence to: Nader Asadi <nader.asadi@huawei.com>.

Published at ICML 2024 Workshop on Foundation Models in the Wild. Copyright 2024 by the author(s).

ers across various downstream tasks (Dettmers et al., 2023; Xu et al., 2023; Gandikota et al., 2023; Shah et al., 2023). LoRA's efficiency has empowered developers to create and share custom models trained on their unique data, resulting in the availability of hundreds of publicly accessible LoRA modules tailored for diverse downstream tasks.

This paper explores the possibility of leveraging pre-trained LoRA modules for efficient fine-tuning on a new task. Inspired by the literature on model merging (Wortsman et al., 2022; Choshen et al., 2022; Ilharco et al., 2022), we explore the composability of LoRA modules, examining whether knowledge from multiple upstream tasks can be combined for tackling new tasks. Specifically, we aim to answer this question: Does combining pre-trained LoRA modules enhance transfer accuracy on unseen tasks?

To answer this question, we adopt a few-shot transfer setting, where we train LoRA modules on diverse upstream tasks and subsequently evaluate various composition strategies on a downstream task with a limited number of samples. We evaluate two combining strategies: (a) *uniform composition*, where upstream LoRA modules are averaged with equal weights, and (b) *learned composition*, where we learn weights for each upstream module for weighted averaging.

Our findings in vision and language models demonstrate that the combination of pre-trained LoRA modules enhances generalization in a few-shot setting. Specifically, both uniform and learned composition methods yield superior transfer accuracy, outperforming full fine-tuning and training a LoRA from scratch as shown in Figure 1. Furthermore, our results indicate that as the number of samples in the downstream task increases, learned composition maintains performance on par with full fine-tuning and regular LoRA training while utilizing significantly fewer trainable parameters.

2. Problem Definition

Low-Rank Adaption (LoRA) Starting from a pre-trained model Θ_0, regular fine-tuning learns a different set of parameters Θ for each downstream task with $|\Theta| = |\Theta_0|$. Instead, LoRA tries to learn a set of task-specific parameters $\Delta\Theta$ with a much smaller-sized set of parameters compared to Θ_0 with $|\Delta\Theta| \ll |\Theta_0|$. Given a pre-trained weight matrix $\mathbf{W}_0 \in \mathbb{R}^{d\times c}$ of the pre-trained model Θ_0, LoRA adds a trainable low-rank decomposition matrix $\Delta\mathbf{W}$ as adapter modules to the original weight matrix $\mathbf{W}_0$:

$$\hat{\mathbf{W}} = \mathbf{W}_0 + \alpha\Delta\mathbf{W},$$

where $\Delta\mathbf{W} = \mathbf{A}\mathbf{B}^\top$ represents a low-rank matrix with rank $r \ll min(d, k)$ where $\mathbf{A} \in \mathbb{R}^{d\times r}$, $\mathbf{B}^\top \in \mathbb{R}^{r\times c}$ and α is a weighting coefficient. Then finding the value of $\Delta\Theta$ can be formulated as the standard maximum-likelihood training with cross-entropy for conditional language modeling:

$$\max_{\Delta\Theta} \sum_{(x,y)\in\mathcal{D}} \sum_{t=1}^{|y|} \log\left(P_{\Theta_0+\Delta\Theta}\left(y_t \mid x, y_{<t}\right)\right)$$

Few-shot Transfer Setup The goal is to build a single model, personalized for a novel domain utilizing the pre-trained LoRA modules from upstream domains. To evaluate the usefulness of the upstream pre-trained LoRA modules for the downstream tasks, we consider the few-shot transfer learning setting. Assuming that we have N distinct upstream tasks denoted as $\mathbb{T} = \{\mathcal{T}_1, ..., \mathcal{T}_N\}$ each having a set of trained LoRA modules. We evaluate the performance of several merging approaches of these upstream modules on a new unseen target domain $\mathcal{T}' \notin \mathbb{T}$. Each upstream domain $\mathcal{T}_n$ is defined by a set of data points $\mathcal{X}_n$, a set of ground truth labels $\mathcal{Y}_n$, and a distribution $\mathcal{D}_n$ over $\mathcal{X}_n$ and $\mathcal{Y}_n$. Similarly, the target domain $\mathcal{T}'$ is defined by a set of data points $\mathcal{X}'$, a set of ground truth labels $\mathcal{Y}'$, and a distribution $\mathcal{D}'$ over $\mathcal{X}'$ and $\mathcal{Y}'$. The few-shot learning task in the target domain consists of a very small subset of training data or *support* set from $\mathcal{D}'$:

$$\mathcal{S}^K = \{(x_k, y_k)\}_{k=1}^K \sim \mathcal{D}', \quad y_i \in \mathcal{Y}'$$

where K represents the number of adaptation samples per class, used to fine-tune the model on downstream task $\mathcal{T}'$. For the evaluation, we use all of the samples in the test or *query* set of the downstream dataset $\mathcal{D}'$.

Objective Assume for each upstream task $\mathcal{T}_n \in \{\mathcal{T}_1, ..., \mathcal{T}_N\}$, we have a set of fine-tuned LoRA modules denoted as $\Delta\mathbf{W}_n \in \{\Delta\mathbf{W}_1, ..., \Delta\mathbf{W}_N\}$. The objective is to find a combination of the upstream LoRA modules using the K samples in support set $\mathcal{S}^K$ of the unseen task $\mathcal{T}'$ to improve the performance on the query or test set. For the language modeling experiments, we follow the procedure from (Raffel et al., 2020) and formulate each task as a text-to-text problem, enabling standard maximum-likelihood training with a cross-entropy loss.

3. Combination Methods

This section highlights different strategies for combining pre-trained LoRA modules. Our objective is to effectively merge these pre-trained low-rank modules, which were originally trained on disjoint auxiliary tasks, to enhance performance in a new unseen downstream task with a limited number of samples. We consider two major recipes for merging the pre-trained adapters: *uniform* and *learned*.

3.1. Uniform Composition

We begin with a pre-trained foundation model Θ_0 that has undergone fine-tuning with LoRA for various auxiliary tasks. Denoting each weight matrix of the foundation model as $\mathbf{W}_0$, LoRA fine-tuning adds a low-rank matrix $\Delta\mathbf{W}_n$ for

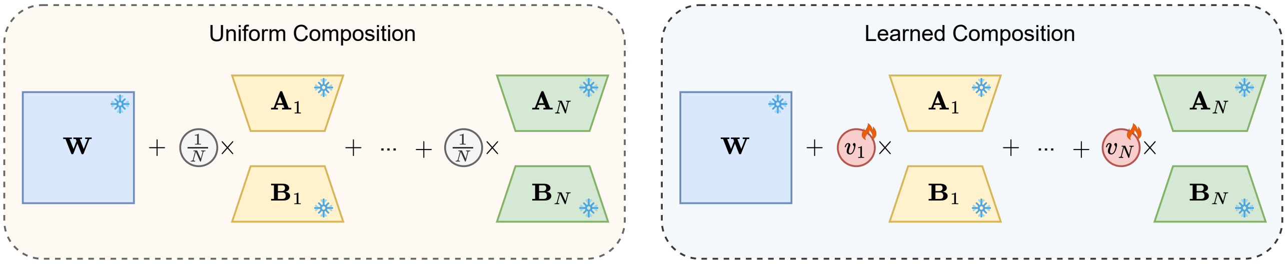


Figure 2. **Method overview.** We start with a foundational model that has undergone LoRA fine-tuning on various tasks. During the few-shot adaptation phase, we apply a *uniform* (left) or *learned* (right) weighted averaging over the pre-trained upstream LoRA weights.

the auxiliary task $\mathcal{T}_n$. The uniform composition is constructed by averaging all fine-tuned LoRA modules as:

$$\hat{\mathbf{W}} = \mathbf{W}_0 + \frac{1}{N}\sum_{n=1}^{N} \Delta\mathbf{W}_n.$$

3.2. Learned Composition

We also explore a more advanced learned composition recipe that optimizes LoRA interpolation weights by gradient-based minibatch optimization. The learned composition allows determining a specific interpolation of LoRA modules that best suits the downstream task $\mathcal{T}'$. It's worth noting that this procedure requires loading all LoRA weights into memory simultaneously. However, due to the low dimensionality of LoRA parameters, this operation is feasible, unlike learning interpolation parameters across large fine-tuned models (Wortsman et al., 2022). Specifically, we learn a weighting vector parameter $\mathbf{v} \in \mathbb{R}^N$, where $v_n \in \mathbb{R}$ denotes to the n-th element of $\mathbf{v}$ representing the weighting coefficient for the adapter of upstream task n as follows:

$$\hat{\mathbf{W}} = \mathbf{W}_0 + \sum_{n=1}^{N} \hat{v}_n \Delta\mathbf{W}_n, \quad (1)$$

where $\hat{v}_n = \frac{e^{v_n}}{\sum_{j=1}^{N} e^{v_j}}$ is the softmax operation applied on the weighting vector $\mathbf{v}$.

4. Experiments

Our experimental analysis aims to answer the following question: Does combining pre-trained LoRA modules enhance transfer accuracy on new tasks? We first explain our benchmark setup, then try to answer this question based on our observations.

Setup For all of the experiments, learning takes place in three phases. The first phase is considered as pre-training of the foundation model Θ_0. For all of the vision experiments, we considered ViT-base (Dosovitskiy et al., 2020) with a patch-size of 32×32, pre-trained on ImageNet-21K (Ridnik et al., 2021). For the NLP experiments, we use the pre-trained Flan-T5 large (Chung et al., 2022); refer to the original paper for more information. The second phase consists of fine-tuning a set of LoRA adapters, on the set of disjoint auxiliary tasks. We refer to this phase as the upstream training stage. The third and final phase consists of a few-shot adaptation to a new unseen task. For the vision task, we focus on the image classification problem, reporting *top-1 accuracy* as our evaluation metric. For the NLP task, we focus on multi-choice question answering and report *exact match*. We summarize the datasets in Table 2 and report the hyperparameter selection method in Appendix B.

4.1. Results

We conducted experiments under task shift for both vision and natural language understanding domains.

Vision Results For vision experiments, we use a subset of the 6 datasets, as used in previous work (Kornblith et al., 2019). We assessed few-shot transfer accuracy across Stanford Cars (Krause et al., 2013), Food101 (Bossard et al., 2014), Sun397 (Xiao et al., 2010), Eurosat (Helber et al., 2019), Flowers (Nilsback & Zisserman, 2008), and CIFAR100 (Krizhevsky et al., 2009). Our task shift experiments were conducted in 6 rounds, where each round considered one of these datasets as the downstream task and the others as the upstream tasks. The results can be seen in Figure 3 for Flowers and CIFAR100. We report the Food101 and EuroSat results in the Appendix. Note that the few-shot downstream adaptation is the third phase of our experimental setup, with the second phase involving the auxiliary training of upstream adapters on the remaining datasets mentioned earlier.

From Figure 3, we can observe that the uniform composition improves the model's performance in the low-shot setting. Specifically, uniform composition outperforms both LoRA and regular fine-tuning methods when we have only 1 or 2 samples per class. On the other hand, the learned composition not only performs better in low-data situations but also maintains good transfer performance across the entire spectrum. This observation answers our initial question: pre-trained LoRA modules on different tasks can indeed enhance the model's performance on new, unseen tasks.

NLP Results For the NLP experiments, we consider a subset of CrossFit benchmark (Ye et al., 2021). We fo-

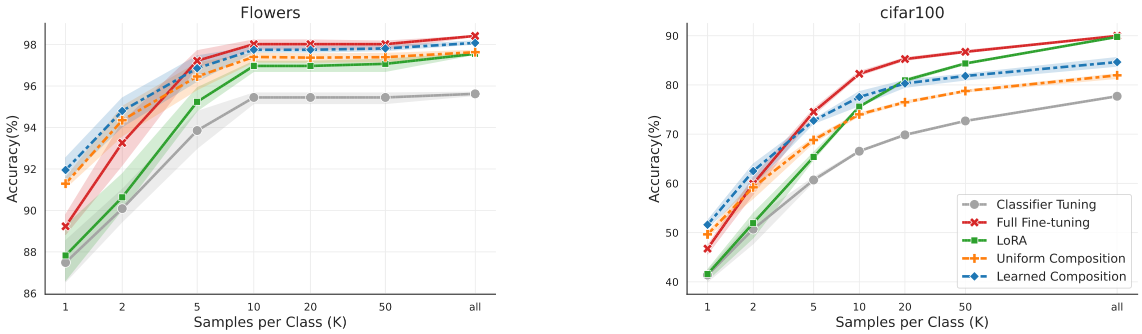


Figure 3. **Vision results** Few-shot transfer results with different number of adaptation samples. We observe that both uniform and learned composition methods significantly improve the performance with few number of adaptation samples, while maintaining comparable performance against regular LoRA fine-tuning in the full-shot scenario.

Method	ARC-Challenge					
	K=1	K=2	K=3	K=4	K=5	$\lvert\Theta\rvert$
Zero-shot	49.31 ±0.0	49.31 ±0.0	49.31 ±0.0	49.31 ±0.0	49.31 ±0.0	0
Full Fine-tuning	57.05 ±1.02	59.06 ±0.98	59.71 ±0.42	**60.94** ±0.58	**62.13** ±0.61	787M
LoRA	56.48 ±1.20	58.44 ±0.93	58.95 ±0.34	59.47 ±0.42	60.16 ±0.65	471M
Uniform Composition	59.11 ±0.0	59.11 ±0.0	59.11 ±0.0	59.11 ±0.0	59.11 ±0.0	0
Learned Composition	**59.40** ±0.87	**59.97** ±0.61	**60.39** ±0.30	60.67 ±0.39	61.12 ±0.43	1.6K

Table 1. **Task shift results (NLP).** We can observe that the uniform composition method improves the transfer performance of Flan-T5 large by 9.8% in the zero-shot setting. Also, the learned composition method beats the full and LoRA fine-tuning baselines with less than 3 and 5 adaptation samples respectively. Note that $|\Theta|$ represents the number of trainable parameters.

cus on multi-choice question answering problems using SciQ (Bhakthavatsalam et al., 2021), CommonSense (Welbl et al., 2017), QuAIL (Rogers et al., 2020), and ARC (Bhakthavatsalam et al., 2021) datasets. We evaluate the few-shot transfer accuracy of Flan-T5 large. Table 1 presents our results on the ARC dataset. We consider SciQ, CommonSense, and QuAIL as the upstream tasks. Note that the uniform composition method is also evaluated in a zero-shot setting. We can observe that the uniform merging of the upstream adapters significantly improves the zero-shot performance of the model. Additionally, the learned composition outperforms regular LoRA fine-tuning when less than five samples are available for training.

Effect of Scaling In Figure 4, we explore how the number of pre-trained upstream LoRA modules affects the transfer accuracy to a new dataset under task shift setting. Figure 4 presents our results on the Food101 dataset in a full-shot scenario and shows that, as the number of pre-trained upstream modules increases, the learned composition of these modules significantly enhances the performance, narrowing the gap compared to full-finetuning. Interestingly, our analysis reveals that the performance achieved by the learned composition of all five upstream LoRA modules surpasses that of the best individual module, suggesting that leveraging a linear combination of LoRA modules can notably improve transfer accuracy across diverse tasks.

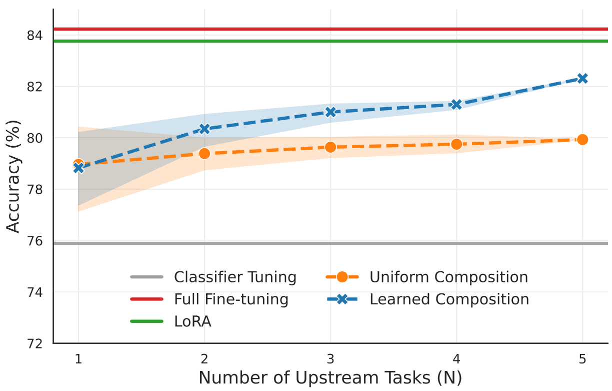


Figure 4. **Effect of scaling** The results on the Food101 dataset in full-shot (K = all) show that increasing the number of pre-trained upstream modules enhances the performance..

5. Conclusion

Our investigation into the composability of LoRA modules demonstrates their efficacy in enhancing transferability for downstream tasks. Both uniform and learned composition approaches prove advantageous, particularly in few-shot settings, surpassing traditional fine-tuning methods and even outperforming training a LoRA from scratch by up to 10.23%. This research underscores the potential of uniform composition for improving transfer accuracy in low-shot settings without introducing additional learnable parameters.

References

Bhakthavatsalam, S., Khashabi, D., Khot, T., Mishra, B. D., Richardson, K., Sabharwal, A., Schoenick, C., Tafjord, O., and Clark, P. Think you have solved direct-answer question answering? try arc-da, the direct-answer ai2 reasoning challenge. *arXiv preprint arXiv:2102.03315*, 2021.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Bossard, L., Guillaumin, M., and Van Gool, L. 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*, pp. 446–461. Springer, 2014.

Choshen, L., Venezian, E., Slonim, N., and Katz, Y. Fusing finetuned models for better pretraining. *arXiv preprint arXiv:2204.03044*, 2022.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

Davari, M. and Belilovsky, E. Model breadcrumbs: Scaling multi-task model merging with sparse masks. *arXiv preprint arXiv:2312.06795*, 2023.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Gandikota, R., Materzynska, J., Zhou, T., Torralba, A., and Bau, D. Concept sliders: Lora adaptors for precise control in diffusion models. *arXiv preprint arXiv:2311.12092*, 2023.

Helber, P., Bischke, B., Dengel, A., and Borth, D. 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.

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 preprint arXiv:2106.09685*, 2021.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Kornblith, S., Shlens, J., and Le, Q. V. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2661–2671, 2019.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., and Raffel, C. A. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35: 1950–1965, 2022.

Matena, M. and Raffel, C. Merging models with fisher-weighted averaging, 2021. *arXiv preprint arXiv:2111.09832*.

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. IEEE, 2008.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Ramé, A., Ahuja, K., Zhang, J., Cord, M., Bottou, L., and Lopez-Paz, D. Model ratatouille: Recycling diverse models for out-of-distribution generalization. In *International Conference on Machine Learning*, pp. 28656–28679. PMLR, 2023.

Ridnik, T., Ben-Baruch, E., Noy, A., and Zelnik-Manor, L. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021.

Rogers, A., Kovaleva, O., Downey, M., and Rumshisky, A. Getting closer to ai complete question answering: A set of prerequisite real tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 8722–8731, 2020.

Shah, V., Ruiz, N., Cole, F., Lu, E., Lazebnik, S., Li, Y., and Jampani, V. Ziplora: Any subject in any style by effectively merging loras. *arXiv preprint arXiv:2311.13600*, 2023.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Welbl, J., Liu, N. F., and Gardner, M. Crowdsourcing multiple choice science questions. *arXiv preprint arXiv:1707.06209*, 2017.

Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pp. 23965–23998. PMLR, 2022.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492. IEEE, 2010.

Xu, C., Guo, D., Duan, N., and McAuley, J. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. *arXiv preprint arXiv:2304.01196*, 2023.

Ye, Q., Lin, B. Y., and Ren, X. Crossfit: A few-shot learning challenge for cross-task generalization in nlp. *arXiv preprint arXiv:2104.08835*, 2021.

A. Datasets

In this section, we provide a summary of datasets used for each evaluation setting in Table 2.

Dataset	# Train	Task	Setting
CIFAR100	50,000	Vision	Task Shift
Food101	75750	Vision	Task Shift
Stanford Cars	12948	Vision	Task Shift
SUN397	108,754	Vision	Task Shift
Flowers	7,169	Vision	Task Shift
Eurosat	27,000	Vision	Task Shift
SciQ	13,679	NLP	Task Shift
CommonSense	12,102	NLP	Task Shift
QuAIL	15,000	NLP	Task Shift
ARC	7,787	NLP	Task Shift

Table 2. Summary of datasets used for each evaluation setting.

B. Hyperparameter selection

In all our experiments, we train LoRA with a rank of 16 for the query, key, and value weight matrices. This practice applies to both the ViT-base and Flan-T5 models unless stated otherwise. In our vision experiments using the ViT-base, we warm up the classifier head for 50 epochs at the beginning of training. For each method, optimal hyperparameters were selected via a grid search performed on the validation set. The selection process was done on a per-dataset basis, where we picked the configuration that maximized the accuracy averaged over different settings. We report performance with the mean and standard deviation, calculated over three random seeds.

C. Results

In this section, we provide our complete experimental results for task and domain shift settings in table format.

C.1. Task Shift Results

Method	Food101							$\lvert\Theta\rvert$
	K=1	K=2	K=5	K=10	K=20	K=50	K=all	
Classifier Tuning	30.94± 1.62	40.21± 1.59	51.77± 0.68	58.43± 0.11	63.17± 0.16	67.52± 0.25	75.89± 0.03	0
Full Fine-tuning	32.92± 1.67	43.31± 1.47	59.29± 0.57	67.25± 0.21	**72.40**± 0.32	**76.36**± 0.20	**84.23**± 0.15	86M
LoRA	31.02± 1.72	41.30± 1.80	55.71± 0.54	64.40± 0.28	70.19± 0.14	75.27± 0.19	83.76± 0.64	0.88M
Uniform Composition	33.71± 0.26	47.50± 0.37	60.06± 0.50	65.50± 0.12	69.33± 0.06	72.78± 0.23	79.93± 0.04	0
Learned Composition	**36.09**± 0.24	**49.88**± 0.50	**63.03**± 0.61	**68.37**± 0.49	72.03± 0.79	76.26± 0.57	82.31± 0.49	108

Method	Eurosat							$\lvert\Theta\rvert$
	K=1	K=2	K=5	K=10	K=20	K=50	K=all	
Classifier Tuning	43.88± 1.62	58.27± 0.83	70.98± 1.48	75.50± 0.23	78.32± 1.00	85.78± 0.28	95.72± 0.05	0
Full Fine-tuning	53.14± 3.39	68.58± 0.78	83.41± 0.13	**87.65**± 0.35	**92.56**± 0.10	**95.37**± 0.34	**98.84**± 0.03	86M
LoRA	44.37± 2.03	58.21± 0.14	72.29± 1.76	76.96± 0.44	80.05± 0.64	88.54± 0.40	98.37± 0.02	0.88M
Uniform Composition	52.20± 4.02	66.86± 3.30	80.23± 0.42	82.24± 0.41	85.32± 0.59	90.76± 0.20	96.68± 0.30	0
Learned Composition	**55.68**± 4.24	**70.53**± 3.47	**85.16**± 0.66	87.50± 0.71	91.13± 0.27	95.21± 0.12	98.05± 0.46	108

Method	Flowers							|Θ|
	K=1	K=2	K=5	K=10	K=20	K=50	K=all	
Classifier Tuning	87.49± 0.82	90.09± 0.67	93.85± 0.73	95.45± 0.22	95.45± 0.22	95.45± 0.22	95.62± 0.07	0
Full Fine-tuning	89.23± 0.41	93.26± 0.85	**97.22**± 0.38	**98.02**± 0.19	**98.02**± 0.19	**98.02**± 0.19	**98.42**± 0.02	86M
LoRA	87.83± 1.09	90.63± 0.80	95.24± 0.49	96.97± 0.21	96.97± 0.21	97.07± 0.26	97.55± 0.04	0.88M
Uniform Composition	91.29± 0.21	94.35± 0.59	96.46± 0.53	97.40± 0.19	97.37± 0.17	97.39± 0.17	97.63± 0.07	0
Learned Composition	**91.95**± 0.46	**94.80**± 0.58	96.86± 0.50	97.75± 0.11	97.75± 0.11	97.81± 0.09	98.08± 0.08	108

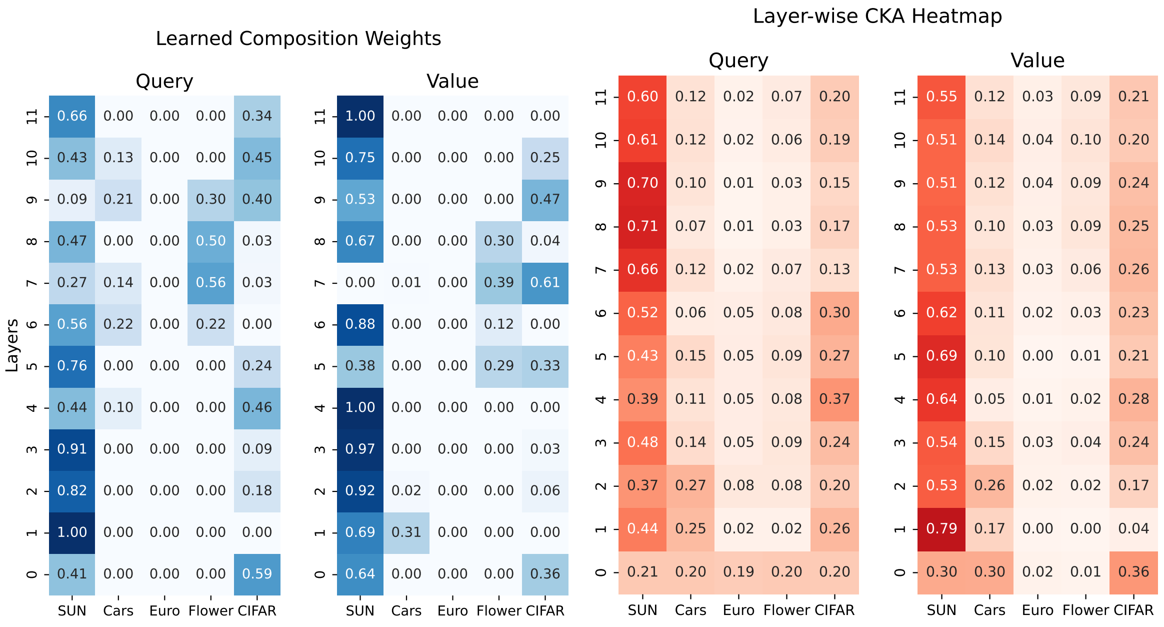
Method	CIFAR100							|Θ|
	K=1	K=2	K=5	K=10	K=20	K=50	K=all	
Classifier Tuning	41.37± 1.00	50.71± 2.17	60.69± 0.41	66.54± 0.16	69.86± 0.19	72.69± 0.09	77.70± 0.02	0
Full Fine-tuning	46.74± 0.80	59.93± 1.53	**74.53**± 0.92	**82.26**± 0.37	**85.26**± 0.25	**86.73**± 0.22	**89.97**± 0.04	86M
LoRA	41.59± 0.92	51.92± 2.17	65.37± 0.68	75.59± 0.18	80.93± 0.29	84.36± 0.18	89.76± 0.06	0.88M
Uniform Composition	49.65± 0.39	59.20± 1.61	68.81± 0.44	74.02± 0.28	76.48± 0.23	78.77± 0.18	81.95± 0.46	0
Learned Composition	**51.60**± 0.65	**62.51**± 2.02	72.74± 0.41	77.54± 1.27	80.30± 0.61	81.82± 0.59	84.64± 0.74	108

Table 3. **Task shift results.** Here K represents the number of training samples for each class and |Θ| presents the total number of trainable parameters *excluding the classifier head.*

D. Analysis

D.1. Visualization of Learned Composition Weights

In Figure 5, we present visualizations of the learned composition vectors **v** and the CKA heatmap for the "query" and "value" weight matrices of attention modules across all layers of the ViT-base model. For this analysis, Food101 serves as the downstream task, while Stanford Cars, SUN397, Eurosat, CIFAR100, and Flowers are selected as the upstream tasks. The CKA values are normalized across the upstreams (x-axis).



(a)

(b)

Figure 5. Visualization of the learned composition weights **v** (a) and the CKA similarity map (b) for the Query and Value weight matrix of ViT-base. Here, the x-axis represents the upstream LoRA module and the y-axis represents the layer number. We can observe a high correlation between the upstream module picked by learned composition and the CKA similarity of upstream and downstream tasks.