π-Tuning: Transferring Multimodal Foundation Models with Optimal Multi-task Interpolation

Chengyue Wu [1 2] **Teng Wang** [1 2] **Yixiao Ge** [2] **Zeyu Lu** [3] **Ruisong Zhou** [4] **Ying Shan** [2] **Ping Luo** [1]

Abstract

Foundation models have achieved great advances in multi-task learning with a unified interface of unimodal and multimodal tasks. However, the potential of such multi-task learners has not been exploited during transfer learning. In this work, we present a universal parameter-efficient transfer learning method, termed **P**redict-**I**nterpolate Tuning (π-Tuning), for vision, language, and vision-language tasks. It aggregates the parameters of lightweight task-specific experts learned from similar tasks to aid the target downstream task. The task similarities are predicted in a unified modality-independent space, yielding a scalable graph to demonstrate task relationships. π-Tuning has several appealing benefits. First, it flexibly explores both intra- and inter-modal transferability between similar tasks to improve the accuracy and robustness of transfer learning, especially in data-scarce scenarios. Second, it offers a systematical solution for transfer learning with multi-task prediction-and-then-interpolation, compatible with diverse types of parameter-efficient experts, such as prompt and adapter. Third, an extensive study of task-level mutual benefits on 14 unimodal and 6 multimodal datasets shows that π-Tuning surpasses fine-tuning and other parameter-efficient transfer learning methods both in full-shot and low-shot regimes. The task graph also enables an in-depth interpretable analysis of task transferability across modalities. The code will be available at https://github.com/TencentARC/pi-Tuning.

[1]Department of Computer Science, The University of Hong Kong [2]ARC Lab, Tencent PCG [3]Shanghai Jiao Tong University [4]School of Mathematical Sciences, Fudan University. Correspondence to: Yixiao Ge <yixiaoge@tencent.com>.
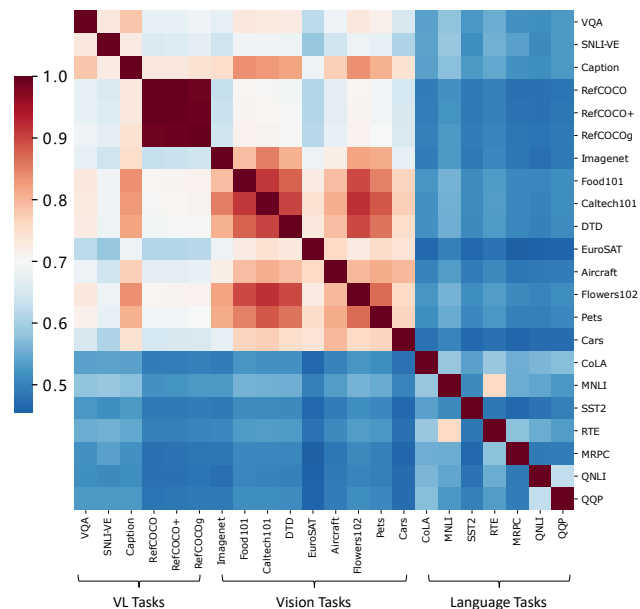
Figure 1. Heatmap of the predicted task similarities, composed of both unimodal and multimodal tasks. Vision-language tasks are more similar to vision tasks compared to language tasks. Best viewed in color.

1. Introduction

With the development of Transformer architectures (Dosovitskiy et al., 2021; Devlin et al., 2018; Brown et al., 2020), foundation models (Cho et al., 2021; Lu et al., 2022; Wang et al., 2022) pre-trained with large-scale data are capable of multiple tasks across modalities in a unified sequence-to-sequence manner, taking one more step toward mimicking the human brain. These foundation models are natural multi-task learners with universal representation and I/O interfaces for both unimodal and multimodal tasks. But unfortunately, these properties have not been fully exploited in downstream tasks, as few studies investigated how to properly transfer these models.

In this work, we tackle the problem of transfer learning of multimodal foundation models with unified sequence-to-sequence interfaces. Most of our experiments are based on OFA (Wang et al., 2022), an open-source model, without

loss of generality. Some previous attempts (Pruksachatkun et al., 2020) empirically observed that pre-finetuning with similar tasks may be beneficial while dissimilar tasks are harmful. It is intuitive to leverage auxiliary tasks that are similar to the target domain to boost transfer learning. The measurement of task similarities turns out to be a critical problem. Rather than the brute-force probing as in Pruksachatkun et al. (2020), we embed tasks into a unified space with Fisher Information Matrix (Achille et al., 2019), yielding task graphs across modalities (see Fig. 1). We are the first to explore task relationships among computer vision (CV), natural language processing (NLP), and vision-language (VL) tasks. The graph is computationally efficient and easily scalable for new tasks, which is especially suitable for recent arts that unify increasingly more tasks in one model.

Given the similar tasks predicted from the computed graph, naïve multi-task fine-tuning is effective for achieving satisfactory performance but inefficient for training, especially with increasing tasks and model sizes. Inspired by the parameter-efficient transfer learning methods (Houlsby et al., 2019; Hu et al., 2021; Li & Liang, 2021) with only a few trainable parameters, we propose **P**redict-**I**nterpolate Tuning (π-Tuning), which interpolates between predicted parameter-efficient experts (*e.g.*, adapter, prompt) for transfer learning. To be specific, as demonstrated in Fig. 2, the parameter-efficient experts are trained individually on each task before being selected according to the predicted task similarities. The weights of selected experts are then efficiently tuned to be ensembled for the downstream task. We empirically and theoretically found that those task-specific experts trained on similar tasks may lie in the basin of the loss landscape for the target task, which makes a simple interpolation of different experts possibly yielding strong performance, with no need for elaborately designed and highly parameterized fusion modules (Pfeiffer et al., 2020).

In addition to the newly proposed transfer learning method, several interesting findings were observed by macroscopically analyzing the relationship between CV, NLP, and VL tasks. 1) VL tasks are closer to CV tasks than NLP tasks, interpreting the state-of-the-art performance VL models achieved in vision benchmarks. 2) Image captioning sits in the central position among all the tasks, *i.e.*, being close to many tasks, demonstrating its importance in VL pre-training. These findings may well inspire future multimodal studies.

Some pioneer works (Vu et al., 2020; Poth et al., 2021; Vu et al., 2022) in NLP attempted to aggregate prompt embeddings from several source tasks to the target domain. However, they focus on limited language tasks and did not systematically come up with the philosophy of task integration, rendering limited scalability for a broader domain.

Our key contributions are three-fold:

- We for the first time investigate the task relationships among vision, language, and vision-language tasks in a unified task space. Building upon the task graph, interesting phenomena are demonstrated across modalities.

- We introduce a new parameter-efficient transfer learning method, namely π-Tuning, which effectively and efficiently aggregates unimodal and multimodal knowledge across tasks.

- Empirically, we conduct extensive experiments on vision, language, and vision-language downstream tasks to demonstrate the superiority, generalization, and scalability of our approach. Theoretically, we indicate the plausibility of similar task interpolation.

2. Method

2.1. Preliminary

We use a unified sequence-to-sequence (Seq2Seq) model (*e.g.*, OFA (Wang et al., 2022)) as it can process heterogeneous data via a universal interface, which is helpful for us to analyze multimodal tasks. We specify this model as $y = f_{\theta_0}(x)$, where θ_0 is the pre-treained weight of OFA. Here, we refer a task to a specific dataset as $\tau_i = (X_i, Y_i)$, where $i \in \{1, 2, \cdots, n\}$ represents the i-th task from all of n tasks. $X_i = \{x_i^1, x_i^2, \cdots, x_i^{s_i}\}$ is the input data, which can be CV, NLP, or VL data and s_i is the dataset size of τ_i. $Y_i = \{y_i^1, y_i^2, \cdots, y_i^{s_i}\}$ is the label set, which later will be processed by a verbalizer (Schick & Schütze, 2020) in OFA. We utilize parameter-efficient tuning methods, which keep the pre-trained weight θ_0 frozen and only tune extra added parameters φ to adapt to multiple downstream tasks. Given a task τ_i, we represent the corresponding optimized parameters as φ_i.

Task Definition. We have a task set $T = \{\tau_1, \tau_2, \cdots, \tau_n\}$ and the pre-trained model f_{θ_0}. For a specific task τ_i, the objective of traditional parameter-efficient tuning is of the following form:

$$\varphi_i \leftarrow \underset{\varphi_i}{\arg\min}\, L(\tau_i; \theta_0, \varphi_i), \tag{1}$$

where L represents the loss function. In the Seq2Seq model, it is typically the Cross-Entropy (CE) loss.

For the target task $\tau_t \in T$, we want to leverage all of the tasks and their corresponding trained parameters $\Phi = \{\varphi_1, \varphi_2, \cdots, \varphi_n\}$ in pursuit of more accurate and robust generalization performance.

Fisher Information Matrix (FIM). The Fisher information matrix (Amari, 1998) can indicate the sensitivity of the model towards a small perturbation and the curvature of the loss surface, which can be used as task embedding (Achille et al., 2019). Formally, it is the expected covariance of the gradients of the log-likelihood for model parameters θ,
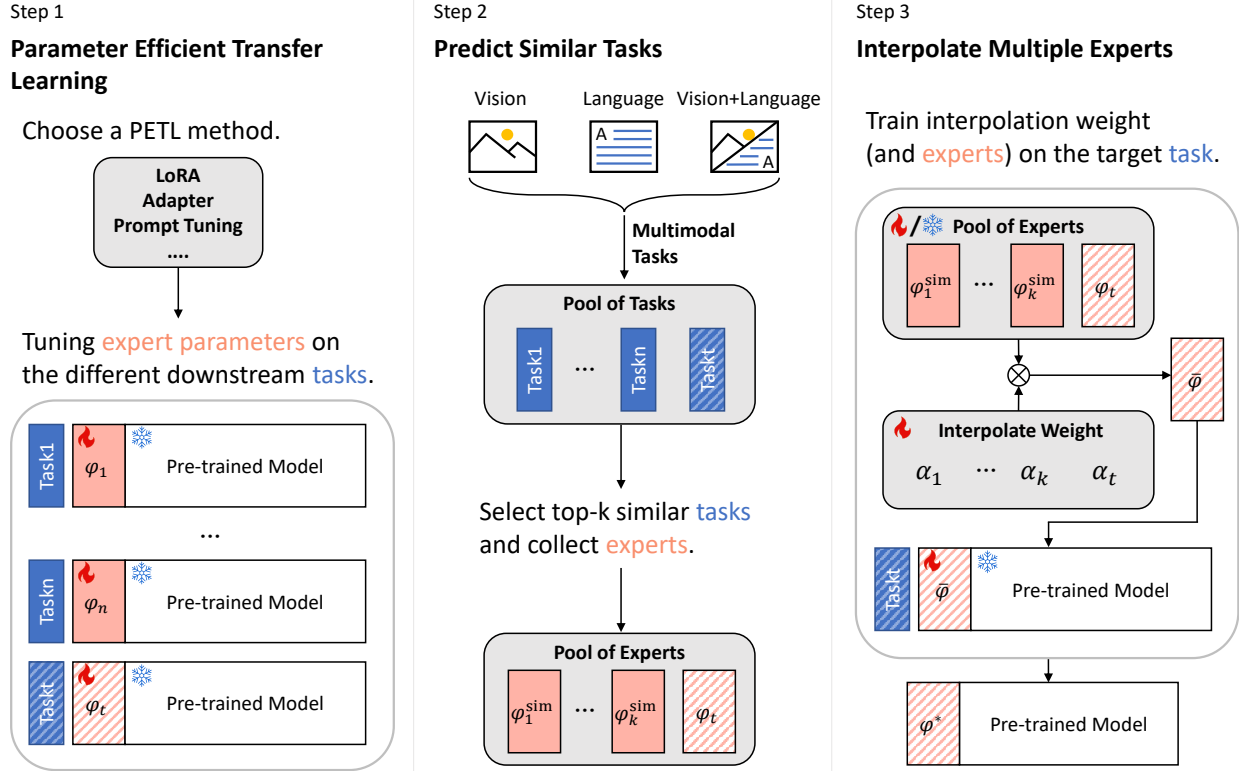
Figure 2. Overview of the π-Tuning method. Step 1 is the traditional parameter-efficient transfer learning (PETL) pipeline. Given the target task, π-Tuning further adds Step 2 and Step 3 to utilize task relationships to enhance the target expert. Specifically, we find the most similar tasks in a large pool of tasks and interpolate those experts with our target expert on the target task.

where $P(x, y)$ represents the data distribution of the task:

$$F_\theta^i = \mathop{\mathbb{E}}_{(x,y) \sim P_\theta(x,y)} \nabla_\theta \log P_\theta(y|x) \nabla_\theta \log P_\theta(y|x)^{\mathrm{T}}. \quad (2)$$

Specifically, we use the empirical Fisher to compute task embedding for task τ_i:

$$F_\theta^i = \frac{1}{s_i} \sum_{j=1}^{s_i} [\nabla_\theta \log P_\theta(y_i^j|x_i^j) \nabla_\theta \log P_\theta(y_i^j|x_i^j)^{\mathrm{T}}]. \quad (3)$$

2.2. π-Tuning

Our core hypothesis is that similar tasks help enrich domain-relevant or task-relevant training data, while dissimilar tasks may be harmful to the model to perform well in the target task. Based on this hypothesis, we first retrieve a subset of similar tasks given the target task τ_t of the task set T. Then we combine parameter-efficient experts trained for those similar tasks, which is considered to contain useful knowledge towards corresponding tasks, with the target task expert via interpolation to seek a transfer gain. We refer to this approach as Predict-Interpolate Tuning (π-Tuning for short). Fig. 2 shows the overall framework.

Task similarity prediction. Since the combination number of the task set T is exponential for n, it is computationally expensive to identify the subset $R \subseteq T$ of similar tasks given a target task brutally. Therefore, we try to embed each task into a vector to compute task similarity. We choose the FIM to extract the task feature as mentioned in Sec. 2.1. We keep the pre-trained weights θ_0 frozen and tune φ_i for each task τ_i via parameter-efficient tuning.

Next, given task τ_i, we compute the task embedding based on the Fisher of the task expert's weights φ. As the dimension of F_θ is extremely large, we make an approximation by only considering its diagonal entries following the practice in Achille et al. (2019). The task embedding F is therefore measured via $F = \mathrm{diag}(F_\theta) = [F_{\theta_1}, F_{\theta_2}, \cdots, F_{\theta_n}]$, where θ_i denotes the i-th parameter of the expert. Intuitively, the approximation is made under the assumption that correlations between different parametric modules in the expert are not essential.

We compute the cosine similarity between the embedding of the target task τ_t and the candidate source task. We then rank the similarity score in descending order to select Top-k similar tasks to form the subset $R = \{\tau_1^{\mathrm{sim}}, \tau_2^{\mathrm{sim}}, \cdots, \tau_k^{\mathrm{sim}}\}$. Both the task set T and the similar task subset R are *dy-*

Method	RefCOCO			RefCOCO+			RefCOCOg		SNLI-VE		COCO Captions				VQA	
	val	testA	testB	val	testA	testB	val-u	test-u	dev	test	B@4	M	C	S	test-dev	test-std
Previous SOTAs																
UNITER (Chen et al., 2019)	81.41	87.04	74.17	75.90	81.45	66.70	74.86	75.77	79.40	79.40	-	-	-	-	73.80	74.00
VILLA (Gan et al., 2020)	82.39	87.48	74.84	76.17	81.54	66.84	76.18	76.71	80.20	80.00	-	-	-	-	74.70	74.90
MDETR (Kamath et al., 2021)	86.75	89.58	81.41	79.52	84.09	70.62	81.64	80.89	80.90	81.20	-	-	-	-	77.70	77.60
VL-T5 (Cho et al., 2021)	-	-	-	-	-	-	71.20	71.30	-	-	34.50	28.70	116.5	21.90	-	70.30
UNICORN (Yang et al., 2021)	88.29	90.42	83.06	80.30	85.05	71.88	83.44	83.93	-	-	35.80	28.40	119.10	21.50	-	-
OFA-Base																
Finetuning (Wang et al., 2022)	88.48	90.67	83.30	81.39	87.15	74.29	82.29	82.31	89.30	89.20	41.00	30.90	138.2	24.20	78.00	78.10
BitFit (Zaken et al., 2021)	76.32	81.21	72.80	67.29	74.14	59.21	68.79	69.61	84.84	84.48	39.80	30.20	134.6	23.86	73.03	73.26
LoRA (Hu et al., 2021)	81.91	85.89	76.90	72.29	79.22	62.28	72.55	73.26	87.83	87.93	39.80	30.20	134.5	23.73	75.57	75.67
Prompt Tuning (Yang et al., 2022)	84.53	85.21	77.36	76.34	81.44	67.68	75.61	76.57	88.18	88.59	39.70	30.10	134.2	23.50	74.31	74.47
Adapter (Houlsby et al., 2019)	86.63	**90.01**	81.71	79.45	84.89	71.36	79.58	80.35	87.90	87.67	39.80	30.60	134.6	23.80	75.59	75.94
π-Adapter	**86.98**	89.99	**81.73**	**80.10**	**85.87**	71.38	**81.72**	**81.75**	**89.23**	89.40	**41.00**	**30.90**	**137.0**	**23.90**	75.88	**76.13**
OFA-Large																
Finetuning	90.05	92.93	85.26	84.60[*]	89.99[*]	77.71[*]	85.89	86.55	90.36[*]	89.91[*]	41.90[*]	31.40[*]	141.8[*]	24.50[*]	80.40	80.70
BitFit	89.61	92.20	84.91	82.60	88.08	75.16	84.66	84.68	89.70	89.42	41.02	30.92	138.8	24.23	78.23	78.44
LoRA	89.56	92.59	84.63	83.00	88.70	75.46	84.48	85.01	89.49	89.15	41.50	31.10	140.4	24.40	78.20	78.16
Prompt Tuning	90.05	92.31	85.59	84.54	89.40	77.77	85.27	85.89	89.19[*]	89.11[*]	41.60[*]	30.80[*]	140.5[*]	24.30[*]	78.30	78.53
Adapter	90.05	92.42	84.83	84.50	89.66	77.26	85.48	85.88	90.04	89.59	**41.80**	31.30	140.6	**24.50**	78.55	78.62
π-Adapter	**90.49**	**92.93**	**85.91**	**84.92**	**90.03**	**77.91**	**86.60**	**86.92**	**90.16**	**90.01**	41.70	**31.40**	140.7	**24.50**	**78.78**	**78.82**

Table 1. Comparison with state-of-the-art (SOTA) PETL and full fine-tuning methods. We report the experimental results on RefCOCO, RefCOCO+, RefCOCOg, SNLI-VE, COCO Image Captioning, and VQA. The overall best result is underlined while **bold** signifies the best among parameter-efficient methods. [*] denotes the results of our re-trained models based on official codes.

namic. If there is a new task arrives, we can add it to our task set and compute its task similarity given the target task to check if it is the top-k similar task.

Interpolation of multiple lightweight experts. Given the similar task subset R, we retrieve the expert subset corresponding to R, *i.e.*, $\phi = \{\varphi_1^{\text{sim}}, \varphi_2^{\text{sim}}, \cdots, \varphi_k^{\text{sim}}\} \subseteq \Phi$.

Then we can derive the combined parameters $\bar{\varphi}$ as follow:

$$\bar{\varphi} = \text{softmax}(\alpha)_0 \varphi_t + \sum_{i=1}^{k} \text{softmax}(\alpha)_i \varphi_i^{\text{sim}}. \quad (4)$$

Then, we may further tune $\bar{\varphi}$ given the target task τ_t[1]:

$$\varphi^* \leftarrow \arg\min_{\bar{\varphi}} L(\tau_t; \theta_0, \bar{\varphi}). \quad (5)$$

There will be no further inference latency introduced by π-Tuning as the dimension of φ^* is as same as before.

Relationship between FIM and landscape. We will provide complementary analytical analysis into the effectiveness of π-Tuning to show that tasks with similar FIM lead to close local optimums so that the similarity of FIM can be a nice indicator for the effectiveness of interpolation. For simplicity, we consider two tasks τ_1 and τ_2

with loss function of cross-entropy, which is formulated as $L = \mathbb{E}_{(x,y)\sim P}[-\log P_\varphi(y|x)]$. Let φ_1 and φ_2 be the corresponding local minimum for τ_1 and τ_2, optimized from the same initial parameters φ_0, respectively. The FIM of each task can be seen as the negative Hessian matrix of loss, which serves as the second derivative of the loss function.

We now state our main theorem in Theorem 2.1, demonstrating the distance between φ_1 and φ_2 can be bounded when they have similar non-singular FIM along the linear path and their gradient at φ_0 is close. The detailed derivation is presented in Appendix B.

Theorem 2.1. *Assume that the gradient of two tasks at the initial parameters φ_0 is close and we have similar non-singular FIM of two tasks along the linear path. Then the gap between two local minimal can be controlled by a constant C, i.e., $\|\varphi_1 - \varphi_2\|_2 \leq C$.*

We note that, as previous works revealed that fine-tuned checkpoints initialized from the same pre-train model lie in the same basin of the error landscape (Neyshabur et al., 2020), we further found that PETL methods trained for similar tasks also have this property. Our method extends weight averaging on fine-tuned checkpoints under varied hyperparameter configurations for a single task (Wortsman et al., 2022) to various PETL methods and various tasks. As combined from different domains, our method shows a more robust performance compared to fine-tuning and original PETL methods when distribution shifts.

---

[1]We achieve better performance when we tune $\phi \cup \{\varphi_t\}$ in addition to α so that our default setting is tuning both of φ and α. Details in Sec. 3.5.

Method	Food101	Caltech101	DTD	EuroSAT	Aircraft	Flowers102	Pets	Cars	AVG
Multimodal Pretrained Baseline Models									
CLIP (Radford et al., 2021)	85.49	93.76	73.40	95.70	40.02	94.94	79.61	62.84	78.22
FLAVA (Singh et al., 2022)	88.51	95.74	77.29	97.26	47.31	96.37	84.82	70.87	82.27
16-shot on OFA-Base									
Adapter	69.27	92.33	54.31	31.49	31.32	93.06	77.81	42.10	61.46
π-Adapter	69.85(+0.58)	92.74(+0.41)	57.33(+3.02)	36.49(+5.00)	43.71(+16.39)	94.52(+1.46)	79.39(+1.58)	54.04(+11.94)	66.01(+4.55)
full data on OFA-Base									
Adapter	85.77	95.17	72.75	93.01	45.24	97.52	89.26	53.71	79.05
π-Adapter	86.16(+0.39)	95.82(+0.65)	73.70(+0.95)	93.94(+0.93)	52.42(+7.18)	98.05(+0.53)	90.35(+1.09)	61.30(+7.59)	81.47(+2.42)

Table 2. Experimental results on eight common vision tasks. We evaluate π-Tuning in both few-shot and full-data scenarios. The predicted auxiliary experts for vision tasks all contain image captioning, verifying the cross-modal transfer benefits.

Method	MNLI	QQP	MRPC	QNLI	RTE	SST2	AVG
Multimodal Pretrained Baseline Models							
VisualBERT (Li et al., 2019)	81.6	89.4	71.9	87.0	56.6	89.4	79.3
UNITER (Chen et al., 2020)	80.9	89.2	69.3	86.0	55.6	89.7	78.5
Uni-Perceiver (Zhu et al., 2022)	81.7	87.1	86.6	89.9	64.3	90.2	83.3
zero shot on OFA-Large							
OFA-L	37.12	37.31	62.99	49.50	49.46	55.85	36.53
π-Adapter	44.36(+7.24)	61.68(+24.37)	69.85(+6.86)	55.98(+6.48)	51.99(+2.53)	56.77(+0.92)	42.58(+6.05)
full data on OFA-Large							
Adapter	85.99	90.70	87.50	92.49	72.20	93.81	87.12
π-Adapter	86.06(+0.07)	91.16(+0.46)	87.75(+0.25)	92.66(+0.17)	76.53(+4.33)	93.81(+0.00)	88.00(+0.88)
full data on T5-Base							
Adapter	86.03	91.02	89.71	92.51	73.57	94.72	87.93
π-Adapter	86.19(+0.16)	91.13(+0.11)	90.20(+0.49)	92.62(+0.11)	82.86(+9.29)	95.07(+0.35)	89.68(+1.75)

Table 3. Experimental results on natural language understanding tasks from the GLUE benchmark (Wang et al., 2018). We experiment with π-Tuning in both zero-shot and full-data settings with respect to two backbone models, OFA and T5.

3. Experiments

This section presents our key experimental findings. We begin with experimental settings (described in Sec. 3.1), and then verify the effectiveness of π-Tuning on both cross-modal and uni-modal tasks (described in Sec. 3.2). Next, we give the rationality of the similarity measurement and the interpolation operation (described in Sec. 3.3). Afterward, we study the task-level transferability of the model after π-tuning(described in Sec. 3.4). Finally, ablation studies of the key design choices are presented (described in Sec. 3.5).

3.1. Experimental Settings

Implementation details. We choose the vision-language foundation model OFA (Wang et al., 2022) as the pretrained model, with the mostly-used base-size and large-size version, whose parameters are 180M and 470M, respectively. We set the default number of auxiliary experts $k = 2$. We follow the few-shot data split used in Zhou et al. (2022a). More experimental details are provided in Appendix A.

Multimodal task pool. We evaluate π-Tuning on a task pool across vision, language, and VL tasks. For vision tasks, we choose 8 common vision recognition tasks. For language

tasks, we evaluate the model on 8 tasks from GLUE (Wang et al., 2018). For VL tasks, we experiment on both understanding and generation tasks, including RefCOCO, Ref-COCO+ (Yu et al., 2016), RefCOCOg (Mao et al., 2016), VQAv2 (Goyal et al., 2017), SNLI-VE (Xie et al., 2019) and COCO image captioning (Chen et al., 2015). We follow the evaluation metrics in Wang et al. (2022) for VL and language tasks and Yang et al. (2022) for vision tasks.

Compared methods. We compare π-tuning with finetuning and other four types of state-of-the-art PETL methods:

- Bitfit (Zaken et al., 2021) optimizes bias terms in all linear layers at every Transformer layer.

- Adapter tuning (Houlsby et al., 2019) insert adapters between transformer layers, consists of a down-project $W_{\text{down}} \in \mathbb{R}^{h \times r}$, followed by a nonlinear activation function, and an up-project $W_{\text{up}} \in \mathbb{R}^{r \times h}$, where h is the hidden size of the transformer model and r is a hyperparameter of adapters as bottleneck dimension. We set $r = 128$ in all experiments.

- Prompt tuning (Li & Liang, 2021) prepends vectors

Method	Training Time (GPU hours)	Throughput (samples/sec)	Deployment Params (%)	Training Params (%)
Bitfit	203	19.04	0.04%	0.04%
LoRA	215	17.78	0.99%	0.99%
Prompt Tuning	292	18.45	1.03%	1.03%
Adapter	344	18.60	2.60%	2.60%
π-Adapter	17	18.60	2.60%	7.42%

Table 4. Computational costs of different methods on RefCOCO, including the parameter proportion for tunable and deployment parts of the network, as well as wall time during training versus inference. Training time is measured by A100 GPU hours. The inference cost is indicated by the throughput, which is the samples processed per second by a single A100 GPU.

Method	RefCOCO			RefCOCO+			RefCOCOg	
	val	testA	testB	val	testA	testB	val-u	test-u
Prompt Tuning	84.53	85.21	77.36	76.34	81.44	67.68	75.61	76.57
π-Prompt	**85.75**	**88.85**	**79.67**	**77.84**	**83.09**	**69.61**	**77.41**	**78.06**
LoRA	81.91	85.89	76.90	72.29	79.22	62.28	72.55	73.26
π-LoRA	**85.82**	**88.97**	**81.77**	**78.41**	**83.95**	**69.22**	**77.53**	**78.38**

Table 5. π-Tuning results based on Prompt Tuning and LoRA for the base size model on referring expression comprehension datasets, indicating that π-Tuning is agnostic to PETL methods and can provide a consistent gain.

to keys and values of the attention module at every layer. Specifically, there is a tunable prompt vector embedding $P \in \mathbb{R}^{L \times l \times h}$, where L is the number of layers and l is a hyper-parameter to define prefix vector length, to retrieve each prefix vector at every layer. We follow the setting of l in Yang et al. (2022).

- LoRA (Hu et al., 2021) decomposes weight matrices of transformer layers as trainable parameters to approximate the weight updates. For a linear projection $h = Wx$ where $W \in \mathbb{R}^{d \times s}$, LoRA adds two trainable matrices $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times s}$, where the rank is a hyperparameter and $r \ll \min(d, s)$ so that the linear projection is modified as $h = Wx + BAx$. We set $r = 16$ in all experiments.

We refer π-tuning applied on adapters to π-Adapter, similarly for π-Prompt and π-LoRA. We primarily use π-Adapter for experiments since its superior performance.

3.2. Comparison with PETL Methods

Multi-/uni-modal tasks. Table 1 shows the results on 6 multimodal datasets of π-tuning as well as previous SOTA results. We can see that (1) π-Tuning improves original adapters across all tasks consistently; (2) π-Tuning outperforms all the other parameter-efficient tuning methods across all tasks, without introducing additional inference latency, indicating that π-Tuning is stronger in storage-constrained scenarios; (3) π-Tuning achieves better or comparable results towards fully finetuning on 5 out of 6 datasets for the large-size model, while using significantly fewer trainable parameters. Although inferior to finetuning for base size, π-Tuning still offers non-trivial performance gain for PETL methods, similar to the results in Yang et al. (2022).

We further show π-Tuning is competitive on uni-modal tasks, including both vision and language tasks. The outcome is demonstrated in Table 2 and Table 3. π-Tuning achieves a consistent performance gain. Specifically, we

observe cross-modal transfer benefits in these experiments, where experts for image captioning can help with vision tasks and language entailment tasks can be benefited from experts for visual entailment. We attribute this to the similar semantics of tasks, as captioning can be regarded as a more detailed classification task and some visual entailment samples may directly rely on language entailment.

We list and discuss the computational costs of different PETL methods during training versus inference. As demonstrated in Table 4, our π-Adapter shows the same throughput and parameters during inference as original adapters. π-Adapter requires more parameters for training due to the interpolation of multiple experts from similar tasks, which is the core insight of our paper. However, the training time for such an interpolation is quite short as the experts have been well-trained. We would like to clarify that the performance improvement of π-Adapter does not come from increasing tunable parameters. As shown in Table 7 of our paper, the performance decreases when solely scaling up the tunable parameters without initialization from experts of similar tasks, indicating the effectiveness of transferring knowledge from similar tasks' experts.

Few/zero-shot setting. To have a better understanding of the benefit of knowledge transferred from similar task experts, we experiment under two low-data settings, zero-shot natural language understanding and few-shot image classification to validate the effectiveness of π-Tuning when data is scarce. We show the 16-shot image classification results in Table 2, where π-Tuning brings more improvement compared to the full-data setting, with an average of 4.55% accuracy improvement across 8 tasks. For the zero-shot setting, as we can not derive the target expert, we test the zeros-shot transfer performance of the target task using the expert of its nearest neighbor (most similar task), The zero-shot natural language understanding results are presented in Table 3, where the average improvement across tasks is more than that under the full-data setting.
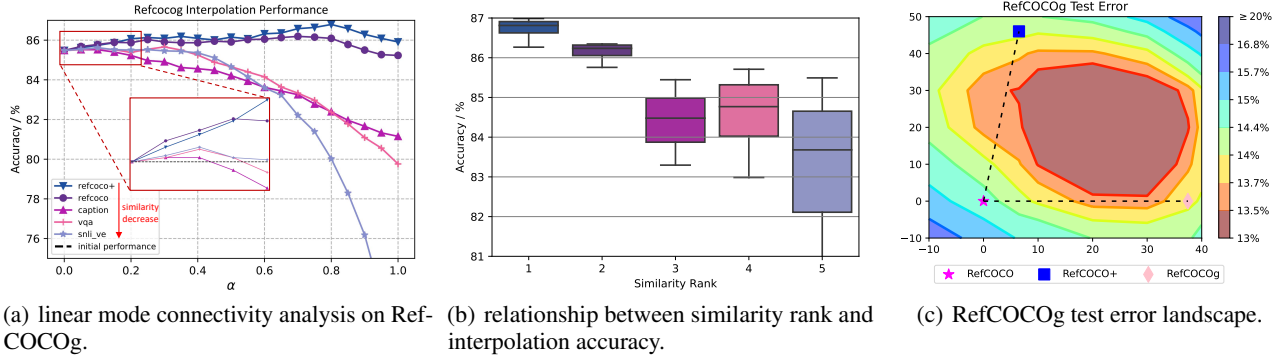
(a) linear mode connectivity analysis on Ref-COCOg.

(b) relationship between similarity rank and interpolation accuracy.

(c) RefCOCOg test error landscape.

Figure 3. (a) Accuracy when linearly interpolating between the parameter-efficient tuning checkpoint of RefCOCOg and checkpoints of other tasks (RefCOCO+ and RefCOCO). The advantage of interpolating is correlated with the FIM similarity between tasks. The interpolation of similar tasks' performance maximum is higher than dissimilar tasks and needs a greater interpolation weight. (b) Direct transfer performance (accuracy at $\alpha = 1$) also correlates to FIM similarity. (c) The test error surface on RefCOCOg (Mao et al., 2016), as a function of model weights in a two-dimensional subspace[3]. It shows that checkpoints of similar tasks lie in the same basin of the surface. Interpolation of those checkpoints could achieve lower test error on the surface. The visualization follows Garipov et al. (2018), which derives an orthonormal basis $\hat{u}, \hat{v}$ by three checkpoints.

Pretrained foundation models. We also extend π-Tuning on a uni-modal pretrained foundation model, T5 (Raffel et al., 2020), to solve natural language understanding tasks. We demonstrate the results in Table 3, where we find that RTE is the task that benefits most with an improvement of 9.29% accuracy.

Experts from PETL methods. In Table 5, experiments on RefCOCO, RefCOCO+, and RefCOCOg datasets with the base-size model show that π-Tuning is agnostic to PETL methods, providing significant and consistent performance gain for LoRA and Prompt Tuning. π-LoRA improves a large margin of 4.20% on average accuracy across three datasets towards original LoRA and 1.94% for π-Prompt towards original Prompt Tuning.

3.3. Task Relationships

Linear mode connectivity. As defined in Frankle et al. (2020), two parameters w_1 and w_2 are linear mode connectivity if the error barrier (Garipov et al., 2018; Draxler et al., 2018) height ≈ 0 along the linear path between them. To provide our intuition, we first study the performance along the linear path between the target task expert φ_t and experts for other tasks from the task set to see whether they have linear mode connectivity. We use adapters as parameter-efficient experts to store task information and RegCOCOg as the target task. We vary interpolation coefficient α from 0 to 1, with an interval of 0.05, and the combined expert $\bar{\varphi}$ is given by $(1 - \alpha)\varphi_t + \alpha\varphi, \varphi \in \Phi$, which is a special case of $k = 1$ as we defined in Sec. 2.2.

As shown in Fig. 3(a), the model can benefit from interpolation compared to the original target task expert φ_t when $\alpha = 0$ and there is linear mode connectivity between the target task expert and experts of similar tasks, while for dissimilar tasks there is large error barrier along the linear path. Furthermore, these results suggest that (1) experts with higher similarity need a greater interpolation coefficient, and (2) direct transfer ($\alpha = 1$) performance and maximum interpolation performance correlate with similarity.

Task similarity measurement. To investigate the correlation between task similarity and interpolation performance, we consider interpolation performance on the validation and test split of RefCOCO, RefCOCO+, and RefCOCOg. We compare the average accuracy across those splits for the interpolated experts with different similarities. The results are illustrated in Fig. 3(b), where the interpolation accuracy increases when experts with higher similarity to the target task are interpolated. We also visualize the task similarity of each task pair from our task space, which consists of both unimodal and multimodal tasks, in Fig. 1. The results illustrate that task embedding of vision tasks is closer to vision&language tasks compared to language and the embedding of image captioning task is similar to almost all the vision and VL tasks. We assume that is one of the reasons why image captioning is important in image-text multimodal pretraining.

Error landscape visualization. We also visualize two slices of the test error landscape when interpolating experts in Fig. 3(c), where the error contours are basin-shaped and none of the individual domain-specific experts is optimal. The results suggest that interpolation may reach a better point in the basin of the landscape.

3.4. Cross-Task Transferability

Distribution shift. The boxplot in Fig. 4 summarizes the relative performance drop of π-Adapter due to test distri-
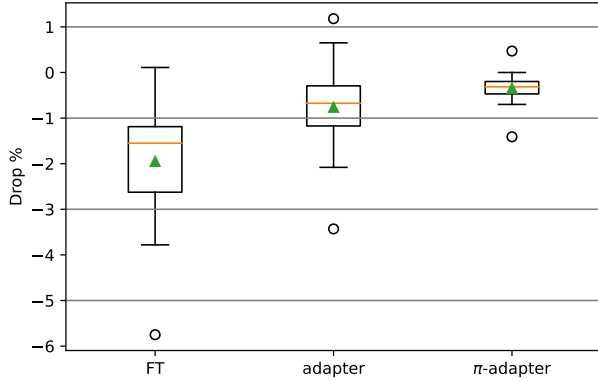
Figure 4. Relative performance drop (%) of π-Adapter compared to domain-specific models on referring expression comprehension datasets (RefCOCO, RefCOCO+, RefCOCOg). FT represents finetuning and the green triangles denote the mean.

Method	RefCOCO			RefCOCO+			RefCOCOg	
	val	testA	testB	val	testA	testB	val-u	test-u
OFA-Base								
Adapter	86.34	89.61	80.82	74.60	83.20	69.79	79.74	80.65
π-Adapter	**87.12**	**90.30**	**82.16**	**79.46**	**84.63**	**71.43**	**80.84**	**82.00**
OFA-Large								
Adapter	90.00	92.88	85.24	83.81	89.08	76.54	85.99	85.96
π-Adapter	**90.55**	**93.12**	**85.85**	**84.77**	**90.31**	**77.68**	**86.91**	**86.88**

Table 6. π-Tuning can improve the performance compared to directly optimization in a multitask setting, achieving task specific model's performance across all task splits.

bution shift compared to domain-specific models, which are optimized directly on the test domain. Specifically, we conduct experiments on referring expression comprehension datasets. As combined with experts trained from other domains, π-Adapter shows more robust performance than adapters and fine-tuning trained on the specific domain when the test distribution is shifted. Even though π-Adapter is further tuned on another specific domain, it still can be benefited from the interpolation with experts trained from other domains, while direct optimization of adapters or fine-tuning all the parameters may fit on the specific domain, lacking domain generalization ability.

Multi-task learning. As interpolation with experts of different tasks can be regarded as an implicit multitask learning, we experiment with π-Tuning in a multitask setting, demonstrated in Table 6 shows. Compared to direct multitask learning on different tasks, we observe that π-Tuning performs better in all task splits, even exceeding the task-specific model's performance.

Method	RefCOCO			RefCOCO+			RefCOCOg	
	val	testA	testB	val	testA	testB	val-u	test-u
w/o init.	89.95	92.36	84.79	83.81	89.31	76.87	85.36	85.68
only scale	**90.67**	92.75	85.55	84.64	89.71	77.03	86.34	86.75
π-Adapter	90.49	**92.93**	**85.91**	**84.92**	**90.03**	**77.91**	**86.60**	**86.92**

Table 7. Ablation of π-Tuning. "w/o init." denotes we randomly initialize auxiliary experts and "only scale" means we only tune the interpolation weight
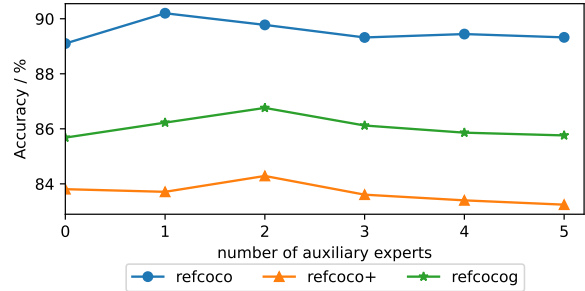

Figure 5. The ablation of the number of experts used in π-Tuning. The performance first rises to a maximum when the number of auxiliary experts increases and then gradually falls.

3.5. Ablation Study

The ablation of π-Tuning. The ablation study of the proposed method is shown in Table 7. "w/o init." denotes we randomly initialize auxiliary experts, which performs even worse than the original adapter. "Only scale" means we only tune the interpolation weight α, achieving better results than the original adapter while slightly worse than π-Adapter, which is reasonable as it only updates a subspace of parameters in π-Tuning.

The ablation of the number of experts. We experimentally explored the relationship between performance on 3 target tasks and the number of auxiliary experts for each of them, demonstrated in Fig. 5. We observe that the performance first rises to a maximum when k increases and then gradually falls, and the best value of k is around 2. We conjecture that the reason may lie in that as interpolation is conducted in descending order of task similarity, early interpolation with highly similar tasks can improve performance while following mixing dissimilar experts has negative inference on the target task.

4. Scope and Limitations

While this work has so far demonstrated that interpolation of the experts learned from similar tasks is a useful technique for improving accuracy and robustness, this section explores the limitations of the approach which are expected to be addressed in future studies.

Task similarity measurement. The task similarities in π-Tuning are measured by a diagonal approximation of FIM following the practice in (Achille et al., 2019). While it works well for multi-task interpolation in our paper, there can be other solutions without the unmanageable computational overhead of a full FIM. Recently, Vu et al. (2022); Zhou et al. (2022b) demonstrate that the parameters of the expert itself can be regarded as the task embedding, which can be a substitute solution to be applied to our approach.

Applicability. We verify the effectiveness of π-Tuning on OFA-Base/Large and T5-Base foundation models due to the limitation of computational resources. We notice that parameter-efficient experts work well in the field of AIGC, like Mou et al. (2023), which combines several adapters tuned for different conditions to achieve controllable image synthesis. More evaluations on much larger backbones and more fields would make the work more practical. And the number of tasks used for interpolation is manually designated in the current version. It is promising to determine the number of tasks adaptively.

5. Related Work

Parameter efficient transfer learning. In recent years, large-scale models pre-training on huge datasets have shown great capability in downstream tasks. However, the cost of directly finetuning the large-scale models is expensive in memory and storage. To mitigate this problem, researchers proposed several PETL methods to adapt large-scale pre-trained models with a few trainable parameters. Houlsby et al. (2019) proposed inserting adapter layers between transformer layers. Hu et al. (2021) proposed injecting trainable low-rank matrices into transformer layers to approximate the weight updates. Li & Liang (2021); Lester et al. (2021); Liu et al. (2021) proposed optimizing the input word embeddings. However, these PETL methods tend to be limited to the target downstream task, ignoring the potential complementarity between different tasks and modalities.

Identifying beneficial task relationships. Poth et al. (2021); Aghajanyan et al. (2021); Lu et al. (2020) have shown that pre-finetuning, an additional large-scale multitask learning stage between pre-training and finetuning, can significantly improve performance on the downstream tasks. However, multitask learning may appear conflicts between different tasks, or even have an adverse impact on target tasks. To identify beneficial task relationships, Bingel & Søgaard (2017) rely on features derived from learning curves and Alonso & Plank (2017) proposed using characteristics of datasets. Zamir et al. (2018) proposed representing the CV task relationships as an affinity matrix and optimizing the affinity matrix to get the most training-efficient task set. Achille et al. (2019); Vu et al. (2020) proposed using Fisher information matrix to extract task relationships.

Averaging model weights. Ensembling the outputs of multiple models is an important method for improving the performance of deep learning models (Dietterich, 2000; Lakshminarayanan et al., 2017). However, when the size of the model is huge, ensembling the outputs of multiple models would be prohibitively expensive. Unlike model ensembles, averaging the weights of models is a foundational technique in convex optimization and deep learning without any extra costs at inference time. Neyshabur et al. (2020) find that when two models are fine-tuned from the same pre-trained initialization, the interpolated model would get at least the accuracy of the endpoints. Wortsman et al. (2022) proposed to produce a better model by averaging the weights of models finetuned with different hyperparameter configurations.

6. Conclusion

This paper presents π-Tuning, a new parameter-efficient transfer learning framework, to exploit universal representation across modalities by taking benefits from similar tasks. Given a target task, π-tuning first retrieves several similar tasks in arbitrary modalities and obtains a more transferable model by the interpolation of cross-modal experts. We theoretically and experimentally demonstrate the retrieved tasks tend to locate around the same basin on the error landscape. Experiments on diverse unimodal and multimodal tasks verify the cross-task transferability of the model in both full-data and low-data scenarios.

References

Achille, A., Lam, M., Tewari, R., Ravichandran, A., Maji, S., Fowlkes, C. C., Soatto, S., and Perona, P. Task2vec: Task embedding for meta-learning. In Proceedings of the IEEE/CVF international conference on computer vision, pp. 6430–6439, 2019.

Aghajanyan, A., Gupta, A., Shrivastava, A., Chen, X., Zettlemoyer, L., and Gupta, S. Muppet: Massive multitask representations with pre-finetuning. In Moens, M., Huang, X., Specia, L., and Yih, S. W. (eds.), Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pp. 5799–5811. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main. 468. URL https://doi.org/10.18653/v1/

2021.emnlp-main.468.

Alonso, H. M. and Plank, B. When is multitask learning effective? semantic sequence prediction under varying data conditions. In Lapata, M., Blunsom, P., and Koller, A. (eds.), Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers, pp. 44–53. Association for Computational Linguistics, 2017. doi: 10.18653/v1/e17-1005. URL https://doi.org/10.18653/v1/e17-1005.

Amari, S.-I. Natural gradient works efficiently in learning. Neural computation, 10(2):251–276, 1998.

Bingel, J. and Søgaard, A. Identifying beneficial task relations for multi-task learning in deep neural networks. In Lapata, M., Blunsom, P., and Koller, A. (eds.), Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers, pp. 164–169. Association for Computational Linguistics, 2017. doi: 10.18653/v1/e17-2026. URL https://doi.org/10.18653/v1/e17-2026.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901, 2020.

Chen, X., Fang, H., Lin, T.-Y., Vedantam, R., Gupta, S., Dollár, P., and Zitnick, C. L. Microsoft coco captions: Data collection and evaluation server. arXiv preprint arXiv:1504.00325, 2015.

Chen, Y., Li, L., Yu, L., Kholy, A. E., Ahmed, F., Gan, Z., Cheng, Y., and Liu, J. UNITER: universal image-text representation learning. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J. (eds.), Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXX, volume 12375 of Lecture Notes in Computer Science, pp. 104–120. Springer, 2020. doi: 10.1007/978-3-030-58577-8_7. URL https://doi.org/10.1007/978-3-030-58577-8_7.

Chen, Y.-C., Li, L., Yu, L., El Kholy, A., Ahmed, F., Gan, Z., Cheng, Y., and Liu, J. Uniter: Learning universal image-text representations. 2019.

Cho, J., Lei, J., Tan, H., and Bansal, M. Unifying vision-and-language tasks via text generation. In International Conference on Machine Learning, pp. 1931–1942. PMLR, 2021.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert:

Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

Dietterich, T. G. Ensemble methods in machine learning. In Kittler, J. and Roli, F. (eds.), Multiple Classifier Systems, First International Workshop, MCS 2000, Cagliari, Italy, June 21-23, 2000, Proceedings, volume 1857 of Lecture Notes in Computer Science, pp. 1–15. Springer, 2000. doi: 10.1007/3-540-45014-9_1. URL https://doi.org/10.1007/3-540-45014-9_1.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Draxler, F., Veschgini, K., Salmhofer, M., and Hamprecht, F. Essentially no barriers in neural network energy landscape. In International conference on machine learning, pp. 1309–1318. PMLR, 2018.

Frankle, J., Dziugaite, G. K., Roy, D., and Carbin, M. Linear mode connectivity and the lottery ticket hypothesis. In International Conference on Machine Learning, pp. 3259–3269. PMLR, 2020.

Gan, Z., Chen, Y.-C., Li, L., Zhu, C., Cheng, Y., and Liu, J. Large-scale adversarial training for vision-and-language representation learning. Advances in Neural Information Processing Systems, 33:6616–6628, 2020.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. Loss surfaces, mode connectivity, and fast ensembling of dnns. Advances in neural information processing systems, 31, 2018.

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and Parikh, D. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 6904–6913, 2017.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for nlp. In International Conference on Machine Learning, pp. 2790–2799. PMLR, 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.

Kamath, A., Singh, M., LeCun, Y., Synnaeve, G., Misra, I., and Carion, N. Mdetr-modulated detection for end-to-end multi-modal understanding. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1780–1790, 2021.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pp. 6402–6413, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/9ef2ed4b7fd2c810847ffa5fa85bce38-Abstract.html.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. In Moens, M., Huang, X., Specia, L., and Yih, S. W. (eds.), Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pp. 3045–3059. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.243. URL https://doi.org/10.18653/v1/2021.emnlp-main.243.

Li, L. H., Yatskar, M., Yin, D., Hsieh, C., and Chang, K. Visualbert: A simple and performant baseline for vision and language. CoRR, abs/1908.03557, 2019. URL http://arxiv.org/abs/1908.03557.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190, 2021.

Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., and Tang, J. GPT understands, too. CoRR, abs/2103.10385, 2021. URL https://arxiv.org/abs/2103.10385.

Lu, J., Goswami, V., Rohrbach, M., Parikh, D., and Lee, S. 12-in-1: Multi-task vision and language representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10437–10446, 2020.

Lu, J., Clark, C., Zellers, R., Mottaghi, R., and Kembhavi, A. Unified-io: A unified model for vision, language, and multi-modal tasks. arXiv preprint arXiv:2206.08916, 2022.

Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A. L., and Murphy, K. Generation and comprehension of unambiguous object descriptions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 11–20, 2016.

Mou, C., Wang, X., Xie, L., Zhang, J., Qi, Z., Shan, Y., and Qie, X. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. arXiv preprint arXiv:2302.08453, 2023.

Neyshabur, B., Sedghi, H., and Zhang, C. What is being transferred in transfer learning? Advances in neural information processing systems, 33:512–523, 2020.

Pfeiffer, J., Kamath, A., Rücklé, A., Cho, K., and Gurevych, I. Adapterfusion: Non-destructive task composition for transfer learning. arXiv preprint arXiv:2005.00247, 2020.

Poth, C., Pfeiffer, J., Rücklé, A., and Gurevych, I. What to pre-train on? efficient intermediate task selection. arXiv preprint arXiv:2104.08247, 2021.

Pruksachatkun, Y., Phang, J., Liu, H., Htut, P. M., Zhang, X., Pang, R. Y., Vania, C., Kann, K., and Bowman, S. R. Intermediate-task transfer learning with pretrained models for natural language understanding: When and why does it work? arXiv preprint arXiv:2005.00628, 2020.

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., Liu, P. J., et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res., 21(140):1–67, 2020.

Schick, T. and Schütze, H. It's not just size that matters: Small language models are also few-shot learners. arXiv preprint arXiv:2009.07118, 2020.

Singh, A., Hu, R., Goswami, V., Couairon, G., Galuba, W., Rohrbach, M., and Kiela, D. FLAVA: A foundational language and vision alignment model. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022, pp. 15617–15629. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01519. URL https://doi.org/10.1109/CVPR52688.2022.01519.

Vu, T., Wang, T., Munkhdalai, T., Sordoni, A., Trischler, A., Mattarella-Micke, A., Maji, S., and Iyyer, M. Exploring and predicting transferability across nlp tasks. arXiv preprint arXiv:2005.00770, 2020.

Vu, T., Lester, B., Constant, N., Al-Rfou, R., and Cer, D. Spot: Better frozen model adaptation through soft prompt transfer. In Proceedings of the 60th Annual Meeting of

the Association for Computational Linguistics (Volume 1: Long Papers), pp. 5039–5059, 2022.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue: A multi-task benchmark and analysis platform for natural language understanding. arXiv preprint arXiv:1804.07461, 2018.

Wang, P., Yang, A., Men, R., Lin, J., Bai, S., Li, Z., Ma, J., Zhou, C., Zhou, J., and Yang, H. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In International Conference on Machine Learning, pp. 23318–23340. PMLR, 2022.

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.

Xie, N., Lai, F., Doran, D., and Kadav, A. Visual entailment: A novel task for fine-grained image understanding. arXiv preprint arXiv:1901.06706, 2019.

Yang, H., Lin, J., Yang, A., Wang, P., Zhou, C., and Yang, H. Prompt tuning for generative multimodal pretrained models. arXiv preprint arXiv:2208.02532, 2022.

Yang, Z., Gan, Z., Wang, J., Hu, X., Ahmed, F., Liu, Z., Lu, Y., and Wang, L. Crossing the format boundary of text and boxes: Towards unified vision-language modeling. arXiv preprint arXiv:2111.12085, 2021.

Yu, L., Poirson, P., Yang, S., Berg, A. C., and Berg, T. L. Modeling context in referring expressions. In European Conference on Computer Vision, pp. 69–85. Springer, 2016.

Zaken, E. B., Ravfogel, S., and Goldberg, Y. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. arXiv preprint arXiv:2106.10199, 2021.

Zamir, A. R., Sax, A., Shen, W. B., Guibas, L. J., Malik, J., and Savarese, S. Taskonomy: Disentangling task transfer learning. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pp. 3712–3722. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00391. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Zamir_Taskonomy_Disentangling_Task_CVPR_2018_paper.html.

Zhou, K., Yang, J., Loy, C. C., and Liu, Z. Learning to prompt for vision-language models. International Journal of Computer Vision, 130(9):2337–2348, 2022a.

Zhou, W., Xu, C., and McAuley, J. Efficiently tuned parameters are task embeddings. arXiv preprint arXiv:2210.11705, 2022b.

Zhu, X., Zhu, J., Li, H., Wu, X., Li, H., Wang, X., and Dai, J. Uni-perceiver: Pre-training unified architecture for generic perception for zero-shot and few-shot tasks. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022, pp. 16783–16794. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01630. URL https://doi.org/10.1109/CVPR52688.2022.01630.

A. Experimentin Setups

A.1. Experimental Setting for VL Tasks

Referring Expression Comprehension   We report the standard metric ACC@0.5 on the validation and test sets. Epochs are set to 100, dropout is set to 0.1, warmup rate is set to 0.06, and label smoothing rate is set to 0.1. For prompt tuning, we follow the experimental setting used in Yang et al. (2022), where the batch size is set to 128, the learning rate is set to 0.03, and the prompt length is set to 100. For LoRA, the batch size is set to 256 for the base, and 1024 for the large, the learning rate is set to 1e-4, and the rank is set to 16. For adapters, the batch size is set to 1024, the learning rate is set to 1e-4, and the bottleneck dimension is set to 128.

Visual Entailment   We report accuracy on both dev and test sets. Dropout is set to 0.1, warmup rate is set to 0.06, and label smoothing rate is set to 0.1. For prompt tuning, we follow the experimental settings in Yang et al. (2022), where the batch size is set to 128, the learning rate is set to 0.03, epochs are set to 100 and the prompt length is set to 64. For LoRA, the batch size is set to 256 for the base, and 512 for the large, epochs are 100, the learning rate is set to 1e-4, and the rank is set to 16. For adapters, the batch size is set to 512, epochs are set to 10, the learning rate is set to 1e-4, and the bottleneck dimension is set to 128.

Image Captioning   We report BLEU@4, METEOR, CIDEr, and SPICE scores on the Karpathy test split. Dropout is set to 0.1, warmup rate is set to 0.06, and label smoothing rate is set to 0.1. For prompt tuning, we follow the experimental setting used in Yang et al. (2022), where the batch size is set to 256, the learning rate is set to 0.03, epochs are set to 100 and the prompt length is set to 64. For LoRA, the batch size is set to 128, epochs are 100, the learning rate is set to 1e-4, and the rank is set to 16. For adapters, the batch size is set to 512 for base and 128 for large, epochs are set to 10, the learning rate is set to 1e-4, and the bottleneck dimension is set to 128.

Visual Question Answering   We conduct experiments on VQA 2.0 and report the score on the test-dev and test-std set. Dropout is set to 0.1, warmup rate is set to 0.04, Exponential Moving Average(EMA) with a decay rate is set to 0.9999, and label smoothing rate is set to 0.1. For prompt tuning, we follow the experimental setting used in Yang et al. (2022), where the batch size is set to 256, the learning rate is set to 0.03, epochs are set to 100 and the prompt length is set to 10. For LoRA, the batch size is set to 128, epochs are 100, the learning rate is set to 1e-4, and the rank is set to 16. For adapters, the batch size is set to 256, epochs are set to 30 for base and 15 for large, and the bottleneck dimension is set to 128. After finetuning, we use beam search to generate our answer.

After we retrieve auxiliary experts, we apply the same optimization step as the initial experts to train the interpolation of experts. All of the experiments are conducted in A100 40G and V100 32G.

A.2. Experimental Setting for Vision Tasks

We select 8 common vision tasks from COOP (Zhou et al., 2022a), following their splits. For few-shot image classification each adapter for 200 epochs, with a batch size of 64 and a learning rate of 1e-4. The ratio for label smoothing is 0.1. We follow the data augmentation used in Wang et al. (2022), where the same random resize cropping, random flipping, RandAug, and random erasing transformations are conducted. We use Mixup and CutMix with an overall 0.5 probability to be performed for each batch and alpha is set to 0.8 and 1.0.

A.3. Experimental Setting for Language Tasks

We select 6 language understanding tasks from GLUE benchmark (Wang et al., 2018) as Wang et al. (2022), including both single-sentence classification tasks and sentence-pair classification tasks. We reuse the instructions of each task used in Wang et al. (2022) in our experiments. For the hyper-parameters of adapters, we tune the training epochs among $\{5, 7, 10\}$, learning rate among $\{0.03, 1e-4, 5e-5\}$, batch size among $\{32, 64, 128\}$. We report the best performance on the development set for each task following Wang et al. (2022).

B. Analysis For Relationship Between FIM and Landscape

Here we will give a detailed analysis of the relationship between FIM and local optimums. We begin this by restating and adding to the notation used in Sec. 2.1.

B.1. Notation and preliminaries

Let θ_0 be the pre-trained weights and φ_0 be the initial extra added parameters. For a model with pre-trained weights θ_0, extra added parameters φ and input vector x, we let $P_\varphi(y|x)$ denote the model's output distribution. The two loss functions of tasks τ_i $(i = 1, 2)$ can be formulated as $L = \mathbb{E}_{(x,y) \sim P_i}[-\log P_\varphi(y|x)]$, where P_i is the corresponding empirical distribution of task τ_i. The FIM F_φ^i of each task τ_i can be seen as the negative of the Hessian matrix of loss, which is formulated as $F_\varphi^i = -\nabla_\varphi^2 L_\varphi^i = \mathbb{E}_{x,y \sim P_i}[\nabla_\varphi \log P_\varphi(y|x) \nabla_\varphi \log P_\varphi(y|x)^T]$. Let φ_1 and φ_2 be the corresponding local minimum for τ_1 and τ_2, optimized from the same initial parameters φ_0, respectively.

B.2. Restatement of Assumption

Before we start our proof of Theorem 2.1, we introduce the assumptions mentioned in Section 2.2 formally.

Assumption B.1. The gradient of two loss functions are similar at the initial point φ_0, *i.e.*, there is a constant C_1 such that $||\nabla_\varphi L_1(\varphi_0) - \nabla_\varphi L_2(\varphi_0)||_2 \le C_1$.

Assumption B.2. The Fisher information matrix in the linear line from φ_0 to φ_1 for two tasks are similar, *i.e.*, there is a constant C_2 for φ along the line to have $||F_\varphi^1 - F_\varphi^2||_2 \le C_2$.

Assumption B.3. The Fisher information matrix for task τ_2 in the interpolation line is similar to which at φ_1, *i.e.*, there is a constant $0 < C_3 < 1$ for φ along the interpolation line to have the following property

$$||F_\varphi^2 - F_{\varphi_1}^2||_1 \le \frac{1 - C_3}{n_0 ||[F_{\varphi_1}^2]^{-1}||_2} = \frac{1 - C_3}{n} \sqrt{\lambda_{\min}([F_{\varphi_1}^2]^t F_{\varphi_1}^2)},$$

where n_0 is the dimension of extra added parameters, $\lambda_{\min}$ means to get the smallest one of eigenvalues.

Assumption B.1 and Assumption B.2 are mild as Assumption B.1 only requires the norm of gradient difference of two tasks in the initial parameters φ_0 can be bounded and B.2 is the formal restatement of our requirement of two similar tasks. Assumption B.3 requires FIM in the interpolation line varies little for avoiding the Fisher information matrix degenerating since when $C_3 > 1$ the FIM in the interpolation line can degenerate. It corresponds to the non-singular requirement in Theorem 2.1. We add this condition for the reason that a quick change of FIM in the interpolation line will lead to difficulty in comparing FIM.

B.3. Proof of Theorem 2.1

We now begin our proof of Theorem 2.1:

Proof. Since φ_1 and φ_2 are local minimums for two loss functions, the gradient of loss there will be zero, *i.e.*

$$\nabla_\varphi L_i(\varphi_i) = 0.$$

Let $\varphi_t = \varphi_0 + t(\varphi_1 - \varphi_0)$ and $\hat{\varphi}_t = \varphi_1 + t(\varphi_2 - \varphi_1)$ for $0 \le t \le 1$. $c = ||[F_{\varphi_1}^2]^{-1}||_2$.

By the Newton-Leibniz formula, we have

$$0 = \nabla_\varphi L_1(\varphi_1) - \nabla_\varphi L_2(\varphi_2)$$

$$= \nabla_\varphi L_1(\varphi_0) - \int_0^1 F_{\varphi_t}^1 (\varphi_1 - \varphi_0) dt - \nabla_\varphi L_2(\varphi_0) + \int_0^1 F_{\varphi_t}^2 (\varphi_1 - \varphi_0) dt + \int_0^1 F_{\hat{\varphi}_t}^2 (\varphi_2 - \varphi_1) dt$$

$$= \nabla_\varphi L_1(\varphi_0) - \nabla_\varphi L_2(\varphi_0) - \int_0^1 [F_{\varphi_t}^1 - F_{\varphi_t}^2](\varphi_1 - \varphi_0) dt + \int_0^1 F_{\hat{\varphi}_t}^2 (\varphi_2 - \varphi_1) dt.$$

Moving the last term to the left, we will have the following statement

$$\left[\int_0^1 F_{\hat{\varphi}_t}^2 dt\right](\varphi_2 - \varphi_1)$$

$$= \nabla_\varphi L_2(\varphi_0) - \nabla_\varphi L_1(\varphi_0) + \left[\int_0^1 [F_{\varphi_t}^1 - F_{\varphi_t}^2] dt\right](\varphi_1 - \varphi_0).$$

14

The formula on the left is the integration of FIM, in the interpolation line, multiplying the difference between two local minimums. Thus we will check the difference between the integration and the FIM at φ_1.

$$\left|\left|\int_0^1 F_{\hat{\varphi}_t}^2 dt - F_{\varphi_1}^2\right|\right|_1 = \left|\left|\int_0^1 F_{\hat{\varphi}_t}^2 - F_{\varphi_1}^2 dt\right|\right|_1 \le \int_0^1 \left|\left|F_{\hat{\varphi}_t}^2 - F_{\varphi_1}^2\right|\right|_1 dt \le \int_0^1 \frac{1-C_3}{n_0||[F_{\varphi_1}^2]^{-1}||_2} dt = \frac{1-C_3}{n_0 c}.$$

The last inequality rises from Assumption B.3. Let $H = \int_0^1 F_{\hat{\varphi}_t}^2 dt$, $H_0 = F_{\varphi_1}^2$. Then the statement above can be reformulated as

$$||H - H_0||_1 \le \frac{1-C_3}{n_0||[F_{\varphi_1}^2]^{-1}||_2} = \frac{1-C_3}{n_0 c}.$$

Since $H_0 = F_{\varphi_1}^2$ is symmetric, H_0 can be diagonalized orthogonally. Thus we can assume $H_0 = P\Lambda P^{-1}$, where Λ is diagonal and P is orthogonal. We have

$$\left|\left|P^{-1}HP - \Lambda\right|\right|_1 = \left|\left|P^{-1}(H - H_0)P\right|\right|_1 \le \left|\left|P^{-1}\right|\right|_1 \cdot ||P||_1 \cdot ||H - H_0||_1 ;$$

Notice that for any square matrix M of size n_0, we have $||M||_1 \le \sqrt{n_0}||M||_2$. We immediately obtain

$$\left|\left|P^{-1}\right|\right|_1 \cdot ||P||_1 \cdot ||H - H_0||_1 \le \sqrt{n_0}\left|\left|P^{-1}\right|\right|_2 \cdot \sqrt{n_0}\left|\left|P\right|\right|_2 \cdot ||H - H_0||_1$$
$$= n_0 ||H - H_0||_1 \le \frac{1-C_3}{||H_0^{-1}||_2} = \frac{1-C_3}{||\Lambda^{-1}||_2}.$$

The first equality rises from the properties of orthogonality of P and P^{-1}. According to Gershgorin's circle theorem, the eigenvalues of $P^{-1}HP$ can be covered by n_0 circles with centers of the diagonal elements of Λ and radius of all $\left|\left|P^{-1}HP - \Lambda\right|\right|_1$. Since the diagonal elements of Λ are of norms at least $\frac{1}{||\Lambda^{-1}||_2}$ and the radius is less than $\frac{1-C_3}{||\Lambda^{-1}||_2}$, the eigenvalues of $P^{-1}HP$ are of norms at least $\frac{1}{||\Lambda^{-1}||_2} - \frac{1-C_3}{||\Lambda^{-1}||_2} = \frac{C_3}{||\Lambda^{-1}||_2}$. Therefore we have

$$||H^{-1}||_2 = ||(P^{-1}HP)^{-1}||_2 \le \frac{||\Lambda^{-1}||_2}{C_3} = \frac{||H_0^{-1}||_2}{C_3} = \frac{c}{C_3}.$$

Since we already have

$$H(\varphi_2 - \varphi_1) = \nabla_\varphi L_2(\varphi_0) - \nabla_\varphi L_1(\varphi_0) + \left[\int_0^1 [F_{\varphi_t}^1 - F_{\varphi_t}^2] dt\right](\varphi_1 - \varphi_0),$$

left multiplying the matrix H^{-1} on both sides we will get the following statement

$$\varphi_2 - \varphi_1 = H^{-1}\left[\nabla_\varphi L_2(\varphi_0) - \nabla_\varphi L_1(\varphi_0) + \left[\int_0^1 [F_{\varphi_t}^1 - F_{\varphi_t}^2] dt\right](\varphi_1 - \varphi_0)\right].$$

Thus the distance between two local minimums can be further bounded as

$$||\varphi_2 - \varphi_1||_2 = \left|\left|H^{-1}\left[\nabla_\varphi L_1(\varphi_0) - \nabla_\varphi L_2(\varphi_0) + \left[\int_0^1 [F_{\varphi_t}^1 - F_{\varphi_t}^2] dt\right](\varphi_1 - \varphi_0)\right]\right|\right|_2$$
$$\le \frac{||H_0^{-1}||_2}{C_3}\left[||\nabla_\varphi L_1(\varphi_0) - \nabla_\varphi L_2(\varphi_0)||_2 + \left|\left|\left[\int_0^1 [F_{\varphi_t}^1 - F_{\varphi_t}^2] dt\right](\varphi_1 - \varphi_0)\right|\right|_2\right]$$
$$\le \frac{||H_0^{-1}||_2}{C_3}\left[||\nabla_\varphi L_1(\varphi_0) - \nabla_\varphi L_2(\varphi_0)||_2 + \left[\int_0^1 ||\delta F_{\varphi_t}||_2 dt\right] \cdot ||\varphi_1 - \varphi_0||_2\right] \le \frac{c}{C_3}[C_1 + C_2 R_0], \quad (6)$$

where δF means $F^1 - F^2$. The last inequality rises from Assumption B.1 and B.2. The last term is a constant, which finishes the proof. $\square$

According to (6), we see that the distance between two local minimums can be bounded by the difference of the gradient $\nabla_\varphi L(\varphi_0)$ of two loss functions at φ_0, and the difference between the two Fisher information matrices in the interpolation line.