Supplementary Materials Organization:

A. Limitations and Broader Impact

Broader Impact. As for positive impact, we design a novel mask-tuning method strategy to select a subset of network parameters in a pre-trained model for few-shot visual recognition tasks. The learned mask-tuning method can further boost the transfer capacity of the existing prompt-based and adapter-based methods.

Limitations. As for limitations, our method, as a general method, has not been verified on open-world detection and segmentation tasks due to limited computational resources. We leave this exploration in the future.

B. Preliminaries of CLIP-related Tuning Methods

CLIP [21] mainly consists of two components: text encoder $\boldsymbol{G}_T$ and image encoder $\boldsymbol{G}_I$, which are designed to project image and text into the same feature embedding space. Concretely, the text encoder is built with the transformer for extracting text features. Meanwhile, the image encoder is used to extract image features that have the same channel dimension as the text features. The architecture of the image encoder can be ResNet [9] or ViT [5]. Cosine similarity between text and image features is utilized for alignment in CLIP. The CLIP, benefiting from the 400 million text-image pairs from the web and the multi-modality structure, achieves exceptional zero-shot transfer capacity in the downstream tasks.

To improve the transferability in the various downstream tasks, some parameter-efficient studies based on these V&L models, *e.g.*, adapter [7, 30] or prompt [33, 29], are proposed. Specifically, Zhou *et al*. [33] change the hand-craft text prompt to a task-specific learnable *text prompt*, which can be formulated as "$[\boldsymbol{T}_1][\boldsymbol{T}_2]\cdots[\boldsymbol{T}_l][class]$". Here, the l refers to the length of the learnable text prompt. The text encoder extracts text features $\theta\prime_i$ from the learned text prompt to match the image features, the same as Eq. (2). The learnable text prompt is optimized with cross-entropy classification loss. Zang *et al*. [29] introduce a unified prompt into the text and image branches. The unified prompt is also a set of learnable parameters $\boldsymbol{U} \in \mathcal{R}^{d*l}$, where the d, l denote the dimension and length of the prompt, respectively. Then the unified prompt is refined by a transformer layer and split into two parts to complete the image and text input, which can be formulated as follows:

$$\boldsymbol{U}' = \text{transformer}(\boldsymbol{U}), \tag{8}$$

$$\{\boldsymbol{U}_t, \boldsymbol{U}_v\} = \boldsymbol{U}', \tag{9}$$

where the $\boldsymbol{U}_t, \boldsymbol{U}_v$ denote text prompt and visual prompt, respectively. Then the prompts are combined with text or image to be used as input for CLIP. For the adapter-based method, Zhang *et al*. [30] build an adapter following the image encoder of CLIP. Given S-shot training data, the weights of adapter $\boldsymbol{A}$ are initialized with the few-shot image features $\boldsymbol{F}_I \in \mathcal{R}^{m\times d}$ encoded by the image encoder $\boldsymbol{G}_I$. The ground truth labels of images are converted into a one-hot vector $\boldsymbol{L}_I \in \mathcal{R}^{m\times k}$. The possibility of assigning image $\boldsymbol{x}_j$ to class $\boldsymbol{y}_i$ can be formulated as:

$$p_t(\boldsymbol{y} = i \mid \boldsymbol{x}_j) = \alpha \boldsymbol{A}(\boldsymbol{f}_j)\boldsymbol{L}_I^i + p(\boldsymbol{y} = i \mid \boldsymbol{x}_j), \tag{10}$$

$$\boldsymbol{A}(\boldsymbol{f}_j) = exp(-\beta(1 - \boldsymbol{f}_j \boldsymbol{F}_I^{\mathrm{T}})), \tag{11}$$

where the α and β are hyper-parameters, $\boldsymbol{L}_I^i \in \mathcal{R}^{m\times 1}$ denotes i-th column of $\boldsymbol{L}_I$, which corresponds to class i. The TIP-Adapter performs better when fine-tuning the adapter $\boldsymbol{A}$ with S-shot training. Our method is orthogonal to the most existing parameter-efficient adaption methods (*e.g*., adapter and prompt) and endows them the ability to customization on downstream needs.

C. Method Details

Different from the common tuning methods that adopt the image/text prompts or adapter modules, we design a new type of tuning method, termed mask tuning method, which masks the network parameters under a learnable selection. Specifically, we apply binary masks on CLIP to search a subset of pre-trained parameters relevant to downstream tasks. In this way, to better understand the proposed mask tuning method, let's take a fully connected layer as an example (other layers, such as convolution and attention, are the same operation). Specifically, given a fully connected layer, the input and output of which are $\boldsymbol{x}_{\boldsymbol{\theta}_i} \in \mathcal{R}^{c_{in}}$ and $\boldsymbol{y}_{\boldsymbol{\theta}_i} \in \mathcal{R}^{c_{out}}$, respectively. The c_{in} denotes the channel dimension of input, and the c_{out} refers to the channel dimension of output. The weight matrix of the fully connected layer is $\boldsymbol{\theta}_i = \mathcal{R}^{c_{out}\times c_{in}}$, which can be expanded as:

$$\boldsymbol{\theta}_i = \begin{bmatrix} \theta_{1,1} & \cdots & \theta_{1,c_{in}} \\ \vdots & \ddots & \vdots \\ \theta_{c_{out},1} & \cdots & \theta_{c_{out},c_{in}} \end{bmatrix}. \tag{12}$$

The fully connected layer can be formulated as follows:

$$\boldsymbol{y}_{\boldsymbol{\theta}_i} = \boldsymbol{\theta}_i \cdot \boldsymbol{x}_{\boldsymbol{\theta}_i} + \boldsymbol{b}, \tag{13}$$

where $\boldsymbol{b} \in \mathcal{R}^{c_{out}}$ is the bias vector. For each weight matrix, we employ a learnable matrix $\boldsymbol{M}$ with initializing value π, which has the same shape as the weight matrix θ. We set a hard threshold α to binarized the learnable matrix $\boldsymbol{M}$ as follows:

$$m_{i,j}^{bin} = \begin{cases} 1, & \text{if } m_{i,j} \geq \alpha \\ 0, & \text{if } m_{i,j} < \alpha \end{cases}, \tag{14}$$

where the $m_{i,j}$ denotes the parameter in the i-th row and j-th column of learnable matrix $\boldsymbol{M}$, and $m_{i,j}^{bin}$ denotes the corresponding parameter in binary mask m^{bin}. Then the updated weight matrix θ' is obtained as following:

$$\boldsymbol{\theta}_{i,\mathcal{M}} := \boldsymbol{\theta}_i \odot \boldsymbol{M}^{bin}, \tag{15}$$

where the $\odot$ refers to Hadamard product.

Since previous works [31, 17] have shown that training task-specific bias has not a significant improvement for the downstream tasks, we only apply the binary mask on the weight matrix to lower the computation cost. Thus, the updated fully connected layer can be formulated as $\boldsymbol{y}_{\boldsymbol{\theta}_i} = \boldsymbol{\theta}_{i,\boldsymbol{M}} \cdot \boldsymbol{x}_{\boldsymbol{\theta}_i} + \boldsymbol{b}$. This method can be easily extended to convolutional layers, where we also only apply binary masks on the weight matrix. The binary mask is optimized with the cross-entropy classification loss. Importantly, since the binarized function shown in Eq. (14) is non-differentiable, we use the gradient of m^b as a noisy estimator to update the learnable matrix m, following the previous work [31, 14]. The optimization can be formulated as:

$$m_{i,j} \leftarrow m_{i,j} - \gamma \frac{\partial \mathcal{L}_{ce}}{\partial m_{i,j}^{bin}}, \tag{16}$$

where the γ denotes the learning rate that controls the sparsity of mask, and the $\mathcal{L}_{ce}$ denotes the loss value obtained from the Cross-Entropy (CE) loss function. Then, we analyze the change in mask weight $\boldsymbol{M}$ for each layer after training on the

target dataset with the CE loss, *i.e.*, $\Delta = \sum \gamma * \frac{\partial \mathcal{L}_{ce}}{\partial \boldsymbol{M}}$. Multi-head self-attention (MSHA) layers play an important role in the mask-tuning method. This suggests binary attention mask $\boldsymbol{M}_a^{bin}$ plays a significant role during fine-tuning, and we leverage that in our method. We mathematically rewrite the Eq. (16) formulated as:

$$m_{a;i,j} \leftarrow m_{a;i,j} - \gamma \frac{\partial \mathcal{L}_{ce}}{\partial m_{a;i,j}^{bin}}, \tag{17}$$

where $m_{a;i,j}$ represents the mask value for the index i, j of the mask matrix $\boldsymbol{M}_a$ in the MHSA layer. Then, we calculate the Gradient Retaining Purity $\mathcal{P}$ as follows

$$\mathcal{P} = \frac{1}{2}\left(1 + \frac{\text{sgn}(\nabla \mathcal{L}_{\text{ce}})\,(\nabla \mathcal{L}_{\text{ce}} + \nabla \mathcal{L}_{\text{kl}})}{|\nabla \mathcal{L}_{\text{ce}}| + |\nabla \mathcal{L}_{\text{kl}}|}\right). \tag{18}$$

The optimization can be formulated as:

$$m_{a;i,j} \leftarrow m_{a;i,j} - \gamma * (1 - l + l * \mathcal{I}[\mathcal{P} > U]) * \frac{\partial \mathcal{L}_{\text{ce}}}{\partial m_{a;i,j}^{bin}}, \tag{19}$$

where U is a tensor composed of i.i.d $U(0,1)$ random variables and $l \in [0,1]$ is a leak parameter. $l < 1$ means we allow $\nabla \mathcal{L}_{\text{ce}}$ leak through.

D. Experimental Details

D.1. Statistic of Datasets

We conduct experiments on 11 publicly available image classification datasets following CoOP [33]. The datasets including ImageNet [4], FGVCAircraft [16], StanfordCars [12], Flowers102 [19], Caltech101 [6], DTD [3], EuroSAT [11], Food101 [1], UCF101 [23], OxfordPets [20], and SUN397 [26]. For distribution shift experiments, we use ImageNet as the source dataset, while ImageNetV2 [22] and ImageNet-Sketch [25] are used as the target dataset. We report the detailed statistics of the 13 datasets in Tab. 9.

Table 9. The detailed statistics of datasets used in experiments.

Dataset	Classes	Training size	Testing size	Task
Caltech101 [6]	100	4,128	2,465	Object recognition
DTD [3]	47	2,820	1,692	Texture recognition
EuroSAT [11]	10	13,500	8,100	Satellite image recognition
FGVCAircraft [16]	100	3,334	3,333	Fine-grained aircraft recognition
Flowers102 [19]	102	4,093	2,463	Fine-grained flowers recognition
Food101 [1]	101	50,500	30,300	Fine-grained food recognition
ImageNet [4]	1,000	1.28M	50,000	Object recognition
OxfordPets [20]	37	2,944	3,669	Fine-grained pets recognition
StanfordCars [12]	196	6,509	8,041	Fine-grained car recognition
SUN397 [26]	397	15,880	19,850	Scene recognition
UCF101 [23]	101	7,639	3,783	Action recognition
ImageNetV2 [22]	1,000	-	10,000	Robustness of collocation
ImageNet-Sketch [25]	1,000	-	50,889	Robustness of sketch domain

D.2. More Implementation Details

We use single hand-craft prompt as text input when applying mask tuning, following [21]. Specifically, for ImageNet and SUN397, the text prompt is set to be "a photo of a [class].". For fine-grained classification datasets, a task-relevant sentence is added, *e.g.*, the text prompt is "a photo of a [class], a type of flower." for Flowers102 dataset. For other datasets, the text prompt is set to be a task-related context, *e.g.*, for UCF101, the text prompt is "a photo of a person doing [class]." We adopt Adam optimizer with CosineAnnealingLR schedule for optimization. For ImageNet, the maximum epoch is set to 10, the learning rate is set to 3e-5. For other datasets, the maximum epoch is set to 30, and the learning rate is set to 8e-5. The few-shot classification task provides limited training data for fine-tuning model, which may lead to overfitting. Thus, we fix l in

Eq. (7) to be 1 for 8/4/2/1-shot experiments to enhance the anti-overfitting ability of R-AMT. For 16-shot classification task, we observe AMT surpasses Zero-shot CLIP by 18.23% on average across 11 datasets. The upstream information introduced by KL loss may limit the transfer ability of our method. So we set $l = 0.3$ for 16-shot experiments to allow the gradients from CE loss leak through. However, considering the large amount of testing data may result in relatively large distribution gap between testing and few-shot training data. We fix $l = 1$ for the ImageNet, SUN397, and Food101 datasets in 16-shot experiments. The code of our method is based on CoOP [33]. We conduct experiments on 1 NVIDIA A100 GPU. All reported results are the average of three runs with different seeds. Moreover, since the learned binary masks by R-AMT are constructed by binary values. We treat each binary element as a bit and encode every 8 bits to a byte for storage, which greatly saves the storage space of the binary masks.

E. More Experimental Analysis

In this section, we report the average accuracy over three runs and demonstrate the error bar in figures and tables. "error bar" refers to standard deviation.

E.1. Computation Cost Evaluation

As shown in Tab. 10, we provide the comparison of the training time and inference time of existing SOTA methods (*e.g.*, CoOp, CoCoOP, Tip-Adapter), AMT, and our R-AMT. We report the one-epoch time training on the 16-shot setting of the ImageNet dataset and the number of images processed by the model in 1 second (*i.e.*, Frames Per Second (FPS)). Compared witn the Tip-adapter, AMT reduces the 10.3 FPS inference speed and requires an extra 25.0 FPS training time, which is acceptable given the performance improvement.

Table 10. The training and inference time comparison.

Settings	CoOp [33]	CoCoOP [32]	Tip-Adapter [30]	AMT	R-AMT
Training Time (images/s)	7.14	11.11	**50.00**	25.00	15.87
Inference Time (images/s)	7.45	12.21	51.81	**62.11**	**62.11**

E.2. Text Prompt Ensembling

We utilize the prompt ensembling of 7 templates to construct the text input on ImageNet, following TIP-Adapter [30]. In Tab. 11, we report the accuracy of R-AMT and R-PMT on 16-shot ImageNet. The R-AMT and R-PMT boost Zero-shot CLIP 4.76% and 5.09% in terms of the accuracy, respectively. The R-PMT improves TIP-Adapter by 0.13% performance. It further indicates the effectiveness of mask tuning in fine-tuning CLIP. Moreover, we combine R-AMT and R-PMT with TIP-Adapter on 16-shot ImageNet. The R-AMT+TIP-Adapter and R-PMT+TIP-Adapter both surpass TIP-Adapter. It means the image encoder assembled a learned binary mask extracts more distinctive image features in the downstream classification task.

Table 11. Classification accuracy (%) on 16-shot ImageNet when using prompt ensembling of 7 templates for text prompt.

Methods	Zero-shot CLIP	R-AMT	R-PMT
Accuracy (%)	68.73	73.49 (+4.76)	73.82 (+5.09)
Error Bar	-	±0.10	±0.20
Methods	**TIP-Adapter [30]**	**R-AMT + TIP-Adapter [30]**	**R-PMT + TIP-Adapter [30]**
Accuracy (%)	73.69	74.20 (+0.51)	74.22 (+0.53)
Error Bar	-	±0.22	±0.53

E.3. Analyzing the Differences between Fine-tuning the Entire Network and Tuning the Mask

In Tab. 12, we report the performance of fine-tuning and mask tuning the image encoder of CLIP on 16-shot ImageNet. The "Fine-tuning" denotes fine-tuning the whole image encoder. We observe fine-tuning the entire network results in performance degradation compared to Zero-shot CLIP. Tuning the Mask also demonstrates clear advantages over the linear probe model. It is also clear that the gaps in the extremely low-data regime between fine-tuning the entire network and tuning the mask, suggest that mask tuning is much more effective than learning a linear classifier from scratch or fine-tuning the entire network for few-shot learning.

Table 12. Comparison with Fine-tuning on 16-shot ImageNet.

Methods	Zero-shot CLIP	Fine-Tuning	Linear Probe	AMT	R-AMT
Accuracy	66.73	64.51	56.03	72.60	**73.07**
Error Bar	-	±0.34	±0.16	±0.12	±0.10

E.4. Analyzing the Different Gradient Regularity Methods

We explore the influence of gradient dropout regularity with 16 shots ImageNet in Sec. 4.4. of the manuscript paper. In this section, we provide more analysis of the different gradient regularity methods and mainly discuss the difference between GradSignDrop [2] and our R-AMT. The experimental results are shown in Tab. 13. We regard how to utilize the CE loss and KL loss as multi-task learning, with a key emphasis on balancing the general knowledge imparted by the KL loss and the specific knowledge captured by the CE loss. Given the low-data regime inherent in this setting, it is crucial to prevent overfitting in the CE loss and instead prioritize exploration to acquire specific knowledge while retaining the general knowledge present in the pre-trained weights (*i.e.*, KL loss). Previous multi-task learning used gradient surgery to balance the different tasks, which does not consider the property of a low-data regime. Thus, directly applying the gradient surgery (*i.e.*, AgreeGrad and GradSign) in the low-data regime does not bring performance improvement. Zhu *et al*. [34] try to adapt PCGrad [28] to this task, which brings a slight 0.1% performance improvement but is 0.37% lower than R-AMT. We analyze that all conflict gradients forced to be projected in the vertical direction bring the overconfidence of general knowledge from KL loss. Our gradient dropout regularity does not change the direction of the CE gradient and provides a transformation of the gradient numerical scale, which can better explore the specific knowledge in the few-shot data regime. In addition, R-AMT adds some level of randomness to the gradient guided by the KL divergence, which helps the model generalize better to downstream tasks.

Table 13. **Ablation studies on different gradient regularity strategies.** The proposed gradient dropout regularity can make better use of general knowledge of KL loss while exploring the knowledge of downstream data.

Method	Accuracy	Gain
Zero-shot CLIP	66.73	-
AMT	72.60±*0.12*	-
AMT+KL loss	71.92±*0.06*	-0.68
AMT+GradSign [2]	71.95±*0.08*	-0.65
AMT+AgreeGrad [18]	68.82±*0.09*	-3.78
AMT+ProGrad [34]	72.70±*0.22*	+0.10
R-AMT	**73.07**±*0.10*	+0.47

E.5. Different Pruning-Based Mask Technologies

Table 14. Different parameter-level mask tuning on 16-shot ImageNet.

Methods	Zero-shot CLIP	R-AMT		
		Filter-wise Pruning [10]	Channel-wise Pruning [15]	Parameter Pruning [27]
Accuracy	66.73	68.32	67.70	**73.07**
Error Bar	-	±0.18	±0.27	±0.10

Recently, structured network pruning techniques [13, 24, 8] have been proposed to remove parameters in groups by pruning filters [10], channels [15], or parameters [27]. Inspired by these network pruning works, we adopt different pruning-based mask technologies from the dimension aspect, which are classified by Filter-wise Pruning, Channel-wise Pruning, and Parameter Pruning. Concretely, we adopt the channel-wise pruning method to the mask tuning method that focuses on the pruning of the input channel, while the filter-wise pruning method focuses on the pruning of the output channel. These two prompt learning are with all the details of dependencies reversed. As shown in Tab. 14, Filter-wise Pruning and Channel-wise Pruning bring relatively low gains in accuracy compared to Zero-shot CLIP on 16-shot ImageNet. It likely neglects some important details in the pre-trained model when we just focus on measuring the importance of filter-wise or channel-wise information. Parameter pruning results in the best performance, indicating that selecting more finely-tuned masks can enhance the search for more appropriate knowledge from pre-trained weights.

E.6. Dynamic Mask Tuning

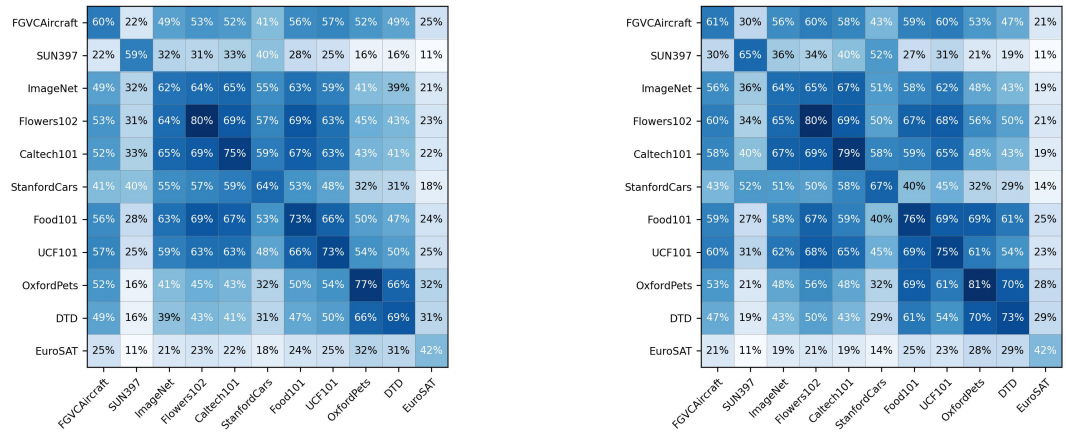
We find the best performance of mask tuning on differernt datasets is achieved when we perform masking on different kinds of layers in Tab. 4. For example, R-AMT surpasses other methods on Caltech101 dataset, while R-MMT reaches the highest accuracy on StanfordCars dataset. R-AMT and R-MMT mean that we apply mask tuning on the MHSA and MLP layers, respectively. Thus, we consider dynamically selecting layers to perform masking. We denote it Dynamic Mask Tuning (R-DMT). Concretely, we aggregate the gradients from CE loss on each layer for one epoch before starting to train the mask. For each element in the leanable mask weight, positive gradient drives the element to be small, as shown in Eq. (17). Once, the value of the element falls below the hard threshold α, the corresponding binary mask becomes 0. Thus, we calculate the mean gradient for each layer and perform masking on the layer with positive mean gradient value. The experimental results are presented in Tab. 15. We observe R-AMT surpasses R-DMT 0.35% on the average across 11 datasets. The gradient of each element is changing during training period. Aggregating gradient before training to decide which layer to applying mask can not well unleash the potential of mask tuning. Thus, we choose performing mask tuning on MHSA layers.

Table 15. Compare dynamically choosing layers with specifying different layers for performing masking on 16-shot datasets.

Method	ImageNet	Caltech101	FGVCAircraft	StanfordCars	Flowers102	OxfordPets	Food101	DTD	EuroSAT	UCF101	SUN397	Average
R-AMT	73.07	**97.00**	58.47	85.93	98.17	93.80	87.47	74.57	**91.80**	86.93	**76.40**	**83.96**
Error Bar	±0.10	±0.37	±0.38	±0.34	±0.09	±0.29	±0.09	±0.56	±0.70	±0.42	±0.03	-
R-MMT	**73.52**	96.77	59.57	**86.43**	98.07	93.83	87.40	**75.73**	84.07	**87.70**	74.23	83.39
Error Bar	±0.15	±0.39	±0.05	±0.09	±0.05	±0.38	±0.16	±0.39	±1.02	±0.16	±0.05	-
R-PMT	73.48	96.63	**60.30**	86.33	**98.27**	93.77	**87.50**	75.60	88.20	87.33	76.12	**83.96**
Error Bar	±0.11	±0.29	±0.82	±0.17	±0.12	±0.25	±0.08	±0.51	±4.69	±0.26	±0.16	-
R-DMT	73.41	96.81	59.70	86.28	97.77	93.41	87.44	75.47	87.88	87.65	73.89	83.61
Error Bar	±0.17	±0.17	±1.00	±0.19	±0.06	±0.43	±0.15	±0.19	±3.47	±0.55	±0.11	-

E.7. Base-to-new Generalization Results

In Tab. 16, we demonstrate the numerical experimental results on each dataset on a 16-shot base-to-new generalization setting. The mask tuning methods (AMT and R-AMT) outperform other methods on 8 out of 11 datasets on the harmonic mean of accuracy on base and new classes. Moreover, we observe the gradient dropout regularity formalism significantly improves the harmonic mean of the accuracy of AMT on fine-grained classification tasks, *e.g.*, StanfordCars, and tasks with a small amount of classes, *e.g.*, EuroSAT. It indicates R-AMT is able to learn more reliable binary masks for fine-grained tasks than AMT. And R-AMT has an anti-overfitting ability, which improves the accuracy of mask tuning when the amount of training classes is limited.



(a) AMT (b) R-AMT

Figure 7. IoU between different binary masks among 11 datasets learned by AMT (a) and R-AMT (b).

E.8. Few-Shot Recognition Accuracy

The full numerical results of Fig. 4 in the main text are presented in Tab. 17. The highest accuracy in each shot setting and dataset are highlighted in red, while the second best is present in orange. The original TIP-Adapter [30] utilizes prompt ensembling to construct text input on ImageNet, which provides better performance than a single prompt on Zero-shot CLIP. Thus, we re-run TIP-Adapter with a single text prompt for a fair comparison. The comparison with TIP-Adapter when using prompt ensembling is presented in Appendix E.2. Overall, R-AMT achieves the best performance on the average of 11 datasets across all shot settings.

F. Visualization

F.1. IoU of Masks among 11 Datasets.

As shown in Fig. 7, we present the IoU of binary masks between two arbitrary datasets on the 16-shot setting. Since we random sample 16 images per class for training three times with different seeds, the binary masks within one dataset are not always the same. This result indicates the knowledge of pre-trained weight is not invariable for downstream classification tasks. We observe that for each dataset the maximum IoU is always itself, which indicates the AMT and R-AMT can find task-specific parameters within CLIP. Moreover, the IoU of binay masks learned by R-AMT within one dataset is higher than AMT. It indicates the R-AMT is able to learn more stable binary masks in different runs.

Table 16. Comparison on the base-to-new generalization setting with CoCoOP [32], ProGrad [34] and CLIP-adapter [7] with 16-shots. H denotes the harmonic mean of the accuracy on base and new classes. All methods are trained on the base classes. We report the average results and standard deviation over three runs for AMT and R-AMT.

	Base	New	H
Zero-shot CLIP	69.34	**74.22**	71.70
CoCoOP	80.47	71.69	75.83
ProGrad	82.79	68.55	75.00
CLIP-adapter	82.62	70.97	76.35
AMT	**86.17**	69.11	76.70
	-	-	-
R-AMT	85.71	72.15	**78.35**
	-	-	-

(a) **Average over 11 datasets**

	Base	New	H
Zero-shot CLIP	27.19	**36.29**	31.09
CoCoOP	33.41	23.71	27.74
ProGrad	42.63	26.97	33.04
CLIP-adapter	39.57	32.27	35.55
AMT	**52.42**	28.11	36.60
	±0.85	±0.75	-
R-AMT	49.22	32.09	**38.85**
	±0.68	±1.11	-

(b) FGVCAircraft

	Base	New	H
Zero-shot CLIP	72.43	68.14	70.22
CoCoOP	75.98	**70.43**	73.10
ProGrad	77.03	68.80	72.68
CLIP-adapter	76.53	66.67	71.26
AMT	**77.23**	70.30	**73.60**
	±0.07	±0.24	-
R-AMT	77.22	70.28	73.59
	±0.17	±0.02	-

(c) ImageNet

	Base	New	H
Zero-shot CLIP	63.37	**74.89**	68.65
CoCoOP	70.49	73.59	72.01
ProGrad	79.00	67.93	73.05
CLIP-adapter	77.13	69.23	72.97
AMT	83.49	62.52	71.50
	±0.44	±0.50	-
R-AMT	**82.90**	69.46	**75.59**
	±0.21	±0.49	-

(d) StanfordCars

	Base	New	H
Zero-shot CLIP	96.84	94.00	95.40
CoCoOP	97.96	93.81	95.84
ProGrad	98.50	91.90	95.09
CLIP-adapter	98.20	93.20	95.63
AMT	**98.88**	**94.61**	**96.70**
	±0.16	±0.27	-
R-AMT	98.88	94.43	96.60
	±0.21	±0.16	-

(e) Caltech101

	Base	New	H
Zero-shot CLIP	70.53	**77.50**	73.85
CoCoOP	82.33	73.45	77.64
ProGrad	83.90	68.50	75.42
CLIP-adapter	85.80	73.63	79.25
AMT	**88.95**	76.22	82.09
	±0.41	±0.55	-
R-AMT	87.87	77.39	**82.30**
	±0.38	±0.67	-

(f) UCF101

	Base	New	H
Zero-shot CLIP	56.48	64.05	60.03
CoCoOP	87.49	60.04	71.21
ProGrad	91.37	56.53	69.85
CLIP-adapter	86.93	**64.20**	**73.86**
AMT	**97.01**	51.61	67.38
	±0.81	±4.06	-
R-AMT	95.79	58.25	72.45
	±1.77	±5.38	-

(g) EuroSAT

	Base	New	H
Zero-shot CLIP	72.08	**77.80**	74.83
CoCoOP	94.87	71.75	81.71
ProGrad	96.27	71.07	81.77
CLIP-adapter	97.70	70.83	82.13
AMT	**98.32**	65.13	78.36
	±0.05	±1.34	-
R-AMT	97.95	70.90	**82.26**
	±0.09	±1.48	-

(h) Flowers102

	Base	New	H
Zero-shot CLIP	90.10	91.22	90.66
CoCoOP	**90.70**	**91.29**	**90.99**
ProGrad	90.17	89.53	89.85
CLIP-adapter	90.40	90.40	90.40
AMT	89.81	90.26	90.03
	±0.08	±0.33	-
R-AMT	90.69	91.14	90.91
	±0.10	±0.24	-

(i) Food101

	Base	New	H
Zero-shot CLIP	69.36	75.35	72.23
CoCoOP	79.74	**76.86**	78.27
ProGrad	80.70	71.03	75.56
CLIP-adapter	81.67	73.93	77.61
AMT	80.99	72.81	76.68
	±0.31	±0.30	-
R-AMT	**82.15**	76.53	**79.24**
	±0.23	±0.25	-

(j) SUN397

	Base	New	H
Zero-shot CLIP	91.17	**97.26**	94.12
CoCoOP	95.20	97.69	**96.43**
ProGrad	94.40	95.10	94.75
CLIP-adapter	94.40	94.10	94.25
AMT	95.53	96.14	95.83
	±0.27	±0.96	-
R-AMT	**95.68**	96.01	95.84
	±0.24	±1.02	-

(k) OxfordPets

	Base	New	H
Zero-shot CLIP	53.24	**59.90**	56.37
CoCoOP	77.01	56.00	64.85
ProGrad	76.70	46.67	58.03
CLIP-adapter	80.47	52.23	63.35
AMT	**85.26**	52.54	65.02
	±0.48	±1.23	-
R-AMT	84.41	57.17	**68.17**
	±0.52	±0.88	-

(l) DTD

Table 17. **Accuracy (%) of few-shot learning, *i.e.*, 16/8/4/2/1-shot, on the 11 datasets.** We report the average accuracy over three runs. "F.A." refers to FGVCAircraft, "S.C." refers to StanfordCars.

shot	Method	F.A.	ImageNet	OxfordPet	Flowers102	EuroSAT	S.C.	Caltech101	UCF101	Food101	SUN397	DTD	Average
-	Zero Shot	24.72	66.73	89.21	71.34	47.60	65.32	92.94	66.75	86.06	62.50	44.39	65.23
16	Linear Prob	36.45	56.03	76.40	94.91	82.67	70.01	90.72	73.72	70.80	67.15	63.42	71.12
16	CoOP	43.29	72.01	91.92	96.93	86.05	82.91	95.47	82.25	84.33	74.58	69.21	79.90
16	TIP-Adapter	45.20	**73.08**	92.66	96.15	88.53	83.04	95.63	84.24	**87.31**	**76.21**	71.57	81.24
16	ProGrad	40.50	72.25	92.76	94.98	84.51	81.48	95.87	81.54	86.76	75.02	65.62	79.21
16	VPT-deep	40.96	70.57	92.91	94.96	91.53	76.13	95.83	82.76	86.18	71.63	69.79	79.39
16	UPT	46.80	72.63	92.95	97.11	90.51	84.33	95.94	84.03	85.00	75.92	70.65	81.44
16	AMT	**59.43**	72.60	**93.43**	**98.07**	**92.00**	**85.70**	**97.10**	**87.00**	85.93	72.27	**74.53**	**83.46**
16	Error Bar	±0.58	±0.12	±0.48	±0.17	±0.75	±0.36	±0.22	±0.62	±0.09	±0.21	±0.25	-
16	R-AMT	**58.47**	**73.07**	**93.80**	**98.17**	**91.80**	**85.93**	**97.00**	**86.93**	**87.47**	**76.40**	**74.57**	**83.96**
16	Error Bar	±0.38	±0.10	±0.29	±0.09	±0.70	±0.34	±0.37	±0.42	±0.09	±0.03	±0.56	-

shot	Method	F.A.	ImageNet	OxfordPet	Flowers102	EuroSAT	S.C.	Caltech101	UCF101	Food101	SUN397	DTD	Average
-	Zero Shot	24.72	66.73	89.21	71.34	47.60	65.32	92.94	66.75	86.06	62.50	44.39	65.23
8	Linear Prob	29.46	49.67	66.36	92.03	77.58	60.90	88.03	69.47	63.99	62.24	57.15	65.17
8	CoOP	39.16	70.68	91.62	94.92	78.71	78.79	94.46	80.02	82.66	71.36	65.01	77.04
8	TIP-Adapter	40.79	**71.42**	91.75	93.94	**83.23**	78.46	95.36	82.03	**86.78**	**73.44**	66.31	78.50
8	ProGrad	37.70	71.06	92.12	93.49	79.29	78.75	94.92	79.64	85.77	72.84	62.35	77.08
8	VPT-deep	36.38	69.83	92.28	91.53	80.75	72.61	95.37	80.16	85.20	69.90	64.06	76.19
8	UPT	39.69	71.60	92.78	95.32	**85.53**	79.95	95.04	80.93	86.14	74.00	65.57	78.78
8	AMT	**47.40**	70.33	**92.47**	**96.47**	82.00	**80.23**	**96.30**	**85.00**	85.07	68.30	**71.30**	**79.53**
8	Error Bar	±0.67	±0.34	±0.19	±0.62	±0.97	±0.12	±0.28	±0.57	±0.17	±0.42	±0.37	-
8	R-AMT	**45.40**	**71.50**	**93.63**	**95.57**	82.53	**80.97**	**96.10**	**84.57**	**87.13**	**73.47**	**70.20**	**80.10**
8	Error Bar	±0.67	±0.28	±0.19	±0.62	±0.97	±0.12	±0.28	±0.57	±0.17	±0.25	±0.37	-

shot	Method	F.A.	ImageNet	OxfordPet	Flowers102	EuroSAT	S.C.	Caltech101	UCF101	Food101	SUN397	DTD	Average
-	Zero Shot	24.72	66.73	89.21	71.34	47.60	65.32	92.94	66.75	86.06	62.50	44.39	65.23
4	Linear Prob	23.70	41.51	56.09	84.84	69.39	48.52	82.95	62.32	55.11	54.61	50.08	57.19
4	CoOP	31.23	68.91	92.23	91.93	72.12	74.50	94.43	76.96	84.35	69.70	59.85	74.20
4	TIP-Adapter	34.90	69.83	91.53	90.74	**77.91**	74.89	94.76	79.14	**86.53**	70.22	61.96	75.67
4	ProGrad	33.70	69.35	92.10	91.19	71.07	75.33	93.99	77.64	84.95	70.70	58.69	74.43
4	VPT-deep	32.99	69.37	**92.40**	85.49	70.87	69.92	94.73	77.14	84.92	68.55	56.08	72.95
4	UPT	33.39	**70.28**	92.10	92.11	75.17	**75.71**	94.09	77.53	85.34	**72.10**	60.87	75.34
4	AMT	**37.80**	69.93	92.03	**93.87**	72.23	75.03	**96.40**	**81.87**	84.73	70.80	**65.47**	**76.38**
4	Error Bar	±0.22	±0.17	±0.45	±0.68	±2.85	±0.58	±0.29	±0.37	±0.25	±0.29	±1.19	-
4	R-AMT	**37.33**	**70.80**	**92.80**	**92.80**	**81.87**	**76.33**	**95.63**	**81.60**	**86.63**	**72.37**	**65.27**	**77.58**
4	Error Bar	±0.19	±0.16	±0.14	±0.37	±1.47	±0.66	±0.19	±0.22	±0.05	±0.37	±1.54	-

shot	Method	F.A.	ImageNet	OxfordPet	Flowers102	EuroSAT	S.C.	Caltech101	UCF101	Food101	SUN397	DTD	Average
-	Zero Shot	24.72	66.73	89.21	71.34	47.60	65.32	92.94	66.75	86.06	62.50	44.39	65.23
2	Linear Prob	17.83	31.51	43.55	73.38	61.74	36.72	78.43	53.54	41.89	44.46	39.46	47.50
2	CoOP	26.85	66.71	90.07	87.63	64.71	70.88	92.70	74.03	84.38	66.98	53.86	70.80
2	TIP-Adapter	**32.78**	68.58	**91.10**	**90.49**	**71.57**	70.07	93.68	76.09	**86.29**	66.79	56.11	**73.05**
2	ProGrad	30.91	66.56	90.45	88.59	66.08	**71.62**	93.09	74.30	84.27	68.28	54.63	71.71
2	VPT-deep	29.36	68.64	90.50	77.60	69.28	68.03	94.70	73.99	84.69	67.55	48.38	70.25
2	UPT	30.00	**69.90**	**92.50**	81.88	68.96	69.44	94.17	74.89	85.02	69.75	52.98	71.77
2	AMT	30.46	69.28	89.34	**88.86**	**70.12**	69.22	**94.48**	**78.46**	84.38	**69.90**	**56.54**	72.82
2	Error Bar	±0.54	±0.11	±0.68	±1.29	±2.84	±0.38	±0.39	±0.76	±0.61	±0.41	±1.86	-
2	R-AMT	**31.72**	**69.92**	90.82	88.41	69.02	**72.46**	**94.61**	**77.75**	**86.26**	**71.00**	**56.32**	**73.48**
2	Error Bar	±0.32	±0.16	±0.45	±1.68	±2.61	±0.30	±0.37	±0.61	±0.25	±0.22	±1.72	-

shot	Method	F.A.	ImageNet	OxfordPet	Flowers102	EuroSAT	S.C.	Caltech101	UCF101	Food101	SUN397	DTD	Average
-	Zero Shot	24.72	66.73	89.21	71.34	47.60	65.32	92.94	66.75	86.06	62.50	44.39	65.23
1	Linear Prob	12.88	22.11	30.04	58.15	50.21	24.61	70.40	41.31	30.13	32.58	29.65	36.55
1	CoOP	21.33	65.82	90.40	78.89	53.62	67.36	93.06	71.50	84.29	67.05	50.91	67.66
1	TIP-Adapter	**29.44**	67.41	**90.79**	**86.26**	63.92	**67.80**	93.34	73.38	**86.13**	64.06	**53.17**	**70.52**
1	ProGrad	27.95	64.40	88.94	**83.63**	55.04	67.08	90.96	71.84	82.68	64.51	**52.74**	68.16
1	VPT-deep	28.23	68.28	90.44	71.95	**66.89**	66.68	93.06	71.03	84.15	66.70	45.38	68.44
1	UPT	28.47	**69.68**	**92.04**	74.67	**66.41**	67.56	93.66	71.93	84.10	68.85	45.09	69.31
1	AMT	28.94	68.98	89.46	83.46	58.80	66.61	**93.75**	**74.31**	83.97	**68.15**	50.71	**69.74**
1	Error Bar	±0.24	±0.20	±0.84	±0.87	±5.27	±0.17	±0.38	±0.49	±0.57	±0.43	±0.96	-
1	R-AMT	**29.47**	**69.35**	89.69	83.14	61.03	**69.30**	**94.15**	**74.08**	**85.12**	**69.13**	51.28	**70.52**
1	Error Bar	±0.18	±0.18	±0.65	±0.51	±1.82	±0.28	±0.50	±0.25	±0.38	±0.22	±1.32	-

References

[1] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Eur. Conf. Comput. Vis.*, pages 446–461, 2014. 3

[2] Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. *Adv. Neural Inform. Process. Syst.*, pages 2039–2050, 2020. 5

[3] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3606–3613, 2014. 3

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 248–255, 2009. 3

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[6] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, pages 178–178, 2004. 3

[7] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021. 1, 8

[8] Xitong Gao, Yiren Zhao, Łukasz Dudziak, Robert Mullins, and Cheng-zhong Xu. Dynamic channel pruning: Feature boosting and suppression. In *Int. Conf. Learn. Represent.*, 2018. 5

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016. 1

[10] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4340–4349, 2019. 5

[11] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pages 2217–2226, 2019. 3

[12] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Int. Conf. Comput. Vis. Worksh.*, June 2013. 3

[13] Lucas Liebenwein, Cenk Baykal, Harry Lang, Dan Feldman, and Daniela Rus. Provable filter pruning for efficient neural networks. In *Int. Conf. Learn. Represent.*, 2019. 5

[14] Tao Lin, Sebastian U Stich, Luis Barba, Daniil Dmitriev, and Martin Jaggi. Dynamic model pruning with feedback. *arXiv preprint arXiv:2006.07253*, 2020. 2

[15] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Int. Conf. Comput. Vis.*, pages 2736–2744, 2017. 5

[16] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 3

[17] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7765–7773, 2018. 2

[18] Lucas Mansilla, Rodrigo Echeveste, Diego H Milone, and Enzo Ferrante. Domain generalization via gradient surgery. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6630–6638, 2021. 5

[19] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729, 2008. 3

[20] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3498–3505, 2012. 3

[21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Int. Conf. Mach. Learn.*, pages 8748–8763, 2021. 1, 3

[22] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *Int. Conf. Mach. Learn.*, pages 5389–5400. PMLR, 2019. 3

[23] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 3

[24] Yang Sui, Miao Yin, Yi Xie, Huy Phan, Saman Aliari Zonouz, and Bo Yuan. Chip: Channel independence-based pruning for compact neural networks. *Adv. Neural Inform. Process. Syst.*, 34, 2021. 5

[25] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Adv. Neural Inform. Process. Syst.*, 2019. 3

[26] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492, 2010. 3

[27] Huanrui Yang, Hongxu Yin, Pavlo Molchanov, Hai Li, and Jan Kautz. Nvit: Vision transformer compression and parameter redistribution. *arXiv preprint arXiv:2110.04869*, 2021. 5

[28] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Adv. Neural Inform. Process. Syst.*, pages 5824–5836, 2020. 5

[29] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Unified vision and language prompt learning. *arXiv preprint arXiv:2210.07225*, 2022. 1

[30] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *Eur. Conf. Comput. Vis.*, pages 493–510, 2022. 1, 2, 4, 7

[31] Mengjie Zhao, Tao Lin, Fei Mi, Martin Jaggi, and Hinrich Schütze. Masking as an efficient alternative to finetuning for pretrained language models. *arXiv preprint arXiv:2004.12406*, 2020. 2

[32] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 16816–16825, 2022. 4, 8

[33] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *Int. J. Comput. Vis.*, pages 2337–2348, 2022. 1, 3, 4

[34] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. *arXiv preprint arXiv:2205.14865*, 2022. 5, 8