MmAP : Multi-Modal Alignment Prompt for Cross-Domain Multi-Task Learning

Yi Xin[1,2*], **Junlong Du**[2*], **Qiang Wang**[2], **Ke Yan**[2†], **Shouhong Ding**[2]

[1]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[2]Youtu Lab, Tencent
xinyi@smail.nju.edu.cn, {jeffdu, albertqwang, kerwinyan, ericshding}@tencent.com

Abstract

Multi-Task Learning (MTL) is designed to train multiple correlated tasks simultaneously, thereby enhancing the performance of individual tasks. Typically, a multi-task network structure consists of a shared backbone and task-specific decoders. However, the complexity of the decoders increases with the number of tasks. To tackle this challenge, we integrate the decoder-free vision-language model CLIP, which exhibits robust zero-shot generalization capability. Recently, parameter-efficient transfer learning methods have been extensively explored with CLIP for adapting to downstream tasks, where prompt tuning showcases strong potential. Nevertheless, these methods solely fine-tune a single modality (text or visual), disrupting the modality structure of CLIP. In this paper, we first propose **M**ulti-**m**odal **A**lignment **P**rompt (**MmAP**) for CLIP, which aligns text and visual modalities during fine-tuning process. Building upon MmAP, we develop an innovative multi-task prompt learning framework. On the one hand, to maximize the complementarity of tasks with high similarity, we utilize a gradient-driven task grouping method that partitions tasks into several disjoint groups and assign a group-shared MmAP to each group. On the other hand, to preserve the unique characteristics of each task, we assign an task-specific MmAP to each task. Comprehensive experiments on two large multi-task learning datasets demonstrate that our method achieves significant performance improvements compared to full fine-tuning while only utilizing approximately $\sim 0.09\%$ of trainable parameters.

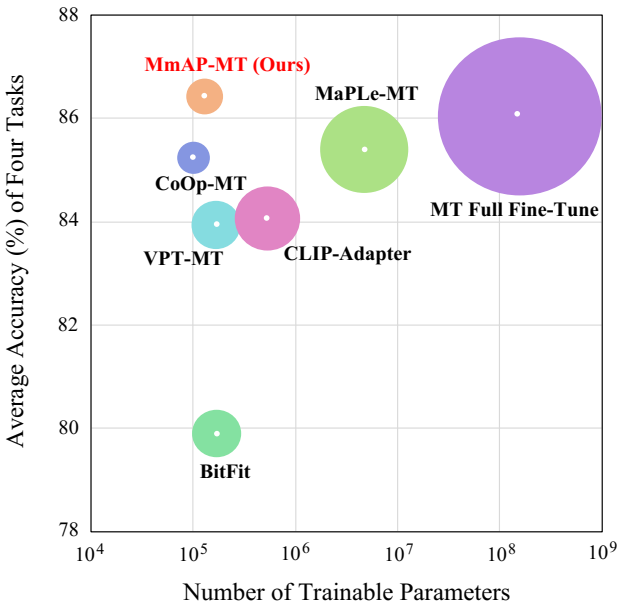


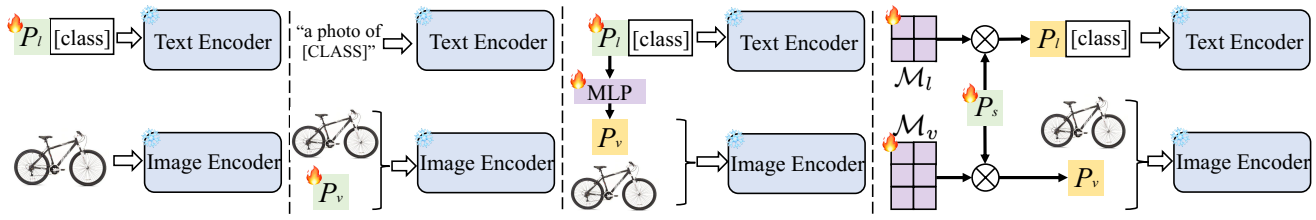
Figure 1: The trade-off between average accuracy over four tasks on Office-Home (Venkateswara et al. 2017) dataset and the number of trainable parameters. The radius of each circle represents the relative amount of trainable parameters.

1 Introduction

Multi-Task Learning (MTL) has surfaced as a potent approach in deep learning that allows for joint training of multiple correlated tasks within a unified network architecture, resulting in enhanced model performance in comparison to Single-Task Learning (STL). The core of MTL lies in learning both the task-shared and the task-specific representations. By capitalizing on shared representations and knowledge across tasks, MTL enhances generalization and mitigates overfitting. Utilizing specific representations allows MTL to preserve the distinct characteristics of each task. Moreover, training a unified model for multiple tasks is generally more parameter-efficient than training several single-task models. Consequently, MTL has garnered considerable interest in various fields, including Computer Vision (Shen et al. 2021; Ye and Xu 2023; Xin et al. 2023), Natural Language Processing (He et al. 2022), etc.

In this work, we mainly focus on vision multi-task learning. Prior research has predominantly concentrated on the design of multi-task model training framework, encompassing encoder-based methods (Gao et al. 2019) and decoder-based methods (Xu, Yang, and Zhang 2023). However, with the growing prowess of vision pre-trained models (e.g., ViT (Dosovitskiy et al. 2021), SwinTransformer (Liu et al. 2021)), directly fine-tuning these models for downstream multi-task leads to substantial performance enhancements

*These authors contributed equally.
†Corresponding author.



(a) Text Prompt - CoOp (b) Visual Prompt - VPT (c) Multi-modal Prompts - MaPLe (d) Multi-modal Alignment Prompt – MmAP (Ours)

Figure 2: Illustrations of (a) text prompt tuning (Zhou et al. 2022), (b) visual prompt tuning (Jia et al. 2022b), (c) multi-modal prompt learning (Khattak et al. 2023) and (d) our multi-modal alignment prompt tuning. 🔥 represents trainable parameters, ❄ represents frozen parameters, $\otimes$ represents Kronecker Product and [class] represents category name.

and has become the mainstream approach for multi-task learning (Liu et al. 2022). In this fine-tuning paradigm, it remains necessary to establish a distinct decoder for each task, with trainable parameters that increase linearly.

To address the above issue, we incorporate the pre-trained vision-language model CLIP (Radford et al. 2021) and consider it tailor-made for vision multi-task learning. On one hand, CLIP is trained to align language and vision modalities using web-scale data (e.g., 400 million text-image pairs), endowing it with a robust capability for zero-shot transfer to vision downstream tasks. On the other hand, the architecture of CLIP offers a distinct advantage. It comprises a text encoder and an image encoder, eliminating the need to establish additional decoder structures for each task. Therefore, we opt for adapting CLIP to address vision multi-task.

Following the conventional pretrain-finetune paradigm, the entire CLIP parameters ($\sim$150M) would require updating, which presents challenges concerning computational and storage expenses. Recently, numerous studies (Zaken, Goldberg, and Ravfogel 2022; Jia et al. 2022b; Gao et al. 2021; Zhou et al. 2022) have introduced parameter-efficient transfer learning techniques to achieve an optimal balance between trainable parameters and performance on downstream tasks. Nonetheless, these existing methods primarily concentrate on pre-trained vision models or language models, with their applicability to more complex vision-language models remaining uncertain. Moreover, these approaches tend to emphasize single-task adaptation, while multi-task adaptation continues to pose a challenge.

To start with, we initially conduct a thorough examination of the performance of existing successful parameter-efficient transfer learning methods when applied to CLIP for vision multi-task learning, as shown in Figure 1. Through our extensive studies, we discover that prompt tuning methods VPT-MT (Jia et al. 2022b), CoOp-MT (Zhou et al. 2022) and MaPLe-MT (Khattak et al. 2023) are more suitable than BitFit (Zaken, Goldberg, and Ravfogel 2022) and Adapter (Gao et al. 2021). This may be attributed to the fact that BitFit and Adapter update model parameters and disrupt the original structural integrity of CLIP. In contrast, prompt tuning methods only modifies input embedding (text or image), as shown in Figure 2. Moreover, we observe that MaPLe-MT outperforms VPT-MT and CoOp-MT, emphasizing the advantages of tuning both modalities simultaneously.

Subsequently, based on our observations, we propose a novel Multi-modal Alignment Prompt (MmAP) for CLIP along with a framework tailored for multi-task image recognition scenarios. Our MmAP generates text prompts and visual prompts through a source prompt to achieve the tuning alignment effect for both modalities. Additionally, we design a multi-task prompt tuning framework based on MmAP. Previous MTL works (Fifty et al. 2021; Standley et al. 2020) have confirmed that training similar tasks together yields a complementary effect, while training dissimilar tasks together results in a negative effect. Therefore, we first employ gradient similarity to group tasks and then assign a group-shared MmAP for joint training. Furthermore, to maintain the independent characteristics of each task, we establish task-specific MmAP for each task individually. We evaluate our method on two large cross-domain multi-task datasets, including Office-Home and MiniDomainNet. Figure 1 displays the results on Office-Home, illustrating that our proposed method has achieved a favorable trade-off between trainable parameters and performance.

Our main contributions are as follows:

- We propose **M**ulti-**m**odal **A**lignment **P**rompt (**MmAP**) for CLIP to favourably align its vision-language representations while parameter-efficient tuning.
- Building upon MmAP, we design a multi-task prompt learning framework for cross-domain image recognition tasks, incorporating both group-shared MmAP and task-specific MmAP.
- We devise a unified library grounded in CLIP to benchmark various parameter-efficient tuning methods for multi-task image recognition. To the best of our knowledge, we are the first to undertake this work.
- Experimental results on two commonly used visual multi-task datasets show that our method achieves competitive performance compared to multi-task full fine-tuning leveraging merely $\sim 0.09\%$ of the CLIP parameters, as shown in Figure 1.

2 Related Work

Multi-Task Learning. Multi-Task Learning (MTL) aims to simultaneously learn multiple tasks by sharing knowledge and computation. There are two classic multi-task in

the field of computer vision. The first is dense scene understanding multi-task, which implements semantic segmentation, surface normal estimation, saliency detection, etc. for each input sample. Current research on multi-task dense scene understanding primarily focuses on decoder structure design (Zhang et al. 2021; Xu, Yang, and Zhang 2023; Liang et al. 2023). The other is cross-domain classification multi-task, and the input data consists of multiple datasets with domain shifts. As multiple domains are involved, current research emphasizes learning shared and private information between domains (Shen et al. 2021; Long et al. 2017).

Vision-Language Model. Foundational vision-language models (e.g., CLIP (Radford et al. 2021) and ALIGN (Jia et al. 2021)) have exhibited remarkable capabilities in various vision tasks. In contrast to models learned with only image supervision, these V-L models encode rich multi-modal representations. Although these pre-trained V-L models learn rich representations, efficiently adapting them to downstream vision tasks remains a challenging problem. Numerous works have demonstrated improved performance on downstream vision tasks by employing tailored methods to adapt V-L models for detection (Li et al. 2022; Zhong et al. 2022), segmentation (Rao et al. 2022; Xu et al. 2022), and recognition (Wortsman et al. 2022). Furthermore, HiPro (Liu et al. 2023) constructs a hierarchical structure to adapt a pre-trained V-L model to various downstream tasks.

Parameter-Efficient Transfer Learning. Parameter Efficient Transfer Learning (PETL) aims to adapt a pre-trained model to new downstream tasks by training only a small number of parameters. Existing PETL methods can be categorized into three groups: parameter tuning, adapter tuning, and prompt tuning. Parameter tuning directly modifies the parameters of a pre-trained model, either by tuning the weights (Hu et al. 2022) or biases (Zaken, Goldberg, and Ravfogel 2022). Adapter tuning inserts trainable bottleneck architectures into a frozen pre-trained model, intending to facilitate learning for downstream tasks, such as AdaptFormer (Chen et al. 2022), VL-Adapter(Sung, Cho, and Bansal 2022), and CLIP-Adapter (Gao et al. 2021). Prompt tuning unifies all downstream tasks into pre-trained tasks via designing a specific template to fully exploit the capabilities of foundation models (Jia et al. 2022a; Khattak et al. 2023; Wang et al. 2023).

3 Method

In this section, we first revisit vision-language models with a focus on CLIP. Subsequently, we introduce our proposed **M**ulti-**m**odal **A**lignment **P**rompt (**MmAP**). Finally, we propose a unified prompt learning framework that incorporates both group-shared MmAP and task-specific MmAP.

3.1 Contrastive Language-Image Pre-training

Image Encoder. In this work, we opt for ViT (Dosovitskiy et al. 2021) as the image encoder to be compatibile with the visual prompt (Jia et al. 2022b). Given an input image $I \in R^{H\times W\times 3}$, the image encoder, consisting of $\mathcal{K}$ transformer layers, splits the image into M fixed-size patches and projects them into patch embeddings $E_0 \in R^{M\times d_v}$. The patch embeddings E_k, accompanied by a learnable class token c_k, are fed into the $(k+1)$-th layer $\mathcal{V}_{k+1}$ of the image encoder, and sequentially processed through the following transformer layers:

$$[c_{k+1}, E_{k+1}] = \mathcal{V}_{k+1}([c_k, E_k]) \quad k = 0, 1, ..., \mathcal{K}-1. \quad (1)$$

To acquire the ultimate image representation x, the class token $c_{\mathcal{K}}$ from the last transformer layer is projected into the V-L latent embedding space via *ImageProj*:

$$x = \text{ImageProj}(c_{\mathcal{K}}). \quad (2)$$

Text Encoder. The text encoder adopts a transformer that contains $\mathcal{K}$ layers to tokenize the input words and project them into word embeddings $W_0 \in R^{N\times d_l}$. The W_k are directly fed into the $(k+1)$-th layer $\mathcal{L}_{k+1}$ of the text encoder:

$$[W_{k+1}] = \mathcal{L}_{k+1}(W_k) \quad k = 0, 1, ..., \mathcal{K}-1. \quad (3)$$

The final text representation z is obtained by projecting the text embeddings associated with the last token of the concluding transformer layer $\mathcal{L}_{\mathcal{K}}$ into V-L latent embedding space via *TextProj*:

$$z = \text{TextProj}(W_{\mathcal{K}}). \quad (4)$$

Zero Shot Prediction. For zero-shot prediction, a carefully designed prompt is introduced into the language branch of CLIP, which serves to reconstruct the textual input by equipping with every class name associated with the downstream tasks (*e.g.*, "a photo of a [CLASS]"). The class with the highest cosine similarity score is then selected as the predicted label $\hat{y}$ for the given image, namely that:

$$p(\hat{y}|x) = \frac{\exp\left(\text{sim}\left(x, z_{\hat{y}}\right)/\tau\right)}{\sum_{i=1}^{C}\exp\left(\text{sim}\left(x, z_i\right)/\tau\right)}, \quad (5)$$

where $sim(,)$ represents the computation of cosine similarity. τ is the temperature coefficient learned by CLIP and C is the total number of classes.

3.2 Multi-modal Alignment Prompt

Prior research has mainly focused on designing prompts for a single modality. For example, VPT investigated visual prompts, while CoOp introduced learnable text prompts. We believe that merely tuning one modality disrupts the text-image matching of CLIP, leading to sub-optimal adaptation for downstream tasks. The most concurrent method MaPLe proposed to use text prompt to generate visual prompt via a MLP with considerable parameters, which exhibits limitations regarding to visual modality and model efficiency.

To address these issues, we propose **M**ulti-**m**odal **A**lignment **P**rompt (**MmAP**) to generate text prompt $P_l \in R^{b\times d_l}$ and image prompt $P_v \in R^{b\times d_v}$ simultaneously. Here, we denote b as the length of prompts, while d_l and d_v indicate the dimension of text and image tokens, respectively. We first initialize a source prompt $P_s \in R^{m\times n}$ and two individual scaling matrices $\mathcal{M}_l \in R^{\frac{b}{m}\times\frac{d_l}{n}}$ and $\mathcal{M}_v \in R^{\frac{b}{m}\times\frac{d_v}{n}}$

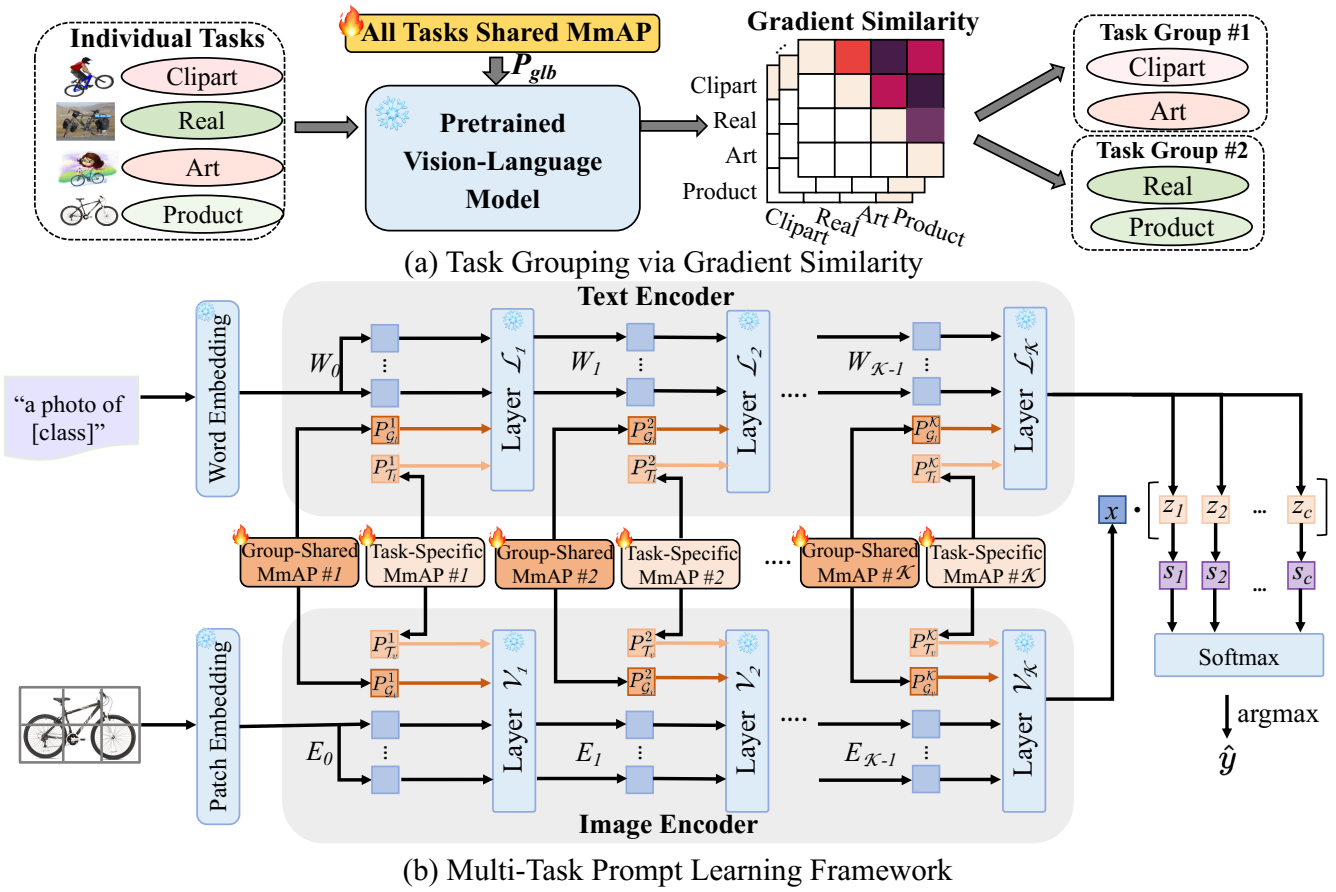


Figure 3: Multi-Task Prompt Learning Framework, including (a) grouping tasks by maximizing the complementarity of tasks with high similarity and (b) employing group-shared and task-specific MmAP for adapting CLIP to downstream tasks.

for two modalities. Then, we apply Kronecker Product to generate prompts for text and image encoders as follows:

$$P_l = \mathcal{M}_l \otimes P_s = \begin{pmatrix} \mathcal{M}_{l_{11}} P_s & \cdots & \mathcal{M}_{l_{1n}} P_s \\ \vdots & \ddots & \vdots \\ \mathcal{M}_{l_{m1}} P_s & \cdots & \mathcal{M}_{l_{mn}} P_s \end{pmatrix}, \quad (6)$$

$$P_v = \mathcal{M}_v \otimes P_s = \begin{pmatrix} \mathcal{M}_{v_{11}} P_s & \cdots & \mathcal{M}_{v_{1n}} P_s \\ \vdots & \ddots & \vdots \\ \mathcal{M}_{v_{m1}} P_s & \cdots & \mathcal{M}_{v_{mn}} P_s \end{pmatrix}. \quad (7)$$

Our proposed MmAP offers two significant advantages. Firstly, the use of the Kronecker Product ensures maximum preservation of the information of source prompt P_s. This facilitates alignment between the text and image prompts. Secondly, the number of learnable parameters is significantly reduced from $\mathcal{K}(d_l + d_v)$ to $mn + \frac{\mathcal{K}(d_l+d_v)}{mn}$, where $\mathcal{K}$ represents the number of transformer layers. This reduction in parameters not only makes the model more efficient but also reduces the risk of overfitting.

3.3 Multi-Task Prompt Learning Framework

In multi-task learning, the joint training of similar tasks can yield mutually beneficial outcomes. Typically, the degree of task similarity can be quantified by evaluating the gradient conflict between tasks. In light of this, we first involves grouping similar tasks together. A shared MmAP is assigned to each group which facilitates the mutual learning and enhancement of tasks within the group. However, to maintain the unique characteristics of each task, we also assign an individual MmAP to each task. This individual MmAP ensures that the distinct features and requirements of each task are adequately catered to. The overall multi-task prompt tuning framework diagram is depicted in Figure 3.

Task Grouping. Existing MTL works (Fifty et al. 2021) have demonstrated that gradient cosine similarity can quantify the similarity of two tasks, i.e., the extent to which two tasks can benefit from joint training. Therefore, we assess the similarity of two tasks by computing gradients on the shared parameters, while keeping the pretrained vision-language model frozen, as shown in Figure 3a.

Specifically, given a global shared MmAP P_{glb} for all tasks, the similarity between the i-th task and the j-th task can be estimated as the following dot product:

$$sim(\mathcal{T}_i, \mathcal{T}_j) = \nabla_{P_{glb}} L_{\mathcal{T}_i}(P_{glb}) \cdot \nabla_{P_{glb}} L_{\mathcal{T}_j}(P_{glb}), \quad (8)$$

where $L_{\mathcal{T}}$ denotes the loss on task $\mathcal{T}$. We posit that when $sim(\mathcal{T}_i, \mathcal{T}_j) > 0$, it indicates that the two tasks exhibit a

mutual gain effect. Moreover, for robust estimation, we average multiple "snapshots" of similarity during the training of the global shared MmAP. At a high level, we concurrently train all tasks, evaluate pairwise task similarity throughout the training process, and identify task groups that maximize the total inter-task similarity.

Multi-Task Prompt Learning. We develop a unified multi-task prompt learning framework upon our proposed MmAP, as depicted in Figure 3b. Given N downstream tasks $\{\mathcal{T}_i\}_{i=1}^{N}$, we first partition them into several disjoint groups according to gradient similarities. For brevity, we denote $\mathcal{G}$ as a task group that consists of $|\mathcal{G}|$ tasks $(1 \leq |\mathcal{G}| \leq N)$. Then we construct *group-shared MmAP* for CLIP that contains $\mathcal{K}$ transformer layers, including source prompts $P_{\mathcal{G}} = \{P_{\mathcal{G}}^k\}_{k=1}^{\mathcal{K}}$, scaling matrices $\mathcal{M}_{\mathcal{G}_l} = \{\mathcal{M}_{\mathcal{G}_l}^k\}_{k=1}^{\mathcal{K}}$ and $\mathcal{M}_{\mathcal{G}_v} = \{\mathcal{M}_{\mathcal{G}_v}^k\}_{k=1}^{\mathcal{K}}$ for language and vision branches, respectively. The *group-shared MmAP* is cumulatively updated by all tasks within group $\mathcal{G}$, achieving complementary benefits across similar tasks. Additionally, for every task in group $\mathcal{G}$, we build *task-specific MmAP* for learning unique task characteristic, including source prompts $P_{\mathcal{T}} = \{P_{\mathcal{T}}^k\}_{k=1}^{\mathcal{K}}$, scaling matrices $\mathcal{M}_{\mathcal{T}_l} = \{\mathcal{M}_{\mathcal{T}_l}^k\}_{k=1}^{\mathcal{K}}$ and $\mathcal{M}_{\mathcal{T}_v} = \{\mathcal{M}_{\mathcal{T}_v}^k\}_{k=1}^{\mathcal{K}}$ for language and vision branches.

During the training of one task $\mathcal{T}$ in group $\mathcal{G}$, we first generate text and image prompts of the k-th layers in two encoders, and then we reconstruct the input tokens by composing the class token, the generated prompts and the text/image tokens from the previous layer. Thereby the calculations of the k-th layers within text and image encoders can be formally represented as:

$$\begin{aligned}[_, _, W_k] &= \mathcal{L}_k\left(\left[P_{\mathcal{G}_l}^k, P_{\mathcal{T}_l}^k, W_{k-1}\right]\right) \\ &= \mathcal{L}_k\left(\left[P_{\mathcal{G}}^k \otimes \mathcal{M}_{\mathcal{G}_l}^k, P_{\mathcal{T}}^k \otimes \mathcal{M}_{\mathcal{T}_l}^k, W_{k-1}\right]\right),\end{aligned} \tag{9}$$

$$\begin{aligned}[c_k, _, _, E_k] &= \mathcal{V}_k\left(\left[c_{k-1}, P_{\mathcal{G}_v}^k, P_{\mathcal{T}_v}^k, E_{k-1}\right]\right) \\ &= \mathcal{V}_k\left(\left[c_{k-1}, P_{\mathcal{G}}^k \otimes \mathcal{M}_{\mathcal{G}_v}^k, P_{\mathcal{T}}^k \otimes \mathcal{M}_{\mathcal{T}_v}^k, E_{k-1}\right]\right).\end{aligned} \tag{10}$$

Here $[\cdot, \cdot]$ refers to the concatenation operation. Finally, *group-shared MmAP* are cumulatively updated by optimizing the following loss:

$$L\left(P_{\mathcal{G}}, \mathcal{M}_{\mathcal{G}_l}, \mathcal{M}_{\mathcal{G}_v}\right) = \sum_{\mathcal{T} \in \mathcal{G}} L_{\mathcal{T}}\left(P_{\mathcal{G}}, \mathcal{M}_{\mathcal{G}_l}, \mathcal{M}_{\mathcal{G}_v}\right), \tag{11}$$

and *task-specific MmAP* are trained via:

$$L\left(P_{\mathcal{T}}, \mathcal{M}_{\mathcal{T}_l}, \mathcal{M}_{\mathcal{T}_v}\right) = L_{\mathcal{T}}\left(P_{\mathcal{T}}, \mathcal{M}_{\mathcal{T}_l}, \mathcal{M}_{\mathcal{T}_v}\right), \tag{12}$$

where $L_{\mathcal{T}}$ is the cross-entropy loss of task $\mathcal{T}$.

4 Experiment

4.1 Benchmark Setting

Datasets. Following prior MTL works (Shen et al. 2021; Long et al. 2017), we consider Office-Home (Venkateswara et al. 2017) and MiniDomainNet (Zhou et al. 2021) datasets to construct our benchmark.

- **Office-Home** contains images from four tasks: Art, Clipart, Product and Real World. Each task covers images from 65 object categories collected under office and home settings. There are about 15,500 images in total.
- **MiniDomainNet** takes a subset of DomainNet, which is an extremely challenging dataset for multi-task learning. MiniDomainNet has 140,000 images distributed among 126 categories. It contains four different tasks: Clipart, Painting, Sketch and Real.

Based on previous research (Shen et al. 2021; Zhou et al. 2022), we randomly select 10% (6-shot per class) and 20% (12-shot per class) samples from Office-Home, and 1% (3-shot per class) and 2% (6-shot per class) samples from MiniDomainNet for training.

Baselines. we compare our method against several tuning baselines, including:

- **Zero-shot** uses hand-crafted text prompt ("a photo of [class]") templates to zero-shot prediction.
- **Single-Task Full Fine-Tuning** updates an individual pretrained model for each task and **Multi-task Full Fine-Tuning** updates an shared pretrained model for all tasks.
- Single-modal prompt tuning methods, including **CoOp** and **VPT**. **CoOp-MT** and **VPT-MT** are the multi-task version, which train a task-shared prompt with samples from all tasks. Additionally, the recent work **MaPLe** serves as one of our baselines. We also construct a multi-task version, referred to as **MaPLe-MT**.
- Other parameter-efficient tuning methods, including **CLIP-Adapter** (Gao et al. 2021), which learns new features on either a visual or a language branch, and **BitFit** (Zaken, Goldberg, and Ravfogel 2022), which tunes the bias parameters of the pre-trained model.

Implementation Details. All experiments are conducted using the PyTorch toolkit on NVIDIA V100 GPU, with CLIP (ViT-B/16) chosen as our default model. To ensure a fair comparison, we maintain consistent hyperparameter settings across all parameter efficient tuning methods. Specifically, we use a batch size of 16/4 and train for 5 epochs for Office-Home/MiniDomainNet. We employ the SGD optimizer with a learning rate of 0.0035.

4.2 Experiment Results

Office-Home. The results are presented in Table 1. Firstly, we observe that our method is on par with Multi-Task Full Fine-Tuning across different data splits (10% or 20%) while requiring only 0.09% (0.13M vs. 149.62M) trainable parameters. This represents a significant breakthrough in parameter-efficient tuning of CLIP for multi-task image recognition. Secondly, our method consistently outperforms other parameter efficient tuning methods. In comparison to prompt methods (*i.e.*, MaPLe-MT, CoOp-MT, and VPT-MT), our method exhibits a significant improvement, highlighting the necessity of integrating visual and text modalities when tuning CLIP and combining the group-shared and the task-specific knowledge.

Regarding the number of trainable parameters, our method ranks second only to CoOp-MT, achieving the best trade-off between accuracy and trainable parameters. Thirdly, we also find that prompt methods outperform CLIP-Adapter and BitFit, indicating that aligning downstream data with CLIP is a more efficient approach.

		Single Task Learning					*Multi Task Learning*						
	Method	Zero	Full FT	CoOp	VPT	MaPLe	Full FT	Adapter	BitFit	CoOp	VPT	MaPLe	**Ours**
10%	Art	82.9	84.9±0.9	84.2±0.3	83.7±0.5	84.4±0.5	**85.8±0.8**	82.3±0.2	79.1±0.1	84.3±0.8	84.0±0.3	84.8±0.6	85.7±0.5
	Clipart	68.3	75.4±0.3	72.6±1.1	70.5±0.3	72.8±0.6	**76.3±1.0**	71.7±0.2	67.8±1.4	73.0±0.1	72.4±0.6	73.3±0.3	**76.3±0.6**
	Product	89.3	91.6±0.3	92.4±0.2	90.9±0.2	92.2±0.1	92.1±1.3	90.8±0.2	86.7±1.5	92.7±0.2	91.7±0.6	92.7±0.4	**92.9±0.5**
	Real	90.1	89.8±0.8	90.5±0.3	89.2±0.6	90.4±0.4	90.2±1.3	89.2±0.2	85.9±0.5	90.7±0.6	90.6±0.1	90.8±0.3	**90.9±0.9**
	Avg.	82.6	85.4±0.6	84.9±0.5	83.6±0.4	85.0±0.4	86.1±1.1	83.5±0.2	79.9±0.9	85.2±0.4	84.7±0.4	85.4±0.4	**86.5±0.6**
20%	Art	84.6	87.1±1.2	85.6±0.6	85.4±0.6	85.9±0.4	87.4±0.8	83.2±1.1	81.7±0.6	86.0±0.2	85.9±0.3	86.3±0.4	**88.2±0.7**
	Clipart	68.2	77.9±0.1	74.5±0.6	71.4±0.4	74.2±0.5	**78.8±1.0**	75.4±0.1	69.6±1.7	73.9±0.6	72.3±0.6	74.2±0.6	77.1±0.6
	Product	89.5	91.9±1.3	93.0±0.4	91.5±0.6	92.8±0.6	93.0±1.0	91.7±0.9	87.2±1.4	92.9±0.4	92.1±0.2	92.9±0.2	**93.5±0.5**
	Real	90.7	89.8±0.6	91.8±0.3	90.9±0.1	91.8±0.4	91.9±0.4	90.6±0.2	86.7±1.0	92.0±0.3	91.7±0.3	92.0±0.5	**92.4±0.3**
	Avg.	83.3	86.7±0.8	86.2±0.5	84.8±0.4	86.2±0.5	**87.8±0.8**	85.2±0.5	85.5±0.4	86.3±0.4	85.5±0.4	86.4±0.4	**87.8±0.5**
	#Params	-	598.48 M	0.04 M	0.68 M	19.2 M	149.62 M	0.53 M	0.17 M	0.01 M	0.17 M	4.8 M	0.13 M

Table 1: Comparison to various methods on *Office-Home*, using the average accuracy (%) over 3 different seeds.

		Single Task Learning					*Multi Task Learning*						
	Method	Zero	Full FT	CoOp	VPT	MaPLe	Full FT	Adapter	BitFit	CoOp	VPT	MaPLe	**Ours**
1%	Clipart	82.6	82.1±1.5	82.7±0.1	82.3±0.1	82.9±0.2	82.8±0.9	82.6±0.1	78.9±0.4	83.4±0.4	83.0±0.3	83.4±0.4	**83.9±0.3**
	Paint	82.3	81.8±0.7	81.8±0.3	81.7±0.3	82.0±0.2	81.5±0.6	80.4±0.3	74.7±0.4	82.3±0.2	81.9±0.6	82.5±0.4	**83.5±0.2**
	Real	91.2	89.1±0.5	91.9±0.3	91.6±0.2	92.0±0.3	89.1±0.6	90.9±0.2	84.2±0.3	91.3±0.1	90.1±0.2	91.4±0.1	**92.2±0.2**
	Sketch	79.9	77.0±0.7	77.1±0.2	78.5±0.3	78.5±0.4	77.2±1.0	78.3±0.6	72.4±0.5	79.2±0.2	78.6±0.6	79.1±0.2	**79.8±0.7**
	Avg.	84.0	82.5±0.9	83.4±0.2	83.5±0.2	83.9±0.3	82.7±0.8	83.0±0.3	77.6±0.4	84.0±0.3	83.4±0.4	84.1±0.3	**84.9±0.4**
2%	Clipart	82.6	82.2±1.3	83.8±0.1	83.5±0.3	83.8±0.4	82.8±0.9	83.1±0.1	81.5±0.2	84.7±0.3	83.8±0.5	84.5±0.3	**85.7±0.4**
	Paint	82.3	82.1±1.4	82.5±0.2	82.4±0.1	82.7±0.2	82.1±0.7	81.5±0.3	76.8±0.2	83.2±0.2	82.2±0.1	83.6±0.4	**85.0±0.2**
	Real	91.2	89.2±0.8	91.9±0.1	91.5±0.1	91.6±0.2	89.3±0.5	90.6±0.2	85.9±0.1	91.7±0.1	90.5±0.1	91.9±0.2	**92.3±0.1**
	Sketch	80.0	77.4±0.5	79.0±0.5	79.6±0.2	79.9±0.3	77.7±1.0	78.7±0.6	74.8±0.2	80.1±0.4	79.0±0.2	80.5±0.3	**81.5±0.2**
	Avg.	84.0	82.7±1.0	84.3±0.2	84.2±0.2	84.5±0.3	83.0±0.8	83.4±0.3	79.8±0.2	84.9±0.4	83.9±0.3	85.1±0.3	**86.1±0.2**
	#Params	-	598.48 M	0.04 M	0.68 M	19.2 M	149.62 M	0.53 M	0.17 M	0.01 M	0.17 M	4.8 M	0.13 M

Table 2: Comparison to various methods on *MiniDomainNet*, using the average accuracy (%) over 3 different seeds.

MiniDomainNet. The results are shown in Table 2. We can draw consistent conclusions with Office-Home. Our method performs the best and achieves 84.9% on the 1% split and 86.1% on the 2% split. However, we observe that the performance of Full Fine-Tuning is not very satisfactory and is worse than most parameter-efficient tuning methods, which is caused by overfitting. Specifically, the task difficulty of MiniDomainNet is significantly increased compared to Office-Home, and concurrently, the number of training data is limited. Moreover, the BitFit method exhibits the worst performance. It updates few parameters of the CLIP using a small amount of data, which severely impairs the original zero-shot capability of CLIP.

The effects of CoOp-MT, VPT-MT, and MaPLe-MT can only approach zero-shot on the 1% split, but when the training data reaches 2%, CoOp-MT and MaPLe-MT surpass zero-shot by 0.9% and 1.1%, respectively. Therefore, to explore the performance under different training data sizes, we set up related experiments, as detailed in ablation study.

4.3 Ablation Study

In this section, we construct various ablation experiments to further analyze our proposed MmAP and multi-task prompt learning framework. At the same time, we also design related experiments for different downstream data size.

Task Specific	Group Shared		Office-Home	
	Task Group	Random	10%	20%
✓	×	×	85.76	86.97
×	✓	×	86.05	87.29
✓	×	✓	85.80	86.92
✓	✓	×	**86.48**	**87.77**
✓	*All in one group*		86.09	87.36

Table 3: Ablation study of Multi-Task Prompt Learning Framework. "Random" means grouping tasks randomly.

Effectiveness of MmAP. To verify the effectiveness of our proposed MmAP, we set up related ablation experiments. As displayed in Figure 5a, a straightforward approach for multi-modal prompts is to tune the text and visual prompts jointly. Another straightforward solution involves sharing text and visual prompts. However, since the dimensions of the Text Encoder ($d_l = 512$) and Image Encoder ($d_v = 768$) of CLIP are not equal, they cannot be shared directly. There-

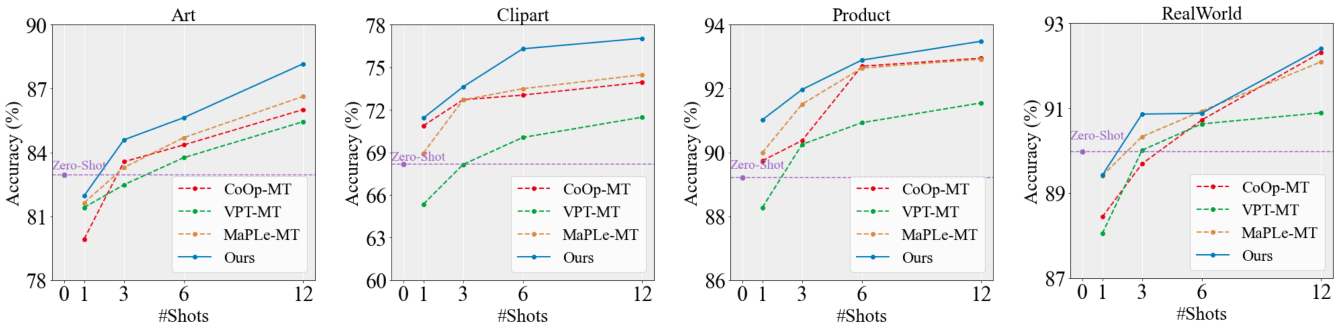


Figure 4: Main results on Office-Home (four tasks) under the k-shot setting. We report the accuracy (%) for 1/3/6/12 shots. Overall, our method attains substantial improvements over zero-shot CLIP and performs favorably against other baselines.

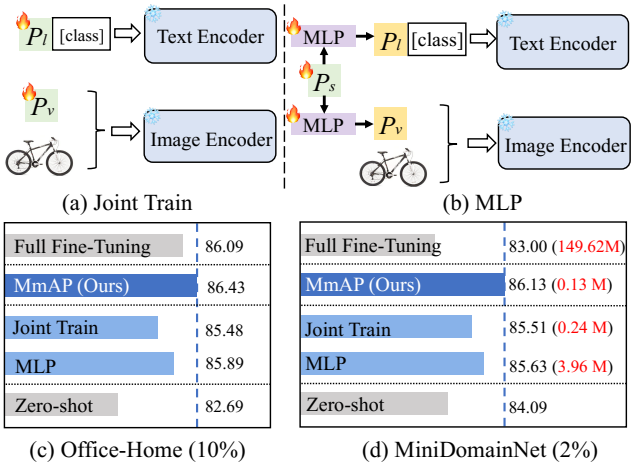


Figure 5: Ablation study of MmAP on Office-Home and miniDomainNet datasets. We construct two baselines: (a) jointly training the text and visual prompts, and (b) utilizing two MLP layers to generate the text and visual prompts.

fore, we design the MLP prompt baseline as another comparison scheme, which employs two MLP layers to generate the text and visual prompts, as shown in 5b.

The results are shown in Figure 5c. Across the four tasks of Office-Home, the MLP baseline exhibits a 0.5% improvement compared to the joint train baseline, demonstrating the effectiveness of establishing a connection between the text prompt and the visual prompt. Additionally, we observe that MmAP achieves a 0.54% improvement compared to the MLP baseline, indicating that the MmAP method is more effective in maximizing information sharing between the text and visual prompts through the Kronecker Product. At the same time, MmAP trainable parameters are greatly reduced relative to the MLP baseline (0.13M vs. 3.96M).

Effectiveness of Multi-Task Prompt Learning Framework. In our multi-task prompt learning framework, task-specific MmAP and group-shared MmAP are the primary components. To verify the importance of each module, we conduct related ablation experiments on Office-Home, and the results are presented in Table 3. To substantiate the effectiveness of task grouping strategy, we incorporate random grouping as a benchmark for comparison. The empirical results elucidate that each module within our framework plays a pivotal role, cumulatively contributing to the superior performance achieved by our multi-task prompt learning framework. Compared to the random grouping, our task grouping performs 0.68% and 0.85% higher under the settings of 10% and 20%, respectively. Compared to the all task in one group, our task grouping performs 0.39% and 0.41% higher under the settings of 10% and 20%. From another perspective, task-specific MmAP surpasses that of CoOp and VPT (results in Table 1), further demonstrating the effectiveness of our MmAP.

Different Downstream Data Size. We examine the impact of training data size on Office-Home (four tasks). We select 1/3/6/12 shots per class and compare our MmAP with CoOp-MT, VPT-MT, and MaPLe-MT. The results for each task and method at different training data scales are presented in Figure 4. The results indicate that our method surpasses all other baselines on the four tasks across data scales, confirming our method's strong generalization. However, we observe that all methods underperform in comparison to Zero-Shot in the 1-shot setting for Art and Real World tasks. This may be due to the fact that 1-shot is too specific to serve as a general representation for the entire task. When provided with 3 or more shots for training, the average performance gap introduced by our method is substantial.

5 Conclusion

In this work, we propose the Multi-modal Alignment Prompt (MmAP) for adapting CLIP to downstream tasks, which achieves the best trade-off between trainable parameters and performance against most of the existing methods. Simultaneously, MmAP addresses the issue of previous single-modal prompt methods (e.g., CoOp and VPT) disrupting CLIP's modal alignment. Building on MmAP, we design a multi-task prompt learning framework, which not only enables similar tasks to be trained together to enhance task complementarity but also preserves the independent characteristics of each task. Our approach achieves significant performance improvements compared to full fine-tuning on two large multi-task learning datasets under limited downstream data while only utilizing $\sim 0.09\%$ trainable parameters.

References

Chen, S.; Ge, C.; Tong, Z.; Wang, J.; Song, Y.; Wang, J.; and Luo, P. 2022. Adaptformer: Adapting vision transformers for scalable visual recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Fifty, C.; Amid, E.; Zhao, Z.; Yu, T.; Anil, R.; and Finn, C. 2021. Efficiently identifying task groupings for multi-task learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Gao, P.; Geng, S.; Zhang, R.; Ma, T.; Fang, R.; Zhang, Y.; Li, H.; and Qiao, Y. 2021. CLIP-Adapter: Better Vision-Language Models with Feature Adapters. In *arXiv preprint arXiv:2110.04544*.

Gao, Y.; Ma, J.; Zhao, M.; Liu, W.; and Yuille, A. L. 2019. Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

He, Y.; Zheng, S.; Tay, Y.; Gupta, J.; Du, Y.; Aribandi, V.; Zhao, Z.; Li, Y.; Chen, Z.; Metzler, D.; et al. 2022. Hyperprompt: Prompt-based task-conditioning of transformers. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Hu, E. J.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Jia, C.; Yang, Y.; Xia, Y.; Chen, Y.-T.; Parekh, Z.; Pham, H.; Le, Q.; Sung, Y.-H.; Li, Z.; and Duerig, T. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Jia, M.; Tang, L.; Chen, B.; Cardie, C.; Belongie, S. J.; Hariharan, B.; and Lim, S. 2022a. Visual prompt tuning. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Jia, M.; Tang, L.; Chen, B.-C.; Cardie, C.; Belongie, S.; Hariharan, B.; and Lim, S.-N. 2022b. Visual Prompt Tuning. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Khattak, M. U.; Rasheed, H. A.; Maaz, M.; Khan, S.; and Khan, F. S. 2023. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, L. H.; Zhang, P.; Zhang, H.; Yang, J.; Li, C.; Zhong, Y.; Wang, L.; Yuan, L.; Zhang, L.; Hwang, J.-N.; et al. 2022. Grounded language-image pre-training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liang, X.; Niu, M.; Han, J.; Xu, H.; Xu, C.; and Liang, X. 2023. Visual Exemplar Driven Task-Prompting for Unified Perception in Autonomous Driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, Y.; Lu, Y.; Liu, H.; An, Y.; Xu, Z.; Yao, Z.; Zhang, B.; Xiong, Z.; and Gui, C. 2023. Hierarchical Prompt Learning for Multi-Task Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, Y.-C.; Ma, C.-Y.; Tian, J.; He, Z.; and Kira, Z. 2022. Polyhistor: Parameter-Efficient Multi-Task Adaptation for Dense Vision Tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Long, M.; Cao, Z.; Wang, J.; and Yu, P. S. 2017. Learning multiple tasks with multilinear relationship networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Rao, Y.; Zhao, W.; Chen, G.; Tang, Y.; Zhu, Z.; Huang, G.; Zhou, J.; and Lu, J. 2022. Denseclip: Language-guided dense prediction with context-aware prompting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shen, J.; Zhen, X.; Worring, M.; and Shao, L. 2021. Variational multi-task learning with gumbel-softmax priors. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Standley, T.; Zamir, A.; Chen, D.; Guibas, L.; Malik, J.; and Savarese, S. 2020. Which tasks should be learned together in multi-task learning? In *Proceedings of the International Conference on Machine Learning (ICML)*.

Sung, Y.; Cho, J.; and Bansal, M. 2022. VL-Adapter: Parameter-Efficient Transfer Learning for Vision-and-Language Tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Venkateswara, H.; Eusebio, J.; Chakraborty, S.; and Panchanathan, S. 2017. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, Q.; Du, J.; Yan, K.; and Ding, S. 2023. Seeing in Flowing: Adapting CLIP for Action Recognition with Motion Prompts Learning. In *Proceedings of the ACM Conference on Multimedia (MM)*.

Wortsman, M.; Ilharco, G.; Kim, J. W.; Li, M.; Kornblith, S.; Roelofs, R.; Lopes, R. G.; Hajishirzi, H.; Farhadi, A.; Namkoong, H.; et al. 2022. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xin, Y.; Du, J.; Wang, Q.; Lin, Z.; and Yan, K. 2023. VMT-Adapter: Parameter-Efficient Transfer Learning for Multi-Task Dense Scene Understanding. In *arXiv preprint arXiv:2312.08733*.

Xu, J.; De Mello, S.; Liu, S.; Byeon, W.; Breuel, T.; Kautz, J.; and Wang, X. 2022. Groupvit: Semantic segmentation emerges from text supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xu, Y.; Yang, Y.; and Zhang, L. 2023. DeMT: Deformable mixer transformer for multi-task learning of dense prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Ye, H.; and Xu, D. 2023. Taskprompter: Spatial-channel multi-task prompting for dense scene understanding. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Zaken, E. B.; Goldberg, Y.; and Ravfogel, S. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Zhang, X.; Zhou, L.; Li, Y.; Cui, Z.; Xie, J.; and Yang, J. 2021. Transfer vision patterns for multi-task pixel learning. In *Proceedings of the ACM Conference on Multimedia (MM)*.

Zhong, Y.; Yang, J.; Zhang, P.; Li, C.; Codella, N.; Li, L. H.; Zhou, L.; Dai, X.; Yuan, L.; Li, Y.; et al. 2022. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022. Learning to prompt for vision-language models. In *International Journal of Computer Vision (IJCV)*.

Zhou, K.; Yang, Y.; Qiao, Y.; and Xiang, T. 2021. Domain adaptive ensemble learning. In *IEEE Transactions on Image Processing (TIP)*.