Hierarchical Prompt Learning for Multi-Task Learning

Yajing Liu[1]*, Yuning Lu[2]*, Hao Liu[1]†, Yaozu An[1], Zhuoran Xu[1], Zhuokun Yao[1],
Baofeng Zhang[1], Zhiwei Xiong[2], Chenguang Gui[1]
[1]JD Logistics [2]University of Science and Technology of China
{liuyajing25,liuhao164,anyaozu,yaozhuokun1,zhangbaofeng13,guichenguang}@jd.com,
lyn0@mail.ustc.edu.cn, zwxiong@ustc.edu.cn

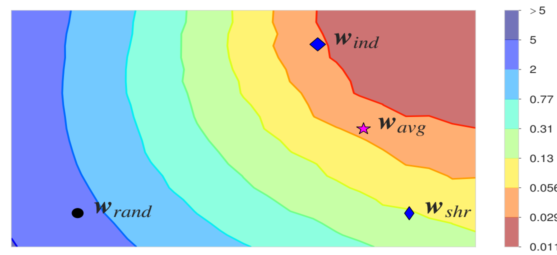
Abstract

Vision-language models (VLMs) can effectively transfer to various vision tasks via prompt learning. Real-world scenarios often require adapting a model to multiple similar yet distinct tasks. Existing methods focus on learning a specific prompt for each task, limiting the ability to exploit potentially shared information from other tasks. Naively training a task-shared prompt using a combination of all tasks ignores fine-grained task correlations. Significant discrepancies across tasks could cause negative transferring. Considering this, we present Hierarchical Prompt (HiPro) learning, a simple and effective method for jointly adapting a pre-trained VLM to multiple downstream tasks. Our method quantifies inter-task affinity and subsequently constructs a hierarchical task tree. Task-shared prompts learned by internal nodes explore the information within the corresponding task group, while task-individual prompts learned by leaf nodes obtain fine-grained information targeted at each task. The combination of hierarchical prompts provides high-quality content of different granularity. We evaluate HiPro on four multi-task learning datasets. The results demonstrate the effectiveness of our method.

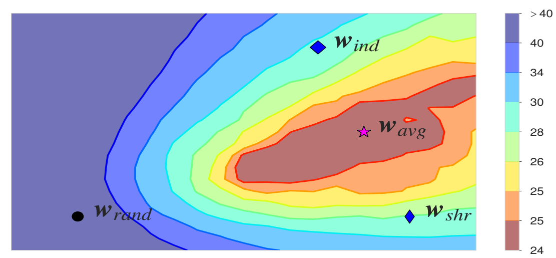
1. Introduction

Vision-language pre-training [23, 34, 49, 71, 74] has recently shown great potential to leverage human language for addressing a wide range of downstream recognition tasks. Vision-language models (VLMs), e.g., CLIP [49] and ALIGN [23], align embeddings of images and texts from massive web data, encouraging the matching image-text pair to be similar and pushing away the unmatched pair [6, 19]. During inference, the task-relevant content in text modality can be provided to query the latent knowledge of the pre-trained VLMs for facilitating visual recognition.

*Equal contribution.
†Corresponding author.



(a) Train loss

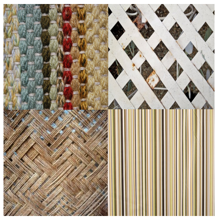


(b) Test error

Figure 1. **Task-shared prompt vs. task-individual prompt on multi-task learning.** We visualize (**a**) train loss and (**b**) test error surface [15] for classifier weights ($\boldsymbol{w}_{rand}$, $\boldsymbol{w}_{ind}$, and $\boldsymbol{w}_{shr}$), which synthesized from the random initialization prompt, task-individual prompt, and task-shared prompt, respectively, on one of the target tasks (i.e., the Art task of the Office-Home dataset [65]). The task-individual prompt is only trained on this task. The task-shared prompt is trained on the combination of all tasks. The average weights ($\boldsymbol{w}_{avg}{=}\frac{1}{2}(\boldsymbol{w}_{shr}{+}\boldsymbol{w}_{ind})$) can perform well to test samples. More details refer to the supplementary materials.

The provided task-relevant texts, often constructed by the *prompt* template and category words, can significantly influence the recognition performance. Prompt *engineering* [23, 49], i.e., manually designing prompts, is a straightforward way to obtain meaningful prompts for adapting VLMs. However, it inevitably introduces artificial bias and relies on time-consuming attempts [49]. Recent advances on prompt *learning* [79, 80] show an alternative way, which

Task: Describable Texture (DTD)



Task-Individual Prompt	Acc (%)
a photo of a {class} texture.	40.54

Task-Shared Prompt	Acc (%)
a photo with repeating {class} pattern.	39.72

Individual Prompt + Shared Prompt	Acc (%)
a photo of a {class} texture. a photo with repeating {class} pattern.	42.73

Task: Satellite Images (EuroSAT)

Task-Individual Prompt	Acc (%)
a centered satellite photo of a {class}.	36.68

Task-Shared Prompt	Acc (%)
a photo with repeating {class} pattern.	36.75

Individual Prompt + Shared Prompt	Acc (%)
a centered satellite photo of a {class}. a photo with repeating {class} pattern.	39.57

Figure 2. **The benefits of prompt learning with multiple tasks.** Note that DTD [9] dataset and EuroSAT [45] dataset employ the same task-shared prompt. Task-individual prompt and task-shared prompt can represent different contents of recognition tasks. Ensembling their zero-shot classifiers can improve performance.

aims to learn the appropriate *soft prompt* in a data-driven manner on each downstream task. With few training data, prompt learning has shown considerable improvement compared with the hand-crafted prompt.

Despite substantial progress, existing approaches [79–81] still focus on adapting VLMs to the individual task. However, challenges in realistic situations demand adapting a model to several similar but different tasks, also known as the problem of multi-task learning [20, 75]. More importantly, current methods learn the specific prompt corresponding to each task, which can not leverage information in other tasks to benefit individual tasks. *Actually, the transferred prompt can be reused for similar tasks.* For example, "a photo of a {class}." is a general prompt for most recognition tasks. Specifically, as shown in Figure 2, for two distinct tasks, i.e., texture images and satellite images, a well-designed prompt can leverage the potential connections across them.

This paper explores how to simultaneously adapt a pre-trained VLM to multiple target tasks through prompting. A straightforward way is to learn the same prompt for all tasks. However, this naïve approach ignores the characteristics of each task and fails to achieve the optimum on each task. Nevertheless, we found that *the task-shared prompt can significantly complement the prompt designed (or learned) individually for each task*. As shown in Figure 2, the task-individual (hand-crafted) prompt captures the fine-grained content of each task. The task-shared (hand-crafted) prompt represents the general content across tasks. The combination of task-shared and task-individual prompts can embrace both general and fine-grained content to enhance recognition.

Another perspective is provided for an in-depth explanation. In Figure 1a, we see that, the classifier weights synthesized from the task-individual prompt (*trained* on the individual target task) have lower training loss than the weights from the task-shared prompt (*trained* on the combination of all tasks). However, the performance of task-individual prompt on the test set is poor (Figure 1b), which implies that the task-individual prompt has the risk of over-fitting. Meanwhile, the task-shared prompt, generalizing on various tasks, can be considered as a regularization to avoid over-fitting. Averaging weights from the task-shared prompt and the task-individual prompt can improve the performance on test data (Figure 1b).

Although similar tasks can facilitate each other by sharing knowledge, we can *not* assume all the offered tasks can benefit from training together. Significant discrepancies across tasks could lead to poor performance, also known as *negative-transfer* [73]. On the other hand, even for the ideal case, i.e., there exists the same beneficial prompt across all tasks, only learning the global (coarse-grained) task-shared prompt neglects the information transferred within some fine-grained task groups.

To address this problem, we present *Hierarchical Prompt* (HiPro) learning to capture multi-grained shared information while mitigating negative transfer between dissimilar tasks. Our HiPro constructs a hierarchical task tree by agglomerative hierarchical clustering based on inter-task affinity. Specifically, the internal node of the tree represents a task group containing a cluster of similar tasks (at descendant leaves). Meanwhile, dissimilar tasks would be divided into different sub-trees, mitigating conflict. For each node, HiPro learns a corresponding prompt to capture the general information of the fine-grained task group. Our HiPro learns not only task-individual prompts (for leaf nodes) but also multi-grained task-share prompts (for non-leaf nodes). For inference, HiPro combines various weights generated from learned prompts, leveraging the information in all tasks to improve the performance of the individual task.

Comprehensive experiments are constructed to validate the effectiveness of our method. HiPro works well on a large-scale multi-task learning benchmark consisting of diverse visual recognition tasks. Compared with the existing prompt learning methods [40,79,80], HiPro has a significant improvement demonstrating the benefit of learning prompts with multiple tasks. Additional visualizations are also provided for analysis.

2. Related Work

Vision-Language Models (VLMs). Foundation models (e.g., GPT-3 [3], PaLM [8], and Florence [74]) trained on massive data show a surprising ability on many applications. In computer vision, milestone works, i.e., CLIP [50] and ALIGN [23], which learn the aligned embedding space of text and images via contrastive learning, demonstrate surprising transferability on downstream tasks. They inspire many researchers to explore better vision-language pre-training [1, 34, 43, 69, 71, 72, 74]. To this day, CLIP, trained on 400 million image-text pairs, is still one of the best VLM released publicly. VLMs also show great potential to address various visual tasks with the language prior, including detection [16, 32, 77], segmentation [31, 53, 68], and recognition [24, 66, 76].

Prompt Learning. Prompt learning is initially proposed for adapting the large pre-trained language models in natural language processing (NLP) [3, 25]. Since various NLP tasks can be unified as the "*text-to-text*" problem [52], the specialized prompt is applied to guide the language model to answer the corresponding question [3, 48, 51]. However, manual crafting of prompts is difficult and often sub-optimal. Recently, automatic prompt generation [18, 26, 30, 33, 57, 78] has emerged as a promising way to adapt language models effectively.

In computer vision, the pioneering work, Context Optimization (CoOp) [80], employs prompt learning to generate an appropriate prompt closer to the task context for improving the recognition of VLMs. Due to its simplicity and effectiveness, many works extend CoOp and apply prompt learning to board vision tasks [5,10,13,24,27,40,63,79,81]. Despite various progressions of existing works, adapting VLMs to multi-task learning with prompting is still an under-explored problem. In addition, although Conditional CoOp [79] also discusses the poor generalization of the task-individual prompt on *unseen classes*, it does not obtain better *in-distribution* generalization, even worse than CoOp. Our HiPro demonstrates that training prompts with data from multiple tasks can effectively improve the in-distribution generalization of prompt learning. The most related work [35] is leveraging prompt learning for multiple perception tasks in autonomous driving scenarios. However, it has many specialized designing for autonomous driving, which is difficult to extend to other multi-task learning settings.

Multi-Task Learning. Multi-task learning (MTL) aims to improve the average performance of multiple target tasks from training together. Common methods design strategies or structures to share information across tasks, including hard sharing [4, 20], soft sharing [11, 42, 70], and learnable sharing [17, 22, 37, 54, 64]. However, training different tasks on a shared model raises the difficulty of optimization and could lead to a negative transfer. Several works attempt to identify the suitable combination of tasks that can benefit from training together, also referred to as *task grouping* [14,60,62,75]. Other popular methods [7,36,38,44,73] aim to improve the optimization dynamics of MTL, e.g., modifying the gradient direction for mitigating conflict [73]. Despite significant progress, the exploration of MTL based on the modern large-scale VLM is still limited, which is an important step for developing the in-the-wild vision system. In addition, our method, guiding the frozen VLM to address various tasks with the lightweight prompt, is an efficient multi-task learner. We also compare HiPro with advanced MTL methods based on their variants of prompt learning. HiPro demonstrates clear improvements compared with MTL baselines.

3. Method

In this section, we introduce our hierarchical prompt (HiPro) learning to effectively adapt a VLM to multiple downstream tasks. Following existing works [79, 80], we use CLIP as the default VLM. Note that our approach can also be applied to other CLIP-like models. We begin with brief reviews of zero-shot CLIP [49], and CoOp [80].

3.1. Prerequisites

Zero-Shot CLIP. Relying on pre-training with web-scale text-image pairs, CLIP [49] learns an aligned feature space of text and image. CLIP consists of an image encoder $f(\cdot)$ and a text encoder $g(\cdot)$. The output vector of encoders is normalized by its L2-norm. Given the pre-defined class names, it can perform zero-shot inference for the test image. Image features of the image $\boldsymbol{x}$ are denoted as $f(\boldsymbol{x})$. Text features of various class descriptions $\{\boldsymbol{t}_i\}_{i=1}^{K}$ can be denoted as $\{g(\boldsymbol{t}_i)\}_{i=1}^{K}$, which are generated by a hand-crafted prompt (e.g., "a photo of a {class}.") and the provided K class names. In this way, the image $\boldsymbol{x}$ can be classified to the i-th class with the largest (cosine) similarity $f(\boldsymbol{x})\cdot g(\boldsymbol{t}_i)$ between their features.

Context Optimization (CoOp). Instead of using the hand-crafted prompt, CoOp [80] aims to learn a soft prompt that is adjusted to the visual context with few training samples. Specifically, let $\boldsymbol{p}$ represents the learnable continuous prompt which is a sequence of tokens. Each token is a vector with the same dimension as the text encoder's input embeddings. The class descriptions $\{\boldsymbol{t}_i(\boldsymbol{p})\}_{i=1}^{K}$ based on the prompt $\boldsymbol{p}$ are construed by combining $\boldsymbol{p}$ and the word embeddings of K class names. Actually, the matrix $[g(\boldsymbol{t}_1(\boldsymbol{p})), .., g(\boldsymbol{t}_K(\boldsymbol{p}))]$ can be considered as the weights of a K-way linear classifier (denoted as $\boldsymbol{w}(\boldsymbol{p})$). Then, CoOp

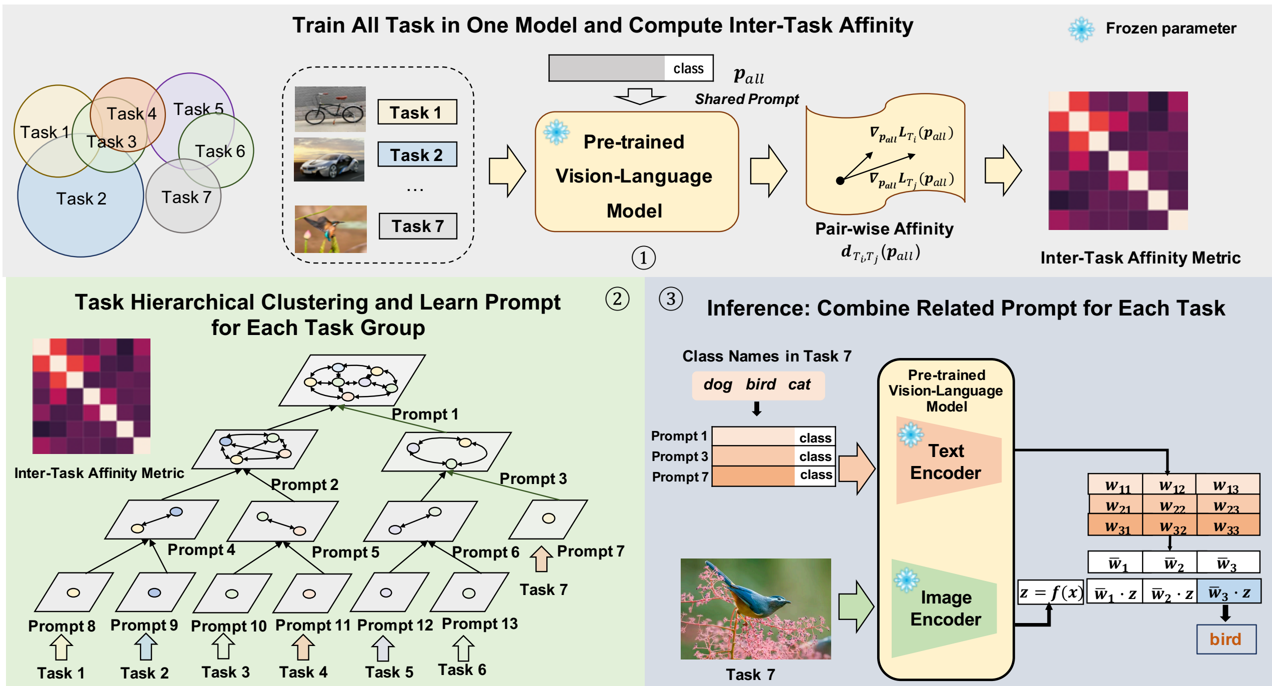


Figure 3. **Hierarchical Prompt Learning.** HiPro first estimates the inter-task affinity based on the gradient direction of the task pair. Given the affinity, a hierarchical task tree is constructed by agglomerative hierarchical clustering. Then, HiPro independently learns a prompt for each node to capture the information on the corresponding task group. At inference time, for each target task, the fusion classifier weights are obtained by averaging the classifier weights generated by the task-related prompts (learned by task groups include the target task).

learns prompt $\boldsymbol{p}$ on the target task $\mathcal{T}$ by the classification loss as follows:

$$\mathcal{L}_{\mathcal{T}}(\boldsymbol{p}) = -\log \frac{\exp(f(\boldsymbol{x}) \cdot g(\boldsymbol{t}_y(\boldsymbol{p}))/\tau)}{\sum_{i=1}^{K} \exp(f(\boldsymbol{x}) \cdot g(\boldsymbol{t}_i(\boldsymbol{p}))/\tau)}, \tag{1}$$

where y and $\boldsymbol{x}$ represent the label and image of the training sample in task $\mathcal{T}$, and τ is a learnable temperature. All parameters of the pre-trained model (i.e., $g(\cdot)$ and $f(\cdot)$) are frozen in training.

3.2. Learning Individual and Shared Knowledge

Our paper aims to jointly learn prompts for various downstream tasks. Only learning the task-individual prompt for each task can not benefit from the shared information across similar tasks (Figure 2). Additionally, the task-individual prompt could be over-fitting to the training data (Figure 1a) with poor generalization on test samples (Figure 1b). Simply training all tasks together for a task-shared prompt ignores the fine-grained knowledge of the individual task and could be under-fitting to each task (Figure 1). Motivated by the above observations, our method learns both task-shared and task-individual prompts, which simultaneously provide general and related content to effectively adapt VLMs.

Given N target tasks $\{\mathcal{T}_i\}_{i=1}^{N}$, $\mathcal{G}$ denotes a task group (i.e., a sub-set of all tasks), which consists of $|\mathcal{G}|$ tasks ($1 \leq |\mathcal{G}| \leq N$). We can extend the prompt learning method (discussed above) to the MTL setting. Let $\boldsymbol{p}_{\mathcal{G}}$ represents the learned prompt for tasks of $\mathcal{G}$. We train $\boldsymbol{p}_{\mathcal{G}}$ by minimizing the following loss:

$$\mathcal{L}(\boldsymbol{p}_{\mathcal{G}}) = \sum_{\mathcal{T}_i \in \mathcal{G}} \mathcal{L}_{\mathcal{T}_i}(\boldsymbol{p}_{\mathcal{G}}), \tag{2}$$

where $\mathcal{L}_{\mathcal{T}_i}$ is the classification loss on the i-th task (Eq. 1).

A straightforward way to achieve our motivation is to simultaneously learn a global task-shared prompt (denoted as $\boldsymbol{p}_{all}$) for all tasks and the task-individual prompt (denoted as $\boldsymbol{p}_j$) for each task (j=1, .., N). Specifically, these N+1 prompts are trained *independently* with their corresponding task group. After training, for j-th target task, we average classifier weights generated from the individual prompt and the shared prompt to obtain a fusion classifier weights $\frac{1}{2}(\boldsymbol{w}(\boldsymbol{p}_{all})+\boldsymbol{w}(\boldsymbol{p}_j))$, which can effectively classify the image features.

This simple method can significantly improve recognition results compared with learning an individual prompt on a single task or learning a shared prompt on all tasks (Table 4). However, a global task-shared prompt can not capture fine-grained knowledge shared within a part of tasks. In addition, significant discrepancies across tasks could lead to

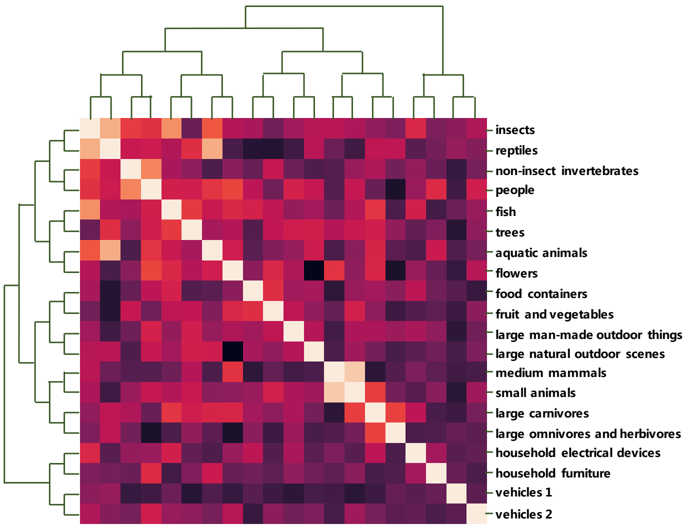


Figure 4. **Inter-task affinity** on CIFAR-100. Tasks with high semantic relevance are clustered together.

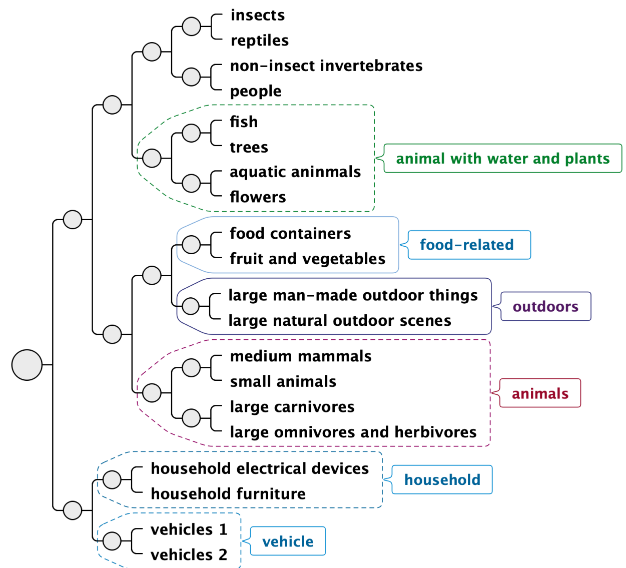


Figure 5. **Hierarchical task tree** of HiPro on CIFAR-100. Different levels of clustering have different granularity, which enables to obtain semantic prompts of different properties.

a degradation in performance. Next, we introduce HiPro, which learns hierarchical prompts for fully exploiting the fine-grained shared information of various task groups.

3.3. Hierarchical Prompt Learning

Overview. The main idea of HiPro is to identify diverse fine-grained groups of similar tasks, allowing each task to benefit from learning with various combinations of other tasks. By fusing multi-grained shared information, HiPro obtains better classifier weights that generalize well on test samples. Specifically, as shown in Figure 3, HiPro first estimates the inter-task affinity based on the gradient direction of the task pairs. Given the affinity, a hierarchical task tree is constructed by agglomerative hierarchical clustering. A node of the task tree represents a task group. Then, HiPro independently learns a prompt for *each* node to capture the transferred information on the corresponding task group. Finally, for each target task, the fusion classifier weights are obtained by averaging the classifier weights generated by the task-related prompts (learned by task groups which include the target task).

Some MTL works [14, 62] also focus on grouping similar tasks and learn a shared network responds to each task group. However, these methods parameterize transferred knowledge by a *neural network*, which would have considerable computational overhead when the number of task groups becomes large. Meanwhile, despite some progress in mode connectivity [15,58], fusing large-scale neural networks to combine the information in multiple task groups is still a challenge in practice. In contrast, HiPro parameterizes various transferred knowledge with prompts and can effectively fuse classifiers on the weight space (Figure 1).

Inter-Task Affinity. The inter-task affinity quantifies the similarity of two tasks, i.e., how much two tasks can benefit from training together. Existing MTL works indicate that the gradient conflict is the crucial reason for performance degradation with joint training [36, 73]. Thus, our methods measure the affinity of two tasks by the similarity of their gradients on shared prompts.

Specifically, given a global task-shared prompt $\boldsymbol{p}_{all}$ for all target tasks, the affinity between the i-th task and the j-th task can be estimated as the following dot product,

$$d_{\mathcal{T}_i,\mathcal{T}_j}(\boldsymbol{p}_{all}) = \nabla_{\boldsymbol{p}_{all}}\mathcal{L}_{\mathcal{T}_i}(\boldsymbol{p}_{all}) \cdot \nabla_{\boldsymbol{p}_{all}}\mathcal{L}_{\mathcal{T}_j}(\boldsymbol{p}_{all}). \qquad (3)$$

In addition, for robust estimation, we average multiple "snapshots" of affinity during training the task-shared prompt, similar to Fifty et al. [14]. Additionally, to reduce sensitivity to prompt initialization, we train multiple task-shared prompts independently and average their affinity estimations. We empirically find that this simple solution without additional forwards is effective in our prompt learning framework. It is no worse than the existing work [14], which estimates affinity by measuring the effect of one task's update on the loss of the other task (Table 4).

Hierarchical Task Clustering and Prompt Learning. Given the inter-task affinity $d_{\mathcal{T}_i,\mathcal{T}_j}$, we can build a hierarchical task tree with agglomerative hierarchical clustering [59] for discovering more fine-grained knowledge shared between some tasks. Specifically, each task is considered an initial cluster. Then, we iteratively find the two most similar clusters and merge them to form a new cluster. To construct a balanced tree, new clusters are temporarily excluded, and clustering continues. All excluded clusters are returned when the remaining clusters are less than 2. This process is repeated until there exists one cluster.

The main challenge of the clustering process is to calculate the affinity (i.e., similarity) between two task groups (clusters) since we only have the pair-wise affinity. Our methods approximate the affinity (denoted as $d(\mathcal{G}_a, \mathcal{G}_b)$) of two task groups ($\mathcal{G}_a$, $\mathcal{G}_b$) by averaging all affinities of their task pairs,

$$d(\mathcal{G}_a, \mathcal{G}_b) = \frac{1}{|\mathcal{G}_a||\mathcal{G}_b|} \sum_{\mathcal{T}_i \in \mathcal{G}_a, \mathcal{T}_j \in \mathcal{G}_b} d_{\mathcal{T}_i, \mathcal{T}_j}, \tag{4}$$

which can also be considered as the average linkage clustering [59]. Each node of the hierarchical task tree represents a task group with potentially shared information. HiPro effectively captures multi-grained shared knowledge from all task groups.

Given M task groups $\{\mathcal{G}_i\}_{i=1}^{M}$ (including groups with a single task), HiPro independently learns M corresponding prompts $\{\boldsymbol{p}_{\mathcal{G}_i}\}_{i=1}^{M}$. Images from all tasks are combined together in a mini-batch to extract features. For each prompt, we minimize Eq. 2 with corresponding image features. Although each prompt is trained independently, the training is still *compact*. Since texts generated by different prompts are fed into the text encoder in a batch, one step can optimize all prompts. Additionally, image features can be reused for different prompts updating without repeating forward.

Combining Task-Related Prompts for Inference. Finally, the averaging classifier weights $\boldsymbol{w}_{\mathcal{T}_j}$ which used to infer test samples of the j-th target task is:

$$\boldsymbol{w}_{\mathcal{T}_j} = \frac{\sum_{i=1}^{M} \boldsymbol{w}(\boldsymbol{p}_{\mathcal{G}_i}) \mathbb{I}(\mathcal{T}_j \in \mathcal{G}_i)}{\sum_{i=1}^{M} \mathbb{I}(\mathcal{T}_j \in \mathcal{G}_i)}, \tag{5}$$

where $\mathbb{I}$ is the indicator function. It also allows the HiPro to have no additional computational overhead for inference.

4. Experiments

We evaluate the performance of HiPro on four multi-task datasets, including Office-Home [65], DomainNet [47], CIFAR-100 [29], and a large-scale multi-task learning benchmark with 10 image classification datasets. We report the average accuracy of each task over 3 runs.

Office-Home [65] contains images collected from four domains (tasks): Art, Clipart, Product, and Real-World. There are 65 shared object categories in different domains. Following MTL works [39, 56], 10% and 20% samples in each task are used for training and the others are used for testing.

DomainNet [47] includes about 0.6 million images distributed among 345 categories. The diversity of categories makes this dataset extremely challenging. It contains six different domains: Clipart, Infograph, Painting, Sketch, Real, and Quickdraw. Following the previous work [56], we use 1% and 2% of labeled data for training.

CIFAR-100 [29] has coarse and fine labels for its images. Each coarse category contains 5 fine-grained classes. Following existing works [54, 55], we treat 20 coarse categories as the 5-way fine-grained classification tasks. 4% and 8% samples of training set are used for training.

Large-Scale MTL Benchmark consists of 10 different downstream tasks, including fine-grained recognition (OxfordPets [46], StanfordCars [28], Flowers102 [45], Food101 [2], and FGVCAircraft [41]), texture recognition (DTD [9]), scene recognition (SUN397 [67]), general recognition (Caltech101 [12]), action recognition (UCF101 [61]), and satellite image recognition (EuroSAT [21]). We construct this experiment to evaluate the performance of our HiPro in real scenarios. Following the splitting of [49, 80], we sample 1, 2, 4, 8, and 16 training samples of each class from downstream tasks for training. Our evaluation metrics are the same as CoOp [80].

Training Details. Following the wildly used setting in prompt learning [79–81], we use CLIP with ResNet-50 vision backbone as our default model. Prompt with 16 tokens are randomly initialized with Gaussian distribution of 0.02 standard deviation [80]. Prompt is trained by the SGD optimizer for 100 epochs with a learning rate of 0.001 and the cosine decay scheduler. Batch size is 20. The checkpoint of the last epoch is used for evaluation. We estimate the inter-task affinity every 5 steps with 8 task-shared prompts.

Comparison methods. We compare HiPro with four prompt learning baselines: (1) **Zero-Shot** CLIP; (2) the standard **CoOp** [80] trained on an individual task; (3) **CoCoOp** [79], which generates a conditional prompt based on the current image; (4) **ProDA** [40] that learns a distribution of diverse prompts. For a fair comparison, we limit the number of learnable prompts of ProDA so that its #parameters are close to HiPro. As the most related method, **CoOp-MTL** is the multi-task version of CoOp, which trains a task-shared prompt with samples from all tasks. We also select several representative methods of MTL and apply them to CoOp-MTL. **PCGrad** [73] projects conflicting gradients to the normal plane for mitigating competing. **IMTL** [38] aims to seek the Pareto point that enables balanced performance across tasks. **TAG** [14] is a similar work to our method, which groups different tasks and trains a shared network for each group.*

*Note that TAG uses a branch-and-bound-like algorithm to select the best combination of tasks, which is an NP-hard problem. On CIFAR-100 (with 20 tasks), it is expected to take many years.

		Single-Task Learning				*Multi-Task Learning*				
	Method	ZeroShot	CoOp	CoCoOp	ProDA	CoOp-MTL	TAG	PCGrad	IMTL	HiPro(Ours)
10%	Art	71.34	71.44±1.26	73.05±0.28	73.38±0.51	71.85±0.69	72.42±0.62	74.15±0.24	74.71±0.72	**75.84±0.50**
	Clipart	51.91	59.57±0.02	57.97±0.47	61.64±0.46	60.44±0.22	59.74±0.65	58.70±0.55	57.43±0.42	**63.28±0.66**
	Real	82.35	83.72±0.55	84.84±0.50	84.29±0.19	82.61±0.39	83.52±0.60	84.47±0.22	85.17±0.12	**85.81±0.14**
	Product	81.75	87.61±0.07	87.44±0.35	88.12±0.21	85.88±0.27	86.06±0.29	86.32±0.26	87.03±0.24	**89.14±0.11**
	Average	71.84	75.59±0.21	75.83±0.25	76.86±0.20	75.20±0.12	75.43±0.67	75.91±0.13	76.08±0.31	**78.52±0.17**
20%	Art	71.42	73.67±0.30	74.94±1.12	75.59±0.40	74.16±0.60	73.36±1.18	75.37±0.66	75.37±0.43	**76.83±0.04**
	Clipart	52.23	63.61±0.65	59.85±0.38	65.71±0.54	62.67±0.38	64.11±0.43	60.98±0.19	59.33+0.20	**67.37±0.41**
	Real	82.59	84.79±0.15	85.98±0.32	86.09±0.35	84.23±0.12	83.79±0.89	85.03±0.18	85.52+0.21	**87.01±0.34**
	Product	81.47	88.62±0.47	88.40±0.19	**89.89±0.23**	86.84±0.27	87.30±0.40	87.08±0.29	88.05+0.11	**90.12±0.12**
	Average	71.93	77.68±0.23	77.29±0.38	79.32±0.17	76.98±0.04	77.14±0.50	77.12±0.23	77.07+0.09	**80.33±0.06**

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

		Single-Task Learning				*Multi-Task Learning*				
	Method	ZeroShot	CoOp	CoCoOp	ProDA	CoOp-MTL	TAG	PCGrad	IMTL	HiPro(Ours)
1%	Clipart	54.82	49.65±0.23	56.48±0.65	56.71±0.32	54.36±0.03	50.94±1.54	57.61±0.12	**58.77±0.22**	**58.79±0.38**
	Real	77.69	76.26±0.30	**79.67±0.10**	78.06±0.11	72.66±0.10	72.40±0.70	77.49±0.27	78.61±0.18	79.06±0.10
	Infograph	40.84	33.23±0.58	43.85±0.23	43.56±0.26	40.68±0.11	34.24±0.41	44.69±0.23	**45.41±0.11**	43.94±0.44
	Sketch	49.24	44.40±0.29	51.46±0.69	50.21±0.27	48.86±0.33	47.11±0.24	52.05±0.17	52.69±0.14	**53.26±0.07**
	Quickdraw	5.95	17.01±0.13	12.01±0.23	15.07±1.35	12.74±0.21	15.23±0.10	11.70±0.23	11.31±0.17	**17.66±0.31**
	Painting	54.59	52.88±0.53	58.73±0.20	58.90±0.34	54.96±0.30	53.40±1.63	58.61±0.17	59.27±0.29	**60.66±0.32**
	Average	47.19	45.57±0.12	50.37±0.15	50.42±0.35	47.38±0.07	45.55±0.43	50.36±0.15	50.98±0.07	**52.23±0.13**
2%	Clipart	54.82	49.59±0.35	56.67±0.29	56.51±0.46	55.34±0.07	51.20±1.01	57.93±0.12	58.79±0.32	**60.06±0.08**
	Real	77.69	76.71±0.30	**80.28±0.11**	78.54±0.05	73.74±0.11	73.48±0.28	77.37±0.28	78.55±0.16	79.77±0.08
	Infograph	40.84	34.37±0.73	**44.91±0.05**	43.20±0.44	40.90±0.41	35.17±0.44	44.19±0.27	**44.90±0.11**	43.61±0.23
	Sketch	49.24	45.72±0.45	52.79±0.12	51.19±0.17	50.31±0.23	49.38±0.33	52.82±0.08	53.42±0.29	**54.67±0.14**
	Quickdraw	5.95	20.38±0.54	13.10±0.23	16.07±0.68	13.75±0.08	17.08±0.39	12.47±0.15	11.60±0.26	**20.43±0.05**
	Painting	54.59	53.07±0.36	59.95±0.21	58.45±0.18	56.22±0.29	54.36±1.62	59.18±0.22	59.66±0.14	**61.57±0.16**
	Average	47.19	46.64±0.24	51.28±0.11	50.66±0.17	48.38±0.03	46.78±0.51	50.66±0.10	51.15±0.17	**53.35±0.10**

Table 2. **Comparison to various methods on DomainNet**, using the average accuracy (%) over 3 runs.

4.1. Visualization

We visualize the inter-task affinity (Figure 4) and the hierarchical task tree (Figure 5) on the CIFAR-100 dataset. Tasks with high semantic relevance are clustered together. Meanwhile, the different levels of clustering have different granularity, which enables to obtain semantic prompts of different properties. More visualizations of clustering results can be found in the supplementary materials.

4.2. Main Results

Office-Home. The results are shown in Table 1. We can see that our HiPro has consistent improvements over different splits compared with other baselines. Compared with CoOp and CoOp-MTL, HiPro shows a large improvement, which indicates the necessity of combining the task-shared and the task-individual knowledge. In addition, our method also outperforms advanced MTL methods. On Office-Home (20%), advanced MTL methods can not outperform the basic MTL baseline CoOp-MTL, even worse than CoOp. However, HiPro shows non-trial improvements compared with CoOp (2.6%) and CoOp-MTL (3.3%).

DomainNet. As shown in Table 2, we can obtain consistent conclusions with Office-Home. Our approach effectively leverages individual and shared information, resulting in large improvements with CoOp and CoOp-MTL. Our HiPro outperforms CoOp by 6.6% on the 1% split and 6.7% on the 2% split, confirming our motivations to use data of multiple tasks for learning prompts. Similarly, our method substantially outperforms CoOp-MTL by 4.8% on the 1% split and 4.9% on the 2% split. Although IMTL and PCGrad are better than HiPro on the Infograph task, they have a significant performance degradation on the Quickdraw dataset. In addition, our method still outperforms them in most tasks, and obtains the best result on average accuracy. CoCoOp outperforms HiPro on Real and Infograph tasks with 2% split. However, it has a longer inference time since the conditional prompt requires to be fed to the text encoder. HiPro is also better than these complex prompt learning methods (i.e., CoCoOp and ProDA) by a large margin in the average accuracy.

CIFAR-100. We provide the detailed results of CIFAR-100 in Table 3. In contrast to DomainNet and OfficeHome,

	Method	*4%*	*8%*
Single-Task Learning	ZeroShot	63.76	63.76
	CoOp	73.60±0.19	76.00±0.10
	CoCoOp	72.56±0.31	75.17±0.21
	ProDA	74.97±0.22	76.66±0.18
Multi-Task Learning	CoOp-MTL	72.77±0.42	75.42±0.11
	PCGrad	73.07±0.16	76.76±0.27
	IMTL	69.28±0.40	71.08±0.13
	HiPro(Ours)	**75.53±0.15**	**77.40±0.09**

Table 3. **Comparison to various methods on CIFAR-100**, using the average accuracy (%) over 3 runs.

different tasks of CIFAR-100 have different concepts. Note that IMTL, which performs well on Office-Home and DomainNet, suffers significant performance degradation on the CIFAR-100 dataset. This phenomenon indicates the sensitivity of IMTL to the dataset. In contrast, our approach still clearly outperforms other comparison method, demonstrating the robustness of our approach.

Large-Scale MTL benchmark The average scores of 10 downstream tasks with various samples are shown in Figure 6. Our HiPro is compared with CoOp, Linear Probe, and Zero-Shot CLIP. Linear Probe CLIP is training a linear classifier with fixed image features. We can see that HiPro significantly and consistently improves the performance of CoOp, which trains on each individual task. It outperforms CoOp by 4.9% on 1-shot. These experiments show that our HiPro, which learns multi-grained task-shared prompts, can capture shared knowledge for effectively adapting VLMs to complex realistic scenarios.

4.3. Ablation Studies

In this section, we construct ablation experiments to further analyze our proposed method HiPro.

Comparison with Shr+Ind. As discussed in Section 3.2, we can learn a global task-sharing prompt to facilitate task-individual prompts for each task. As shown in Table 4, this simple approach can significantly improve performance compared to CoOp-MTL (with only task-shared prompt) or CoOp (with only task-individual prompts), verifying our motivation for simultaneously learning task-shared prompts and task-individual prompts. In addition, our HiPro outperforms Shr+Ind clearly on the DomainNet dataset, which suggests the benefit of learning fine-grained shared prompts. HiPro and Shr+Ind are very closed in Office-Home. The main reason is that OfficeHome has only 4 tasks allowing HiPro to learn 1-2 additional groups.

Comparison with Random Grouping. We compare the task groups obtained from HiPro with the randomly generated task groups. In Table 4, our HiPro outperforms random group tasks, which demonstrates the significance of our hierarchical task clustering.

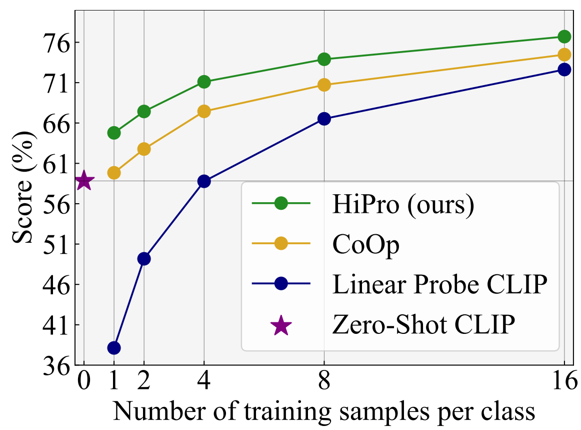


Figure 6. **Average results of 10 image classification tasks.** Comparison with prompt-based methods of leveraging VLM, i.e., hand-crafted prompts (zero-shot CLIP [49]) and prompt tuning (CoOp [80]), and the linear probing. We report the average results on 10 downstream datasets with various training samples.

Method	*DomainNet*		*Office-Home*	
	1%	2%	10%	20%
CoOp	45.57±0.12	46.64±0.24	75.59±0.21	77.68±0.23
CoOp-MTL	47.38±0.07	48.38±0.03	75.20±0.12	76.98±0.04
Shr+Ind	50.72±0.07	51.89±0.04	78.04±0.08	**80.04±0.22**
Rand Group	48.96±3.38	49.42±3.84	77.73±0.18	79.32±0.31
HiPro+TAG	51.88±0.07	**53.14±0.12**	**78.24±0.17**	**80.12±0.30**
HiPro (Ours)	**52.23±0.13**	**53.35±0.10**	**78.52±0.17**	**80.33±0.06**

Table 4. **Ablation Studies** on DomainNet and Office-Home.

Estimating the affinity by TAG [14]. TAG measures the affinity between two tasks by observing the effect of the optimization of one task on the training loss of the other task. In Table 4, We empirically find that, in our prompt learning framework, the simple gradient inner product can have a similar performance to TAG.

5. Conclusion

In this paper, we investigate the limitations of learning specific prompts corresponding to each task or sharing consistent prompts for all tasks, and demonstrate the combination of task-shared and task-individual prompts can significantly improve the results. We propose the hierarchy prompt learning to further explore task relatedness, which hierarchically clusters tasks into a tree structure. Specifically, the task-shared prompt learned by internal nodes of the tree explores information in other tasks to benefit individual tasks, while the task-individual prompt learned by leaf nodes obtains fine-grained representations targeted at each task. The results and visualizations on three datasets demonstrate the effectiveness of our method.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. *arxiv:2204.14198*, 2022. 3

[2] Lukas Bossard, Matthieu Guillaumin, and Luc J. Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, 2014. 6

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *arXiv:2005.14165*, 2020. 3

[4] Rich Caruana. Multitask learning: A knowledge-based source of inductive bias. In *ICML*, 1993. 3

[5] Haoran Chen, Zuxuan Wu, and Yu-Gang Jiang. Multi-prompt alignment for multi-source unsupervised domain adaptation. 2022. 3

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1

[7] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, 2018. 3

[8] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek B Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Benton C. Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier García, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Díaz, Orhan Firat, Michele Catasta, Jason Wei, Kathleen S. Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *arXiv:2204.02311*, 2022. 3

[9] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014. 2, 6

[10] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guo Chun Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *CVPR*, 2022. 3

[11] Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *ACL*, 2015. 3

[12] 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. *CVIU*, 2007. 6

[13] Chengjian Feng, Yujie Zhong, Zequn Jie, Xiangxiang Chu, Haibing Ren, Xiaolin Wei, Weidi Xie, and Lin Ma. Promptdet: Towards open-vocabulary detection using uncurated images. In *ECCV*, 2022. 3

[14] Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. In *NeurIPS*, 2021. 3, 5, 6, 8

[15] T. Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P. Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *NeurIPS*, 2018. 1, 5

[16] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *ICLR*, 2022. 3

[17] Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. Learning to branch for multi-task learning. In *ICML*, 2020. 3

[18] Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. Ptr: Prompt tuning with rules for text classification. *arXiv:2105.11259*, 2021. 3

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *TPAMI*, 2020. 2, 3

[21] 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*, 2019. 6

[22] Siyu Huang, Xi Li, Zhi-Qi Cheng, Zhongfei Zhang, and Alexander Hauptmann. Gnas: A greedy neural architecture search method for multi-attribute learning. In *ACMMM*, 2018. 3

[23] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 1, 3

[24] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge J. Belongie, Bharath Hariharan, and Ser Nam Lim. Visual prompt tuning. In *ECCV*, 2022. 3

[25] Zhengbao Jiang, Frank F. Xu, J. Araki, and Graham Neubig. How can we know what language models know? *TACL*, 2020. 3

[26] Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know. In *EMNLP*, 2020. 3
[27] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *ECCV*, 2022. 3
[28] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, 2013. 6
[29] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 6
[30] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv:2104.08691*, 2021. 3
[31] Boyi Li, Kilian Q. Weinberger, Serge J. Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. In *arXiv:2201.03546*, 2022. 3
[32] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. In *CVPR*, 2022. 3
[33] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL*, 2021. 3
[34] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In *ICLR*, 2022. 1, 3
[35] Xiwen Liang, Yangxin Wu, Jianhua Han, Hang Xu, Chunjing Xu, and Xiaodan Liang. Effective adaptation in multi-task co-training for unified autonomous driving. *arXiv:2209.08953*, 2022. 3
[36] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *NeurIPS*, 2021. 3, 5
[37] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *ECCV*, 2018. 3
[38] Liyang Liu, Yi Li, Zhanghui Kuang, J Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *ICLR*, 2021. 3, 6
[39] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Philip S Yu. Learning multiple tasks with multilinear relationship networks. In *NeurIPS*, 2017. 6
[40] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *CVPR*, 2022. 2, 3, 6
[41] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. *arXiv:1306.5151*, 2013. 6
[42] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, 2016. 3
[43] Norman Mu, Alexander Kirillov, David A. Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. In *ECCV*, 2022. 3
[44] Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. *arXiv:2202.01017*, 2022. 3
[45] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008. 2, 6
[46] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, 2012. 6
[47] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *CVPR*, 2019. 6
[48] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. Language models as knowledge bases? *arXiv:1909.01066*, 2019. 3
[49] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 3, 6, 8
[50] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. 3
[51] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 3
[52] Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv:1910.10683*, 2020. 3
[53] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *CVPR*, 2022. 3
[54] Dripta S Raychaudhuri, Yumin Suh, Samuel Schulter, Xiang Yu, Masoud Faraki, Amit K Roy-Chowdhury, and Manmohan Chandraker. Controllable dynamic multi-task architectures. In *CVPR*, 2022. 3, 6
[55] Clemens Rosenbaum, Tim Klinger, and Matthew Riemer. Routing networks: Adaptive selection of non-linear functions for multi-task learning. In *ICLR*, 2017. 6
[56] Jiayi Shen, Xiantong Zhen, Marcel Worring, and Ling Shao. Variational multi-task learning with gumbel-softmax priors. In *NeurIPS*, 2021. 6
[57] Taylor Shin, Yasaman Razeghi, Robert L. Logan, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *EMNLP*, 2020. 3
[58] Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. *arXiv:1910.05653*, 2020. 5
[59] Peter H. A. Sneath and Robert R. Sokal. Numerical taxonomy: The principles and practice of numerical classification. 1973. 5, 6

[60] Xiaozhuang Song, Shun Zheng, Wei Cao, James Yu, and Jiang Bian. Efficient and effective multi-task grouping via meta learning on task combinations. In *NeurIPS*, 2022. 3

[61] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv:1212.0402*, 2012. 6

[62] Trevor Scott Standley, Amir Roshan Zamir, Dawn Chen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? *arXiv:1905.07553*, 2020. 3, 5

[63] Ximeng Sun, Ping Hu, and Kate Saenko. Dualcoop: Fast adaptation to multi-label recognition with limited annotations. *arXiv:2206.09541*, 2022. 3

[64] Simon Vandenhende, Stamatios Georgoulis, Bert De Brabandere, and Luc Van Gool. Branched multi-task networks: deciding what layers to share. *arXiv:1904.02920*, 2019. 3

[65] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017. 1, 6

[66] Mitchell Wortsman, Gabriel Ilharco, Mike Li, Jong Wook Kim, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *CVPR*, 2022. 3

[67] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 6

[68] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and X. Wang. Groupvit: Semantic segmentation emerges from text supervision. In *CVPR*, 2022. 3

[69] Jianwei Yang, Chengkun Li, Pengchuan Zhang, Bin Xiao, Ce Liu, Lu Yuan, and Jianfeng Gao. Unified contrastive learning in image-text-label space. In *CVPR*, 2022. 3

[70] Yongxin Yang and Timothy M Hospedales. Trace norm regularised deep multi-task learning. *arXiv:1606.04038*, 2016. 3

[71] Lewei Yao, Runhu Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. In *ICLR*, 2022. 1, 3

[72] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv:2205.01917*, 2022. 3

[73] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *NeurIPS*, 2020. 2, 3, 5, 6

[74] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel C. F. Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision. *arXiv:2111.11432*, 2021. 1, 3

[75] Amir Roshan Zamir, Alexander Sax, Bokui (William) Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, 2018. 2, 3

[76] Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Jiao Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv:2111.03930*, 2021. 3

[77] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chengkun Li, Noel C. F. Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, and Jianfeng Gao. Regionclip: Region-based language-image pretraining. In *CVPR*, 2022. 3

[78] Zexuan Zhong, Dan Friedman, and Danqi Chen. Factual probing is [mask]: Learning vs. learning to recall. In *NAACL*, 2021. 3

[79] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, pages 16816–16825, 2022. 1, 2, 3, 6

[80] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 2022. 1, 2, 3, 6, 8

[81] Beier Zhu, Yulei Niu, Yucheng Han, Yuehua Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. *arXiv:2205.14865*, 2022. 2, 3, 6