Hierarchical Prompt Learning for Multi-Task Learning

Yajing Liu\textsuperscript{1\,*}, Yuning Lu\textsuperscript{2\,*}, Hao Liu\textsuperscript{1\,†}, Yaozu An\textsuperscript{1}, Zhuoran Xu\textsuperscript{1}, Zhuokun Yao\textsuperscript{1}, Baofeng Zhang\textsuperscript{1}, Zhiwei Xiong\textsuperscript{2}, Chenguang Gui\textsuperscript{1}
\textsuperscript{1}JD Logistics \textsuperscript{2}University of Science and Technology of China

\{liuyajing25,liuhao164,anyaozu,yaozuokun1,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.

\textsuperscript{*}Equal contribution.
\textsuperscript{†}Corresponding author.

\textbf{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 ($w_{\text{rand}}, w_{\text{ind}},$ and $w_{\text{shr}}$), 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 ($w_{\text{avg}} = \frac{1}{2}(w_{\text{shr}} + w_{\text{ind}})$) 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
The benefits of prompt learning with multiple tasks. Figure 2. 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, 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 \( x \) are denoted as \( f(x) \). Text features of various class descriptions \( \{t_i\} \) can be denoted as \( \{g(t_i)\} \), 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 \( x \) can be classified to the \( i \)-th class with the largest (cosine) similarity \( f(x) \cdot g(t_i) \) 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 \( p \) 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 \( \{t_i(p)\} \) based on the prompt \( p \) are construed by combining \( p \) and the word embeddings of \( K \) class names. Actually, the matrix \( [g(t_1(p)), ..., g(t_K(p))] \) can be considered as the weights of a \( K \)-way linear classifier (denoted as \( w(p) \)). Then, CoOp
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 $\{T_i\}_{i=1}^N$, $\mathcal{G}$ 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 $p_\mathcal{G}$ represents the learned prompt for tasks of $\mathcal{G}$. We train $p_\mathcal{G}$ by minimizing the following loss:

$$L(p_\mathcal{G}) = \sum_{T_i \in \mathcal{G}} L_{T_i}(p_\mathcal{G}),$$

where $L_{T_i}$ 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 $p_{\text{all}}$) for all tasks and the task-individual prompt (denoted as $p_j$) for each task ($j=1, \ldots, N$). 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}(w(p_{\text{all}})+w(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
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 $p_{all}$ for all target tasks, the affinity between the $i$-th task and the $j$-th task can be estimated as the following dot product,

$$d_{T_i, T_j}(p_{all}) = \nabla_{p_{all}} C_{T_i}(p_{all}) \cdot \nabla_{p_{all}} C_{T_j}(p_{all}).$$ (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 $d_{T_i, T_j}$, 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 \(d(G_a, G_b)\)) of two task groups \((G_a, G_b)\) by averaging all affinities of their task pairs,

\[
\begin{align*}
    d(G_a, G_b) = \frac{1}{|G_a||G_b|} \sum_{T_i \in G_a, T_j \in G_b} d_{T_i, T_j},
\end{align*}
\]

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 \(\{G_i\}_{i=1}^M\) (including groups with a single task), HiPro independently learns \(M\) corresponding prompts \(\{p_{G_i}\}_{i=1}^M\). 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 \(w_{T_j}\) which used to infer test samples of the \(j\)-th target task is:

\[
    w_{T_j} = \frac{\sum_{i=1}^M w(p_{G_i}) \mathbb{1}(T_j \in G_i)}{\sum_{i=1}^M \mathbb{1}(T_j \in G_i)},
\]

where \(\mathbb{1}\) 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 FGVC Aircraft [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) CoOp [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.*
### 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 cannot outperform the basic MTL baseline CoOp-MTL, even worse than CoOp. However, HiPro shows non-trivial 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,
Table 3. Comparison to various methods on CIFAR-100, using the average accuracy (%) over 3 runs.

<table>
<thead>
<tr>
<th>Method</th>
<th>DomainNet</th>
<th>Office-Home</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZeroShot</td>
<td>0.00 ± 0.00</td>
<td>0.00 ± 0.00</td>
</tr>
<tr>
<td>CoOp</td>
<td>75.59 ± 0.21</td>
<td>76.68 ± 0.23</td>
</tr>
<tr>
<td>CoOp-MTL</td>
<td>74.53 ± 0.59</td>
<td>77.36 ± 0.31</td>
</tr>
<tr>
<td>Shr+Ind</td>
<td>51.12 ± 0.07</td>
<td>51.63 ± 0.12</td>
</tr>
<tr>
<td>rand Group</td>
<td>49.39 ± 0.53</td>
<td>49.92 ± 0.17</td>
</tr>
<tr>
<td>HiPro (Ours)</td>
<td>53.14 ± 0.10</td>
<td>53.35 ± 0.10</td>
</tr>
</tbody>
</table>

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

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 close 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.

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


[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


[23] Chao Jia, Yinfeng 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


[71] Lewis 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


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


[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