Visual Prompt Tuning

Menglin Jia*[1,2], Luming Tang*[1]
Bor-Chun Chen[2], Claire Cardie[1], Serge Belongie[3]
Bharath Hariharan[1], and Ser-Nam Lim[2]

[1]Cornell University [2]Meta AI [3]University of Copenhagen

Abstract. The current *modus operandi* in adapting pre-trained models involves updating all the backbone parameters, *i.e.*, full fine-tuning. This paper introduces Visual Prompt Tuning (VPT) as an efficient and effective alternative to full fine-tuning for large-scale Transformer models in vision. Taking inspiration from recent advances in efficiently tuning large language models, VPT introduces only a small amount (less than 1% of model parameters) of trainable parameters in the input space while keeping the model backbone frozen. Via extensive experiments on a wide variety of downstream recognition tasks, we show that VPT achieves significant performance gains compared to other parameter efficient tuning protocols. Most importantly, VPT even outperforms full fine-tuning in many cases across model capacities and training data scales, while reducing per-task storage cost. Code is available at github.com/kmnp/vpt.

1 Introduction

For a variety of recognition applications, the most accurate results are now obtained by adapting large *foundation models* pre-trained on massive curated or raw data, a finding that mirrors developments in natural language processing (NLP) [5].[1] At first glance,this is a success story: one can make rapid progress on multiple recognition problems simply by leveraging the latest and greatest foundation model. In practice, however, *adapting* these large models to downstream tasks presents its own challenges. The most obvious (and often the most effective) adaptation strategy is *full fine-tuning* of the pre-trained model on the task at hand, end-to-end. However, this strategy requires one to store and deploy a separate copy of the backbone parameters for every single task. This is an expensive and often infeasible proposition, especially for modern *Transformer*-based architectures, which are significantly larger than their convolutional neural networks (ConvNet) counterparts, *e.g.*, ViT-Huge [15] (632M parameters) *vs*. ResNet-50 [24] (25M parameters). We therefore ask, **what is the best way to adapt large pre-trained Transformers to downstream tasks in terms of effectiveness and efficiency**?

*Equal contribution.

[1]As pointed out in [5], all state-of-the-art models in contemporary NLP are now powered by a few Transformer-based models (*e.g.*, BERT [13], T5 [49], BART [34], GPT-3 [6]) This also applies to vision-language field recently, *i.e.*, CLIP [48].

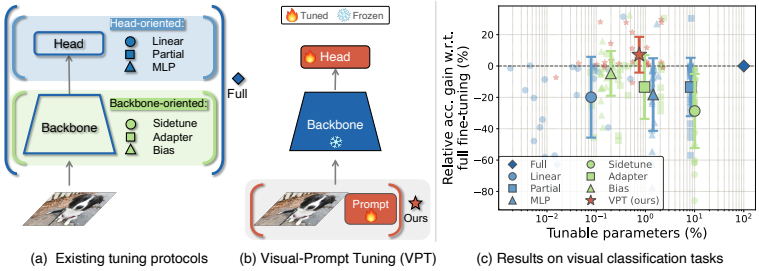

Fig. 1. Visual-Prompt Tuning (VPT) *vs*. other transfer learning methods. (a) Current transfer learning protocols are grouped based on the tuning scope: Full fine-tuning, Head-oriented, and Backbone-oriented approaches. (b) VPT instead adds extra parameters in the input space. (c) Performance of different methods on a wide range of downstream classification tasks adapting a pre-trained ViT-B backbone, with mean and standard deviation annotated. VPT outperforms Full fine-tuning 20 out of 24 cases while using less than 1% of all model parameters

One straightforward approach is to turn to other strategies that we have perfected for adapting ConvNets to new tasks, as in Fig. 1(a). A popular approach is to fine-tune only a subset of the parameters, such as the classifier head [42,28,10] or the bias terms [7]. Prior research has also looked at adding additional residual blocks (or *adapters*) to the backbone [51,66]. One could implement similar strategies for Transformers. However, in general these strategies *under-perform* full fine-tuning in accuracy.

We explore a different route in this paper. Instead of altering or fine-tuning the pre-trained Transformer itself, we modify the *input* to the Transformer. Drawing inspiration from the recent advances on Prompting in NLP [37,35,33,38], we propose a new simple and efficient method to adapt transformer models for downstream vision tasks (Fig. 1(b)), namely **Visual-Prompt Tuning** (VPT). Our method only introduces a small amount of task-specific learnable parameters into the input space while freezing the entire pre-trained Transformer backbone during downstream training. In practice, these additional parameters are simply prepended into the input sequence of each Transformer layer and learned together with a linear head during fine-tuning.

On 24 downstream recognition tasks spanning different domains using a pre-trained ViT backbone, VPT beats all other transfer learning baselines, even surpassing full fine-tuning in 20 cases, while maintaining the advantage of storing remarkably fewer parameters (less than 1% of backbone parameters) for each individual task (Fig. 1(c)). This result demonstrates the distinctive strength of *visual* prompting: whereas in NLP, prompt tuning is only able to *match* full fine-tuning performance under certain circumstances [33]. VPT is especially effective in the low-data regime, and maintains its advantage across data scales. Finally, VPT is competitive for a range of Transformer scales and designs (ViT-Base/Large/Huge, Swin). Put together, our results suggest that VPT is one of the most effective ways of adapting ever-growing vision backbones.

2 Related Work

Transformer models [56] have gained huge success in NLP [13,49,6]. The triumph of the Transformer architecture also extends to various computer vision tasks, including image classification [15,39], object detection [8,36], semantic and panoptic segmentation [54,68,60], video understanding [20,61,17] and few-shot learning [14], surpassing previous state-of-the-art approaches. Transformers are also being widely used in recent self-supervised pre-training methods [10,23,3]. Given their superior performance and much larger scale compared to ConvNets, how to efficiently adapt Transformers to different vision tasks remains an important open problem. Our proposed VPT provides a promising path forward.

Transfer learning has been extensively studied for vision tasks in the context of ConvNets [71] and many techniques have been introduced including side tuning [66], residual adapter [50], bias tuning [7], *etc*. Relatively little attention has been paid to vision Transformers adaptation and how well these aforementioned methods perform on this brand new type of architecture remains unknown. On the other hand, given the dominance of large-scale pre-trained Transformer-based Language Models (LM) [13,49,6], many approaches [22,21,27] have been proposed to efficiently fine-tune LM for different downstream NLP tasks [59,58]. Among them, we focus on the following two representative methods in our experiments for benchmarking purposes: Adapters [47] and BitFit [4].

Adapters [26] insert extra lightweight modules inside each Transformer layer. One adapter module generally consists of a linear down-projection, followed by a nonlinear activation function, and a linear up-projection, together with a residual connection [46,47]. Instead of inserting new modules, [7] proposed to update the bias term and freeze the rest of backbone parameters when fine-tuning ConvNets. BitFit [3] applied this technique to Transformers and verified its effectiveness on LM tuning. Our study demonstrates that VPT, in general, provides improved performance in adapting Transformer models for vision tasks, relative to the aforementioned two well-established methods in NLP.

Prompting [37] originally refers to prepending language instruction to the input text so that a pre-trained LM can "understand" the task. With manually chosen prompts, GPT-3 shows strong generalization to downstream transfer learning tasks even in the few-shot or zero-shot settings [6]. In addition to the follow-up works on how to construct better prompting texts [53,29], recent works propose to treat the prompts as task-specific continuous vectors and directly optimize them via gradients during fine-tuning, namely Prompt Tuning [35,33,38]. Compared to full fine-tuning, it achieves comparable performance but with 1000× less parameter storage. Although prompting has also been applied to vision-language models recently [48,70,31,63,18], prompting is still limited to the input of *text* encoders. Due to the disparity between vision and language modalities, in this paper we ask: can the same method can be applied successfully to image encoders? We are the first work (see related concurrent works [52,62,11,2]) to tackle this question and investigate the generality and feasibility of visual prompting via *extensive* experiments spanning multiple kinds of recognition tasks across multiple domains and backbone architectures.

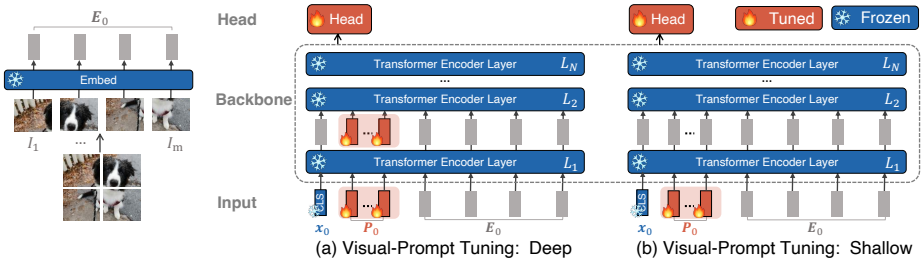

Fig. 2. Overview of our proposed Visual-Prompt Tuning. We explore two variants: (a) prepend a set of learnable parameters to each Transformer encoder layer's input (VPT-DEEP); (b) only insert the prompt parameters to the first layer's input (VPT-SHALLOW). During training on downstream tasks, only the parameters of prompts and linear head are updated while the whole Transformer encoder is frozen.

3 Approach

We propose Visual-Prompt Tuning (VPT) for adapting large pre-trained vision Transformer models. VPT injects a small number of learnable parameters into Transformer's input space and keeps the backbone frozen during the downstream training stage. The overall framework is presented in Fig. 2. We first define the notations in Sec. 3.1, then describe VPT formally in Sec. 3.2.

3.1 Preliminaries

For a plain Vision Transformer (ViT) [15] with N layers, an input image is divided into m fixed-sized patches $\{I_j \in \mathbb{R}^{3\times h\times w} \mid j \in \mathbb{N}, 1 \le j \le m\}$. h, w are the height and width of the image patches. Each patch is then first embedded into d-dimensional latent space with positional encoding:

$$\mathbf{e}_0^j = \texttt{Embed}(I_j) \qquad \mathbf{e}_0^j \in \mathbb{R}^d, j = 1, 2, \dots m \ . \tag{1}$$

We denote the collection of image patch embeddings, $\mathbf{E}_i = \{\mathbf{e}_i^j \in \mathbb{R}^d \mid j \in \mathbb{N}, 1 \le j \le m\}$, as inputs to the $(i+1)$-th Transformer layer (L_{i+1}). Together with an extra learnable classification token (`[CLS]`), the whole ViT is formulated as:

$$[\mathbf{x}_i, \mathbf{E}_i] = L_i([\mathbf{x}_{i-1}, \mathbf{E}_{i-1}]) \qquad i = 1, 2, \dots, N \tag{2}$$

$$\mathbf{y} = \texttt{Head}(\mathbf{x}_N) \ , \tag{3}$$

where $\mathbf{x}_i \in \mathbb{R}^d$ denote `[CLS]`'s embedding at L_{i+1}'s input space. $[\cdot, \cdot]$ indicates stacking and concatenation on the sequence length dimension, *i.e.*, $[\mathbf{x}_i, \mathbf{E}_i] \in \mathbb{R}^{(1+m)\times d}$. Each layer L_i consists of Multiheaded Self-Attention (MSA) and Feed-Forward Networks (FFN) together with LayerNorm [1] and residual con-

nections [24]. A neural classification head is used to map the final layer's [CLS] embedding, $\mathbf{x}_N$, into a predicted class probability distribution $\mathbf{y}$.[2]

3.2 Visual-Prompt Tuning (VPT)

Given a pre-trained Transformer model, we introduce a set of p continuous embeddings of dimension d, *i.e.*, *prompts*, in the input space after the Embed layer. Only the task-specific prompts are being updated during fine-tuning, while the Transformer backbone is kept frozen. Depending on the number of Transformer layers involved, our approach has two variants, VPT-SHALLOW and VPT-DEEP, as shown in Fig. 2.

VPT-Shallow. Prompts are inserted into the first Transformer layer L_1 only. Each prompt token is a learnable d-dimensional vector. A collection of p prompts is denoted as $\mathbf{P} = \{\mathbf{p}^k \in \mathbb{R}^d \mid k \in \mathbb{N}, 1 \leq k \leq p\}$, the shallow-prompted ViT is:

$$[\mathbf{x}_1, \mathbf{Z}_1, \mathbf{E}_1] = L_1([\mathbf{x}_0, \mathbf{P}, \mathbf{E}_0]) \tag{4}$$

$$[\mathbf{x}_i, \mathbf{Z}_i, \mathbf{E}_i] = L_i([\mathbf{x}_{i-1}, \mathbf{Z}_{i-1}, \mathbf{E}_{i-1}]) \qquad i = 2, 3, \ldots, N \tag{5}$$

$$\mathbf{y} = \texttt{Head}(\mathbf{x}_N) \ , \tag{6}$$

where $\mathbf{Z}_i \in \mathbb{R}^{p \times d}$ represents the features computed by the i-th Transformer layer, and $[\mathbf{x}_i, \mathbf{Z}_i, \mathbf{E}_i] \in \mathbb{R}^{(1+p+m) \times d}$. The colors • and • indicate learnable and frozen parameters, respectively. Notably for ViT, $\mathbf{x}_N$ is invariant to the location of prompts since they are inserted after positional encoding, *e.g.*, $[\mathbf{x}_0, \mathbf{P}, \mathbf{E}_0]$ and $[\mathbf{x}_0, \mathbf{E}_0, \mathbf{P}]$ are mathematically equivalent. This also applies to VPT-Deep.

VPT-Deep. Prompts are introduced at *every* Transformer layer's input space. For $(i+1)$-th Layer L_{i+1}, we denote the collection of input learnable prompts as $\mathbf{P}_i = \{\mathbf{p}_i^k \in \mathbb{R}^d \mid k \in \mathbb{N}, 1 \leq k \leq m\}$. The deep-prompted ViT is formulated as:

$$[\mathbf{x}_i, \_\_, \mathbf{E}_i] = L_i([\mathbf{x}_{i-1}, \mathbf{P}_{i-1}, \mathbf{E}_{i-1}]) \qquad i = 1, 2, \ldots, N \tag{7}$$

$$\mathbf{y} = \texttt{Head}(\mathbf{x}_N) \ . \tag{8}$$

Storing Visual Prompts. VPT is beneficial in presence of multiple downstream tasks. We only need to store the learned prompts and classification head for each task and re-use the original copy of the pre-trained Transformer model, significantly reducing the storage cost. For instance, given a ViT-Base with 86 million (M) parameters and $d = 768$, 50 shallow prompts and deep prompts yield additional $p \times d = 50 \times 768 = 0.038$M, and $N \times p \times d = 0.46$M parameters, amounting to only 0.04% and 0.53% of all ViT-Base parameters, respectively.

[2] Some Transformer architectures in Vision such as Swin [39] do not use [CLS] and treat global pooled $\mathbf{E}_N$ as input for Head. We follow their designs when adapting VPT to these Transformer variants. See the Appendix for more details.

4 Experiments

We evaluate VPT for a wide range of downstream recognition tasks with pre-trained Transformer backbones across scales. We first describe our experimental setup in Sec. 4.1, including the pre-trained backbone and downstream tasks, and a brief introduction of alternative transfer learning methods. Then we demonstrate the effectiveness and practical utility of our method in Sec. 4.2. We also systematically study how different design choices would affect performance (Sec. 4.3), which leads to an improved understanding of our approach.

4.1 Experiment Setup

Pre-trained Backbones. We experiment with two Transformer architectures in vision, Vision Transformers (ViT) [15] and Swin Transformers (Swin [39]). All backbones in this section are pre-trained on ImageNet-21k [12]. We follow the original configurations, *e.g.*, number of image patches divided, existence of `[CLS]`, *etc*. More details are included in the Appendix.

Baselines. We compare both variants of VPT with other commonly used fine-tuning protocols:

`(a)` FULL: fully update *all* backbone and classification head parameters.

`(b)` Methods that focus on the classification head. They treat the pre-trained backbone as a feature extractor, whose weights are fixed during tuning:
- LINEAR: only use a linear layer as the classification head.
- PARTIAL-k: fine-tune the last k layers of backbone while freezing the others, as adopted in [64,67,45,23]. It redefines the boundary of backbone and classification head.
- MLP-k: utilize a multilayer perceptron (MLP) with k layers, instead of a linear layer, as classification head.

`(c)` Methods that update a subset backbone parameters or add new trainable parameters to backbone during fine-tuning:
- SIDETUNE [66]: train a "side" network and linear interpolate between pre-trained features and side-tuned features before being fed into the head.
- BIAS [7,4]: fine-tune only the bias terms of a pre-trained backbone.
- ADAPTER [26,46,47]: insert new MLP modules with residual connection inside Transformer layers.

Downstream Tasks. We experiment on the following two collections of datasets:

FGVC consists of 5 benchmarked Fine-Grained Visual Classification tasks including CUB-200-2011 [57], NABirds [55], Oxford Flowers [44], Stanford Dogs [32] and Stanford Cars [19]. If a certain dataset only has `train` and `test` sets publicly available, we randomly split the training set into `train` (90%) and `val` (10%), and rely on `val` to select hyperparameters.

VTAB-1k [65] is a collection of 19 diverse visual classification tasks, which are organized into three groups: *Natural* - tasks that contain natural images captured using standard cameras; *Specialized* - tasks that contain images captured via specialized equipment, such as medical and satellite imagery; and *Structured* - tasks that require geometric comprehension like object counting. Each task

Table 1. ViT-B/16 pre-trained on supervised ImageNet-21k. For each method and each downstream task group, we report the average test accuracy score and `number of wins in (·)` compared to FULL. "Total params" denotes total parameters needed for all 24 downstream tasks. "Scope" denotes the tuning scope of each method. "Extra params" denotes the presence of additional parameters besides the pre-trained backbone and linear head. Best results among all methods except FULL are **bolded**. VPT outshines the full fine-tuning 20 out of 24 cases with significantly less trainable parameters

	ViT-B/16 (85.8M)	Total params	Scope		Extra params	FGVC	VTAB-1k		
			Input	Backbone			Natural	Specialized	Structured
	Total # of tasks					5	7	4	8
(a)	FULL	24.02×		✓		88.54	75.88	83.36	47.64
(b)	LINEAR	1.02×				79.32 (0)	68.93 (1)	77.16 (1)	26.84 (0)
	PARTIAL-1	3.00×				82.63 (0)	69.44 (2)	78.53 (0)	34.17 (0)
	MLP-3	1.35×			✓	79.80 (0)	67.80 (2)	72.83 (0)	30.62 (0)
(c)	SIDETUNE	3.69×		✓	✓	78.35 (0)	58.21 (0)	68.12 (0)	23.41 (0)
	BIAS	1.05×		✓		88.41 (3)	73.30 (3)	78.25 (0)	44.09 (2)
	ADAPTER	1.23×		✓	✓	85.66 (2)	70.39 (4)	77.11 (0)	33.43 (0)
(ours)	VPT-SHALLOW	1.04×	✓		✓	84.62 (1)	76.81 (4)	79.66 (0)	46.98 (4)
	VPT-DEEP	1.18×				**89.11** (**4**)	**78.48** (**6**)	**82.43** (**2**)	**54.98** (**8**)

of VTAB contains 1000 training examples. Following [65], we use the provided 800-200 split of the `train` set to determine hyperparameters and run the final evaluation using the full training data. We report the average accuracy score on `test` set within three runs.

We report the average accuracy on the FGVC datasets, and the average accuracy on each of the three groups in VTAB. The individual results on each task are in the Appendix, as are image examples of these aforementioned tasks.

4.2 Main Results

Tab. 1 presents the results of fine-tuning a pre-trained ViT-B/16 on averaged across 4 diverse downstream task groups, comparing VPT to the other 7 tuning protocols. We can see that:

1. **VPT-Deep outperforms Full (Tab. 1(a)) on 3 out of the 4 problem classes** (20 out of 24 tasks), while using significantly fewer total model parameters (1.18× *vs*. 24.02×). Thus, *even if storage is not a concern*, VPT is a promising approach for adapting larger Transformers in vision. Note that this result is in contrast to comparable studies in NLP, where prompt tuning matches, but *does not exceed* full fine-tuning [33].
2. **VPT-Deep outperforms all the other parameter-efficient tuning protocols (Tab. 1(b,c)) across all task groups**, indicating that VPT-DEEP is the best fine-tuning strategy in storage-constrained environments.
3. Although sub-optimal than VPT-DEEP, VPT-SHALLOW still offers non-trivial performance gain than head-oriented tuning methods in Tab. 1(b), indicating that VPT-SHALLOW is a worthwhile choice in deploying multi-task fine-tuned models if the storage constraint is severe.

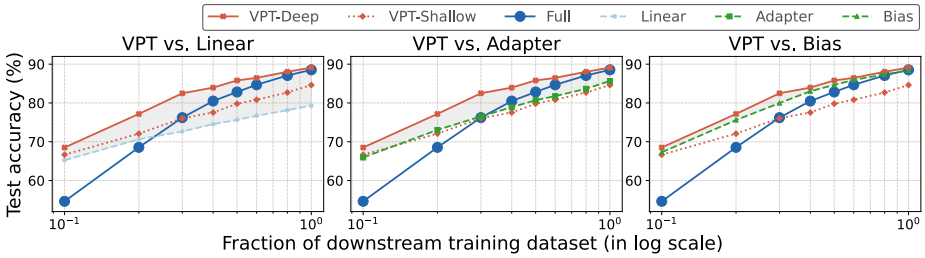

Fig. 3. Performance comparison on different downstream data scales, averaged across 5 FGVC tasks. VPT-DEEP is compared with LINEAR (left), ADAPTER (middle) and BIAS (right). Highlighted region shows the accuracy difference between VPT-DEEP and the compared method. Results of VPT-SHALLOW are FULL presented in all plots for easy reference. The size of markers are proportional to the percentage of tunable parameters in log scale

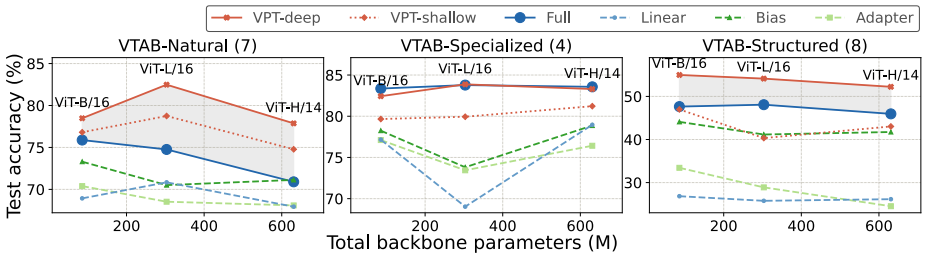

Fig. 4. VPT *vs.* FULL across model scales (ViT-B, ViT-L and ViT-H), for 3 VTAB task groups. Highlighted region shows the accuracy difference between VPT-DEEP and the full fine-tuning (FULL). The size of markers are proportional to the percentage of trainable parameters in log scale

VPT on different downstream data size. We look at the impact of training data size on accuracy in the FGVC tasks (VTAB has only 1k training examples). We vary the training data between 10% and 80% and compare all methods. The same pre-trained ViT-B is used for downstream training. Task-averaged results for each method on different training data scales are presented in Fig. 3.

Fig. 3 shows that VPT-DEEP outperforms all the other baselines across data scales. Digging deeper, methods that use less trainable parameters, *i.e.*, VPT, LINEAR, ADAPTER, BIAS, dominate over FULL in the low-data regimes. This trend, however, is *reversed* when more training data is available for LINEAR and ADAPTER. In contrast, VPT-DEEP still consistently outperforms FULLacross training data sizes. Although BIAS offers similar advantages, it still marginally under-performs VPT-DEEP across the board (Fig. 3 right).

VPT on different backbone scales. Fig. 4 shows VTAB-1k performance under 3 different backbone scales: ViT-**B**ase/**L**arge/**H**uge. VPT-DEEP is signif-

Table 2. Different Transformer architecture: Swin-B pre-trained on supervised ImageNet-21k as backbone. For each method and each downstream task group, we report the average test accuracy score and `number of wins in (·)` compared to FULL. The column "Total params" denotes total parameters needed for all 19 downstream tasks. Best results among all methods except FULL are **bolded**

	Swin-B (86.7M)	Total params	VTAB-1k Natural	VTAB-1k Specialized	VTAB-1k Structured
	Total # of tasks		7	4	8
(a)	FULL	19.01×	79.10	86.21	59.65
(b)	LINEAR	1.01×	73.52 (5)	80.77 (0)	33.52 (0)
	MLP-3	1.47×	73.56 (5)	75.21 (0)	35.69 (0)
	PARTIAL	3.77×	73.11 (4)	81.70 (0)	34.96 (0)
(c)	BIAS	1.06×	74.19 (2)	80.14 (0)	42.42 (0)
(ours)	VPT-SHALLOW	1.01×	**79.85** (**6**)	82.45 (0)	37.75 (0)
	VPT-DEEP	1.05×	76.78 (**6**)	**84.53** (0)	**53.35** (0)

icantly better than LINEAR and VPT-SHALLOW across all 3 backbone choices and 3 subgroups of VTAB-1k. More importantly, the advantages of VPT-DEEP over FULL indeed still hold as the model scale increases, *i.e.*, VPT-DEEP significantly outperforms FULL on *Natural* and *Structured* groups, while offering nearly equivalent performance on *Specialized*.

VPT on hierarchical Transformers. We extend VPT to Swin [39], which employs MSA within local shifted windows and merges patch embeddings at deeper layers. For simplicity and without loss of generality, we implement VPT in the most straightforward manner: the prompts are attended within the local windows, but are ignored during patch merging stages. The experiments are conducted on the ImageNet-21k supervised pre-trained Swin-**B**ase. VPT continues to outperform other parameter-efficient fine-tuning methods (`b, c`) for all three subgroups of VTAB Tab. 2, though in this case FULL yields the highest accuracy scores overall (at a heavy cost in total parameters).

It is surprising that the advantage of VPT-DEEP over VPT-SHALLOW diminishes for *Natural*: VPT-SHALLOW yields slightly better accuracy scores than full fine-tuning.

4.3 Ablation on Model Design Variants

We ablate different model design choices on the supervised ImageNet-21k pre-trained ViT-Base and evaluate them on VTAB, with same setup in Tab. 1. See more in the Appendix.

Prompt Location. An important distinction between VPT and other methods is the extra learnable parameters introduced as *inputs* for the Transformer layers. Fig. 5 ablates different choices on how and where to insert prompts in the input space, and how they would affect the final performance.

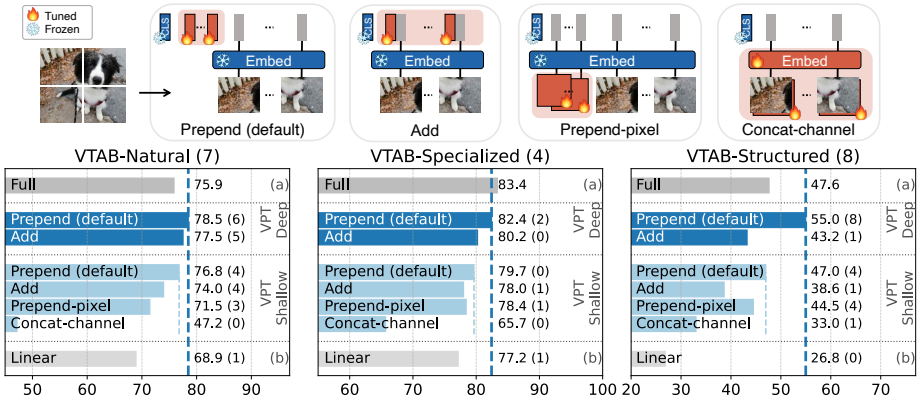

Fig. 5. Ablation on prompt location. We illustrate different location choices at top, and present the results at bottom. For easy comparison, two blue dashed lines represent the performance of the default VPT-DEEP and VPT-SHALLOW respectively

Prepend or Add? Instead of prepending prompts to the sequence of the image patches embeddings $\mathbf{E}_i$ as described in Sec. 3.2, another option is to directly *add* prompts element-wise to those embeddings, keeping the Transformer's input sequence length the same as before. Though this variant is competitive to FULL in some cases (*e.g.*, VTAB-*Natural*), its performance generally falls behind with the default `Prepend` in both deep and shallow settings. More discussion on this phenomenon is in the Appendix.

Latent or pixel space? Instead of inserting the prompts as latent vectors for the first Transformer layer, one could introduce prompts in the *pixel* level before the `Embed` layer in Eq. (1), *i.e.*, `Prepend-pixel` and `Concat-channel`. Fig. 5 shows that the adaption performance *decreases* for these two variants. For example, the accuracy score of prepending shallow prompts before the projection layer (`Prepend-pixel`) drops 6.9%, compared to the default prepending in the embedding space (`Prepend`) on VTAB-*Natural*. The performance further deteriorates (even as large as 30 accuracy scores drop on VTAB-*Natural*) if we instead concatenate a new channel to the input image (`Concat-channel`). These observations suggest that it's easier for prompts to learn condensed task-dependent signals in the latent input space of Transformers.

Prompt Length. This is the only additional hyper-parameter needed to tune for VPT compared to full fine-tuning. For easy reference, we also ablate two other baselines on their individual additional hyper-parameters, *i.e.*, number of layers for MLP and reduction rate for ADAPTER. As shown in Fig. 6, the optimal prompt length varies across tasks. Notably, even with as few as only *one* prompt, VPT-DEEP still significantly outperforms the other 2 baselines, and remains competitive or even better compared to full fine-tuning on VTAB-*Structured* and *Natural*.

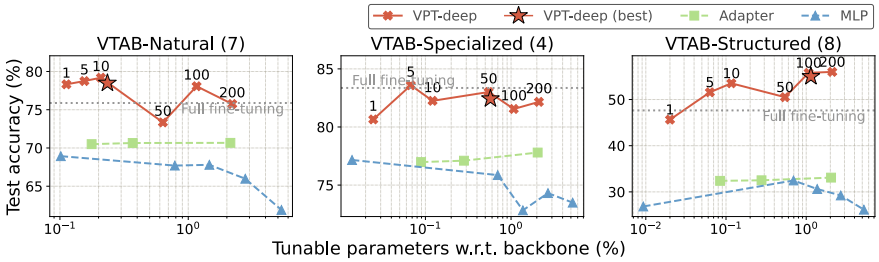

Fig. 6. Ablation on prompt length. We vary the number of prompts for VPT-DEEP and show the averaged results for each VTAB subgroup. The averaged best VPT-DEEP results for each task is also shown for easy reference

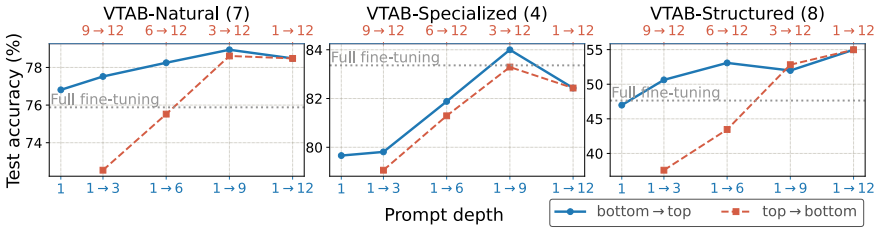

Fig. 7. Ablation on prompt depth. We select the best prompt length for each variant with `val` sets. $i \rightarrow j$ indicates the Transformer layer indices that prompts are inserted into. The 1-st layer refers to the one closest to input. ViT-B has 12 layers in total

Prompt Depth. Fig. 7 ablates which and how many layers to insert prompts. Each variant reports the best prompt length selected with `val` set. VPT's performance is positively correlated with the prompt depth in general. Yet the accuracy drops if we insert prompts from top to bottom, suggesting that prompts at earlier Transformer layers matter more than those at later layers.

Final Output. Following the original configuration of ViT, we use the final embedding of `[CLS]`, *i.e.*, $\mathbf{x}_N$, as the classification head input, which is also the default setting in our ViT experiments. As shown in Fig. 8, if we use the average pooling on image patch output embeddings $\mathbf{E}_N$ as final output (`Image-pool`), the results essentially remain the same (*e.g.*, 82.4 *vs*. 82.3 for VTAB-*Specialized*). However, if the pooling involves final prompt outputs $\mathbf{Z}_N$ (`Prompt-pool` and `Global-pool`), the accuracy could drop as large as 8 points.

5 Analysis and Discussion

Visualization. Fig. 9 shows t-SNE [41] visualizations of $\mathbf{x_N}$, *i.e.*, embeddings of `[CLS]` after the last Transformer layer and before the classification head, for 3 tasks in VTAB (SVNH [43], EuroSAT [25], Clevr/count [30]), one for each

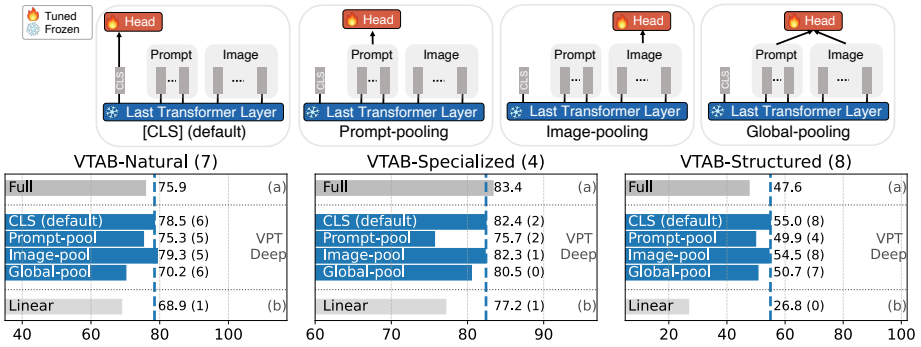

Fig. 8. Ablation on final output. Illustration of different strategies is included at top, and results of those are presented at the bottom section. For easy comparison, the blue dashed line represents the performance of default VPT-DEEP

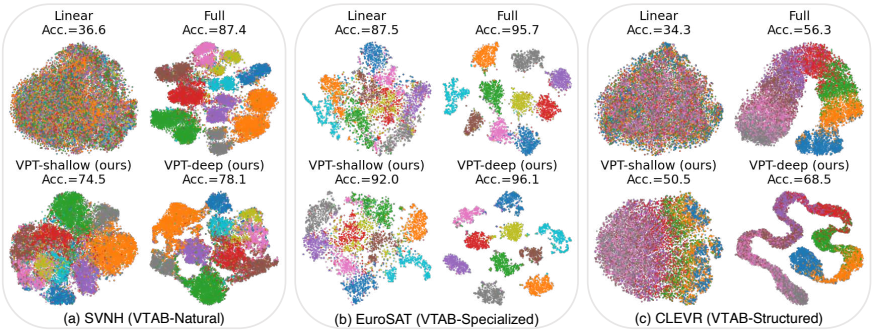

Fig. 9. t-SNE visualizations of the final `[CLS]` embedding $\mathbf{x}_N$ of 3 VTAB tasks from the `test` set, from Tab. 1. VPT could produce linearly separable features without updating backbone parameters

subgroup. All plots show that VPT-DEEP enables linearly separable representations while using less parameters than FULL. We also observe that extra tunable parameters for every Transformer layer (VPT-DEEP) improve the performance, compared to VPT-SHALLOW, which only inserts prompts for the first layer's input. Interestingly on Clevr/count (Fig. 9(c)), VPT-DEEP and FULL recover the underlying manifold structure of the task (counting objects in images *vs*. street number or landscape recognition), unlike VPT-SHALLOW and LINEAR.

Apply VPT to more vision tasks. We explore the feasibility of VPT beyond visual classification, by evaluating ADE20K [69] semantic segmentation task with a Transformer model, SETR-PUP [68]. It adds a standard ConvNet head to the ViT backbone to perform segmentation. The de-facto approach is still fully fine-tuning the pre-trained backbone together with the ConvNet head (FULL). We include two more protocols for comparison: only update the head lay-

Table 3. Semantic Segmentation: ADE20k [69] validation results with SETR [68] on ViT-L. The best mIoU scores among all methods but FULL are **bolded**. Results of fully fine-tuning a ResNet-101 [9] are included. SS/MS: single/multi-scale inference

Backbone	ViT-L/16					ResNet-101
Method	FULL [68]	HEAD ONLY	BIAS	VPT-DEEP	VPT+BIAS	FULL [9]
mIoU-SS	48.31	35.12	43.40	42.11	**44.04**	45.47
mIoU-MS	50.07	37.46	45.33	44.06	**45.63**	46.27
Tunable params (M)	318.31	13.18	13.46	13.43	15.79	63.0

Table 4. Different pre-trained objectives: MAE [23] and MoCo v3 [10] with a ViT-B backbone. For each method and each downstream task group, we report the average test accuracy score and `number of wins in (·)` compared to FULL. "Total params" denotes total parameters needed for all 24 downstream tasks. Best results among all methods except FULL are **bolded**

		MAE				MoCo v3			
	ViT-B/16 (85.8M)	**Total params**	**VTAB-1k** Natural	Specialized	Structured	**Total params**	**VTAB-1k** Natural	Specialized	Structured
	Total # of tasks		7	4	8		7	4	8
(a)	FULL	19.01×	59.29	79.68	53.82	19.01×	71.95	84.72	51.98
(b)	LINEAR	1.01×	18.87 (0)	53.72 (0)	23.70 (0)	1.01×	67.46 (4)	81.08 (0)	30.33 (0)
	PARTIAL-1	2.58×	**58.44** (**5**)	**78.28** (1)	47.64 (**1**)	2.58×	72.31 (**5**)	**84.58** (**2**)	47.89 (1)
(c)	BIAS	1.03×	54.55 (1)	75.68 (1)	**47.70** (0)	1.03×	72.89 (3)	81.14 (0)	**53.43** (**4**)
	ADAPTER	1.17×	54.90 (3)	75.19 (1)	38.98 (0)	1.22×	**74.19** (4)	82.66 (1)	47.69 (2)
(ours)	VPT-SHALLOW	1.01×	39.96 (1)	69.65 (0)	27.50 (0)	1.01×	67.34 (3)	82.26 (0)	37.55 (0)
	VPT-DEEP	1.04×	36.02 (0)	60.61 (1)	26.57 (0)	1.01×	70.27 (4)	83.04 (0)	42.38 (0)

ers (HEAD ONLY), update head layers and bias vectors in the backbone (BIAS). In Tab. 3, we report `val` mIoU results with and without multi-scale inference. Though parameter-efficient protocols could not compete with FULL, VPT is still comparable with BIAS. Notably, VPT offers competitive results to a fully fine-tuned state-of-the-art ConvNet model (DeepLab v3+ [9]), while tuning significantly less parameters (15M *vs*. 64M, respectively).

Apply VPT to more pre-training methods. In addition to the backbones pre-trained with labeled data, we experiment with two self-supervised objectives: MAE [23] and MoCo v3 [10]. Tab. 4 reports the results on VTAB-1k with ViT-B. We observe that both variants of VPT surpass LINEAR, yet the comparisons among other techniques are less conclusive. For MAE, other parameter-efficient methods, *e.g.*, PARTIAL-1, outperform both VPT and LINEAR. In the case of MoCo v3, VPT no longer holds the best performance, though it is still competitive with the others. This suggests that these two self-supervised ViTs are fundamentally different from the supervised ones in previous sections. Exactly why and how these differences arise remain open questions.

Apply VPT to ConvNets. We examine the idea of adding trainable parameters in the input space of ConvNets: padding both height and width by p

Table 5. Apply VPT to ConvNets: ResNet-50 and ConvNeXt-Base. For each method and each downstream task group, we report the average test accuracy score and `number of wins in (·)` compared to FULL. "Total params" denotes total parameters needed for all 19 downstream tasks. Best results among all methods except FULL are **bolded**

		ConvNeXt-Base (87.6M)				ResNet-50 (23.5M)			
		Total params	VTAB-1k			Total params	VTAB-1k		
			Natural	Specialized	Structured		Natural	Specialized	Structured
	Total # of tasks		7	4	8		7	4	8
(a)	FULL	19.01×	77.97	83.71	60.41	19.08×	59.72	76.66	54.08
(b)	LINEAR	1.01×	74.48 (5)	81.50 (0)	34.76 (**1**)	1.08×	63.75 (**6**)	77.60 (**3**)	30.96 (0)
	PARTIAL-1	2.84×	73.76 (4)	81.64 (0)	39.55 (0)	4.69×	64.34 (**6**)	**78.64** (2)	**45.78** (1)
	MLP-3	1.47×	73.78 (5)	81.36 (**1**)	35.68 (**1**)	7.87×	61.79 (**6**)	70.77 (1)	33.97 (0)
(c)	BIAS	1.04×	69.07 (2)	72.81 (0)	25.29 (0)	1.10×	63.51 (**6**)	77.22 (2)	33.39 (0)
(ours)	Visual-Prompt Tuning	1.02×	**78.48** (**6**)	**83.00** (**1**)	**44.64** (**1**)	1.09×	**66.25** (**6**)	77.32 (2)	37.52 (0)

learnable prompt pixels for the input image. Though this operation seems unconventional, we implement VPT this way given there is no obvious solution to add location-invariant prompts similar to the Transformer counterparts. In fact this approach has been explored before in the adversarial attack literature [16]. The value of p in our experiment is 2 orders of magnitude smaller than previous work: *e.g.*, 5 *vs.* 263. Most importantly, we cast this idea in the lens of transfer learning. See the Appendix for more discussion.

Tab. 5 presents the results for ConvNeXt-B [40] (pre-trained on ImageNet-21k) and ResNet-50 [24] (pre-trained on ImageNet-1k), respectively. VPT works well in a larger ConvNet backbone, ConvNeXt-B, offering accuracy gains over other sparse tuning protocols (`b`, `c`), and outperforming FULL on 8 out of 19 cases. The advantages of VPT, however, diminish with smaller ConvNet (ResNet-50), as there is no clear winner for all 19 VTAB-1k tasks.

6 Conclusion

We present Visual Prompt Tuning, a new parameter-efficient approach to leverage large vision Transformer models for a wide range of downstream tasks. VPT introduces task-specific learnable prompts in the input space, keeping the pre-trained backbone fixed. We show that VPT can surpass other fine-tuning protocols (often including full fine-tuning) while dramatically reducing the storage cost. Our experiments also raise intriguing questions on fine-tuning dynamics of vision Transformers with different pre-training objectives, and how to transfer to broader vision recognition tasks in an efficient manner. We therefore hope our work will inspire future research on how best to tap the potential of large foundation models in vision.

Acknowledgement. Menglin is supported by a Meta AI research grant awarded to Cornell University, Luming and Bharath is supported by NSF IIS-2144117, Serge is supported in part by the Pioneer Centre for AI, DNRF grant number P1. We would like to thank Alexander Rush, Yin Cui for valuable suggestions and discussion.

References

1. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016) 4
2. Bahng, H., Jahanian, A., Sankaranarayanan, S., Isola, P.: Visual prompting: Modifying pixel space to adapt pre-trained models. arXiv preprint arXiv:2203.17274 (2022) 3
3. Bao, H., Dong, L., Piao, S., Wei, F.: BEit: BERT pre-training of image transformers. In: ICLR (2022) 3
4. Ben Zaken, E., Goldberg, Y., Ravfogel, S.: BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 1–9. Association for Computational Linguistics, Dublin, Ireland (May 2022). https://doi.org/10.18653/v1/2022.acl-short.1, https://aclanthology.org/2022.acl-short.1 3, 6
5. Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258 (2021) 1
6. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) NeurIPS. vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020) 1, 3
7. Cai, H., Gan, C., Zhu, L., Han, S.: Tinytl: Reduce memory, not parameters for efficient on-device learning. NeurIPS **33**, 11285–11297 (2020) 2, 3, 6
8. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV. pp. 213–229. Springer (2020) 3
9. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV. pp. 801–818 (2018) 13
10. Chen*, X., Xie*, S., He, K.: An empirical study of training self-supervised vision transformers. In: ICCV (2021) 2, 3, 13
11. Conder, J., Jefferson, J., Jawed, K., Nejati, A., Sagar, M., et al.: Efficient transfer learning for visual tasks via continuous optimization of prompts. In: International Conference on Image Analysis and Processing. pp. 297–309. Springer (2022) 3
12. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009) 6
13. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019) 1, 3
14. Doersch, C., Gupta, A., Zisserman, A.: Crosstransformers: spatially-aware few-shot transfer. NeurIPS **33**, 21981–21993 (2020) 3
15. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth

16x16 words: Transformers for image recognition at scale. In: ICLR (2020) 1, 3, 4, 6
16. Elsayed, G.F., Goodfellow, I., Sohl-Dickstein, J.: Adversarial reprogramming of neural networks. In: ICLR (2019) 14
17. Feichtenhofer, C., Fan, H., Li, Y., He, K.: Masked autoencoders as spatiotemporal learners. arXiv preprint arXiv:2205.09113 (2022) 3
18. Ge, C., Huang, R., Xie, M., Lai, Z., Song, S., Li, S., Huang, G.: Domain adaptation via prompt learning. arXiv preprint arXiv:2202.06687 (2022) 3
19. Gebru, T., Krause, J., Wang, Y., Chen, D., Deng, J., Fei-Fei, L.: Fine-grained car detection for visual census estimation. In: AAAI (2017) 6
20. Girdhar, R., Carreira, J., Doersch, C., Zisserman, A.: Video action transformer network. In: CVPR. pp. 244–253 (2019) 3
21. Guo, D., Rush, A., Kim, Y.: Parameter-efficient transfer learning with diff pruning. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 4884–4896. Association for Computational Linguistics, Online (Aug 2021) 3
22. He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., Neubig, G.: Towards a unified view of parameter-efficient transfer learning. In: ICLR (2022) 3
23. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: CVPR. pp. 16000–16009 (2022) 3, 6, 13
24. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016) 1, 5, 14
25. Helber, P., Bischke, B., Dengel, A., Borth, D.: 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 **12**(7), 2217–2226 (2019) 11
26. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for nlp. In: ICML. pp. 2790–2799. PMLR (2019) 3, 6
27. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021) 3
28. Jia, M., Wu, Z., Reiter, A., Cardie, C., Belongie, S., Lim, S.N.: Exploring visual engagement signals for representation learning. In: ICCV (2021) 2
29. Jiang, Z., Xu, F.F., Araki, J., Neubig, G.: How can we know what language models know? Transactions of the Association for Computational Linguistics **8**, 423–438 (2020) 3
30. Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., Girshick, R.: Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In: CVPR (2017) 11
31. Ju, C., Han, T., Zheng, K., Zhang, Y., Xie, W.: Prompting visual-language models for efficient video understanding. arXiv preprint arXiv:2112.04478 (2021) 3
32. Khosla, A., Jayadevaprakash, N., Yao, B., Fei-Fei, L.: Novel dataset for fine-grained image categorization. In: First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition. Colorado Springs, CO (June 2011) 6
33. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 3045–3059. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (Nov 2021) 2, 3, 7

34. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7871–7880 (2020) 1
35. Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 4582–4597. Association for Computational Linguistics, Online (Aug 2021) 2, 3
36. Li, Y., Xie, S., Chen, X., Dollar, P., He, K., Girshick, R.: Benchmarking detection transfer learning with vision transformers. arXiv preprint arXiv:2111.11429 (2021) 3
37. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. arXiv preprint arXiv:2107.13586 (2021) 2, 3
38. Liu, X., Ji, K., Fu, Y., Du, Z., Yang, Z., Tang, J.: P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. arXiv preprint arXiv:2110.07602 (2021) 2, 3
39. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV (2021) 3, 5, 6, 9
40. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. CVPR (2022) 14
41. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008) 11
42. Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., Van Der Maaten, L.: Exploring the limits of weakly supervised pretraining. In: ECCV (2018) 2
43. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011 (2011) 11
44. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing. pp. 722–729. IEEE (2008) 6
45. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: ECCV. pp. 69–84. Springer (2016) 6
46. Pfeiffer, J., Kamath, A., Rücklé, A., Cho, K., Gurevych, I.: Adapterfusion: Non-destructive task composition for transfer learning. arXiv preprint arXiv:2005.00247 (2020) 3, 6
47. Pfeiffer, J., Rücklé, A., Poth, C., Kamath, A., Vulić, I., Ruder, S., Cho, K., Gurevych, I.: Adapterhub: A framework for adapting transformers. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020): Systems Demonstrations. pp. 46–54. Association for Computational Linguistics, Online (2020) 3, 6
48. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. pp. 8748–8763. PMLR (2021) 1, 3

49. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research **21**(140), 1–67 (2020) 1, 3
50. Rebuffi, S.A., Bilen, H., Vedaldi, A.: Learning multiple visual domains with residual adapters. NeurIPS **30** (2017) 3
51. Rebuffi, S.A., Bilen, H., Vedaldi, A.: Efficient parametrization of multi-domain deep neural networks. In: CVPR. pp. 8119–8127 (2018) 2
52. Sandler, M., Zhmoginov, A., Vladymyrov, M., Jackson, A.: Fine-tuning image transformers using learnable memory. In: CVPR. pp. 12155–12164 (2022) 3
53. Shin, T., Razeghi, Y., Logan IV, R.L., Wallace, E., Singh, S.: Autoprompt: Eliciting knowledge from language models with automatically generated prompts. arXiv preprint arXiv:2010.15980 (2020) 3
54. Strudel, R., Garcia, R., Laptev, I., Schmid, C.: Segmenter: Transformer for semantic segmentation. In: CVPR. pp. 7262–7272 (2021) 3
55. Van Horn, G., Branson, S., Farrell, R., Haber, S., Barry, J., Ipeirotis, P., Perona, P., Belongie, S.: Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In: CVPR. pp. 595–604 (2015) 6
56. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeurIPS **30** (2017) 3
57. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011) 6
58. Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.: Superglue: A stickier benchmark for general-purpose language understanding systems. NeurIPS **32** (2019) 3
59. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.: GLUE: A multi-task benchmark and analysis platform for natural language understanding. In: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. pp. 353–355. Association for Computational Linguistics, Brussels, Belgium (Nov 2018). https://doi.org/10.18653/v1/W18-5446, https://aclanthology.org/W18-5446 3
60. Wang, H., Zhu, Y., Adam, H., Yuille, A., Chen, L.C.: Max-deeplab: End-to-end panoptic segmentation with mask transformers. In: CVPR. pp. 5463–5474 (2021) 3
61. Wang, R., Chen, D., Wu, Z., Chen, Y., Dai, X., Liu, M., Jiang, Y.G., Zhou, L., Yuan, L.: Bevt: Bert pretraining of video transformers. In: CVPR. pp. 14733–14743 (2022) 3
62. Wang, Z., Zhang, Z., Lee, C.Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., Pfister, T.: Learning to prompt for continual learning. In: CVPR. pp. 139–149 (2022) 3
63. Yao, Y., Zhang, A., Zhang, Z., Liu, Z., Chua, T.S., Sun, M.: Cpt: Colorful prompt tuning for pre-trained vision-language models. arXiv preprint arXiv:2109.11797 (2021) 3
64. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? NeurIPS **27** (2014) 6
65. Zhai, X., Puigcerver, J., Kolesnikov, A., Ruyssen, P., Riquelme, C., Lucic, M., Djolonga, J., Pinto, A.S., Neumann, M., Dosovitskiy, A., et al.: A large-scale study of representation learning with the visual task adaptation benchmark. arXiv preprint arXiv:1910.04867 (2019) 6, 7

66. Zhang, J.O., Sax, A., Zamir, A., Guibas, L., Malik, J.: Side-tuning: a baseline for network adaptation via additive side networks. In: ECCV. pp. 698–714. Springer (2020) 2, 3, 6
67. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV. pp. 649–666. Springer (2016) 6
68. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: CVPR. pp. 6881–6890 (2021) 3, 12, 13
69. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. IJCV **127**(3), 302–321 (2019) 12, 13
70. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. arXiv preprint arXiv:2109.01134 (2021) 3
71. Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., He, Q.: A comprehensive survey on transfer learning. Proceedings of the IEEE **109**(1), 43–76 (2020) 3