NEURAL PROMPT SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

The size of vision models has grown exponentially over the last few years, especially after the emergence of Vision Transformer. This has motivated the development of parameter-efficient tuning methods, such as learning adapter layers or visual prompt tokens, which allow a tiny portion of model parameters to be trained whereas the vast majority obtained from pre-training are frozen. However, designing a proper tuning method is non-trivial: one might need to try out a lengthy list of design choices, not to mention that each downstream dataset often requires custom designs. In this paper, we view the existing parameter-efficient tuning methods as "prompt modules" and propose **Neural prOmpt seArcH (NOAH)**, a novel approach that learns, for large vision models, the optimal design of prompt modules through a neural architecture search algorithm, specifically for each downstream dataset. By conducting extensive experiments on over 20 vision datasets, we demonstrate that NOAH (i) is superior to individual prompt modules, (ii) has good few-shot learning ability, and (iii) is domain-generalizable.

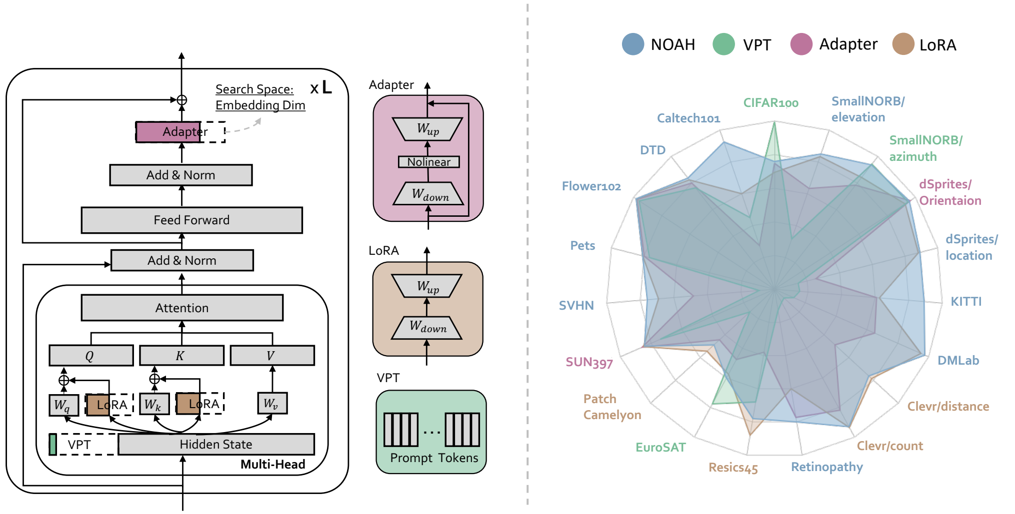
1 INTRODUCTION

The size of vision models has grown exponentially from tens of millions a few years ago (*e.g.*, ResNet (He et al., 2016)) to today's hundreds of millions (Dosovitskiy et al., 2020), or even billions (Brown et al., 2020; Devlin et al., 2018; Reed et al., 2022), for Transformers (Vaswani et al., 2017). Such an increase can cause a number of problems to transfer learning (Houlsby et al., 2019; Jia et al., 2022), and the first and foremost is that fine-tuning becomes more difficult as large model size can easily lead to overfitting in a typical-sized dataset, let alone the increase of compute and storage costs.

Recently, there is a growing interest in developing parameter-efficient tuning methods (Houlsby et al., 2019; Hu et al., 2021; Jia et al., 2022). The key idea is to insert a tiny trainable module to a large pre-trained model and only adjust its parameters by optimizing some task-specific losses like the cross-entropy for classification problems. The most representative methods are Adapter (Houlsby et al., 2019), Low-Rank Adaptation (LoRA) (Hu et al., 2021), and Visual Prompt Tuning (VPT) (Jia et al., 2022). As exemplified in Fig. 1(a), Adapter is a bottleneck-shaped neural network appended to a network block's output; LoRA is a "residual" layer consisting of rank decomposition matrices; VPT prepends additional tokens to the input of a Transformer block, which can be seen as adding learnable "pixels."

By evaluating the three parameter-efficient tuning methods on a commonly-used transfer learning benchmark, *i.e.*, VTAB-1k (Zhai et al., 2019), we identify a couple of critical issues. *First*, none of the three methods performs consistently well on all datasets, as illustrated in Fig. 1(b). For instance, when it comes to scene structure understanding tasks, VPT outperforms Adapter and LoRA on SmallNORB/azimuth (LeCun et al., 2004), but its performance plunges on SmallNORB/elevation (LeCun et al., 2004) and Clevr/count (Johnson et al., 2017), which is largely behind the two competitors. The results suggest that, for a specific dataset, one needs to perform an extensive evaluation on different tuning methods in order to identify the most suitable one. *Second*, performance is found to be sensitive to the selection of model parameters, such as Adapter's feature dimension or the token length in VPT—this is also observed by Jia et al. (2022) that the optimal token length in VPT varies from 1 to 200 on different datasets.

In this work, we view the existing parameter-efficient tuning methods as *prompt modules* and propose to *automatically search for the optimal prompt design from data via a neural architecture search*



(a) (b)

Figure 1: Our approach, neural prompt search, or NOAH for short, subsumes three representative parameter-efficient tuning methods (i.e., Adapter, LoRA and VPT) and learns from data the optimal design through neural architecture search (a). The approach is motivated by the observation that none of the three individuals shows dominance on the VTAB-1k benchmark (b). The colors of the datasets' names indicate which method performs the best. Clearly, NOAH is the best overall approach. It achieves the best performance on 10 out of 19 datasets.

(NAS) algorithm. Specifically, we introduce the concept of Neural prOmpt seArcH (NOAH) for large vision models, particularly those equipped with the Transformer block (Dosovitskiy et al., 2020). The search space is constructed by subsuming Adapter (Houlsby et al., 2019), LoRA (Hu et al., 2021) and VPT (Jia et al., 2022) into each Transformer block, as depicted in Fig. 1(a). The specific model parameters, including the feature dimension for Adapter and LoRA and the token length for VPT, are determined by a one-shot NAS algorithm.

We conduct extensive experiments on VTAB-1k (Zhai et al., 2019), which is composed of 19 diverse vision datasets and covers a wide spectrum of visual domains like objects, scenes, textures and satellite imagery. The results show that NOAH significantly outperforms the individual prompt modules on 10 out of 19 datasets while the performance on the remaining is highly competitive (see Fig. 1(b) for an overview of the results). We also evaluate on few-shot learning and domain generalization where the results also confirm the superiority of NOAH to the hand-crafted prompt modules.

Our contributions are summarized as follows. (i) We present a systematic study of three representative prompt modules and expose some critical issues associated with performance and efficiency. (ii) A novel concept, neural prompt search, is proposed to address the challenge of hand-engineering prompt modules. (iii) An efficient NAS-based implementation of NOAH is provided. (iv) We demonstrate that NOAH is better than individual prompt modules in downstream transfer learning, few-shot learning, and domain generalization. The models and code will be released.

2 NEURAL PROMPT SEARCH

2.1 BACKGROUND

Vision Transformer We first briefly review Vision Transformer (ViT) (Dosovitskiy et al., 2020), to which our approach is mainly applied. ViT consists of alternating blocks of multihead self-attention (MSA) and multi-layer perceptron (MLP). Given an input sequence $x \in \mathbb{R}^{N \times D}$ where N denotes the token length and D is the embedding dimension, MSA first maps x to queries $Q \in \mathbb{R}^{N \times d}$, keys $K \in \mathbb{R}^{N \times d}$ and values $V \in \mathbb{R}^{N \times d}$ using three projection matrices, $W_q \in \mathbb{R}^{D \times d}$, $W_k \in \mathbb{R}^{D \times d}$ and $W_v \in \mathbb{R}^{D \times d}$, respectively, where d means the hidden dimension. Then, MSA computes the weighted sums over the values based on the self-attention between the queries and keys,

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d}})V, \tag{1}$$

where $\frac{1}{\sqrt{d}}$ is a scaling factor.

Below we briefly review the three representative—and top-performing—parameter-efficient tuning methods, *i.e.*, Adapter (Houlsby et al., 2019), LoRA (Hu et al., 2021) and VPT (Jia et al., 2022), which will be incorporated into our search space. An illustration of these three methods can be found in Fig. 1(a). Note that VPT has been studied for vision models while Adapter and LoRA have only been studied for language models.

Adapter is essentially a bottleneck-like neural network consisting of a down-sample layer $W_{down} \in \mathbb{R}^{d\times r}$ and an up-sample layer $W_{up} \in \mathbb{R}^{r\times d}$, where r denotes the down-sampled dimension. A non-linear activation function $\phi(\cdot)$, such as ReLU, is inserted in-between. The computation can be formulated as

$$h' = \phi(hW_{down})W_{up}, \tag{2}$$

where $h \in \mathbb{R}^{N\times d}$ is a normalized output of the MLP in a Transformer block.

LoRA aims to update the two projection layers, W_q (for queries) and W_k (for keys), in an indirect way by optimizing their rank-decomposed changes, $\triangle W_q = W_q^{down}W_q^{up}$ and $\triangle W_k = W_k^{down}W_k^{up}$, where $W_{q/k}^{down} \in \mathbb{R}^{D\times r}$ and $W_{q/k}^{up} \in \mathbb{R}^{r\times d}$ (r is the down-projection dimension). For a specific input x, we have

$$Q = xW_q + s \cdot xW_q^{down}W_q^{up}, \quad K = xW_k + s \cdot xW_k^{down}W_k^{up}, \tag{3}$$

where s is a fixed scaling parameter for modulating the updates.

Visual Prompt Tuning (VPT) prepends a set of learnable tokens to the input of a Transformer block, which can be viewed as adding some learnable pixels in the input space. We investigate the best-performing version, VPT-Deep, which applies prompt tuning to multiple layers (Jia et al., 2022). We call this module VPT for brevity hereafter and formulate it in mathematical terms below. A typical input $x \in \mathbb{R}^{N\times D}$ to a Transformer block contains a learnable class token [CLS] of D-dimension and a sequence of image patch embeddings $\mathrm{E} = \{\mathrm{e}_i|\mathrm{e}_i \in \mathbb{R}^D, i = 1, ..., N-1\}$ where the positional embeddings are omitted. VPT adds m learnable tokens, $\mathrm{P} = \{\mathrm{p}_k|\mathrm{p}_k \in \mathbb{R}^D, k = 1, ..., m\}$, to x, which then becomes

$$x = [\mathrm{CLS}, \mathrm{P}, \mathrm{E}]. \tag{4}$$

2.2 PROMPT SEARCH ALGORITHM

As discussed, none of the individual parameter-efficient tuning methods, or *prompt modules* called in this paper, shows dominance in the transfer learning benchmark. Our approach, neural prompt search (NOAH), incorporates Adapter (Houlsby et al., 2019), LoRA (Hu et al., 2021) and VPT (Jia et al., 2022) into each Transformer block and learns the design that best suits a dataset through neural architecture search (NAS). Specifically, we employ a one-shot NAS algorithm, AutoFormer (Chen et al., 2021), for prompt module search. Our supernet is a ViT-like model composed of 12 Transformer blocks (layers). Below we detail the search space and how the search is done.

Search Space As shown in Fig. 1(a), we embed the three prompt modules into each Transformer block following the guidelines proposed in the original work (Houlsby et al., 2019; Hu et al., 2021; Jia et al., 2022). Concretely, we install VPT in the input position, add LoRA alongside the two projection matrices as residuals, and insert Adapter after the normalized output of the MLP. The search space mainly contains the model parameters associated with the three prompt modules. Specifically, each prompt module has two sets of parameters to search from: (i) the embedding dimension $\in \{1, 5, 10, 50, 100\}$; (ii) the depth $\in \{3, 6, 9, 12\}$ (zero-based indexing is adopted.) A depth means up to which layer a module is applied, *e.g.*, depth $= 3$ for VPT means layers 0, 1 and 2 have VPT installed while the remaining layers, 3 to 11, do not have VPT. For VPT, the embedding dimension means the token length whereas for Adapter and LoRA, the embedding dimension means the down-sampled dimension, *i.e.*, r.

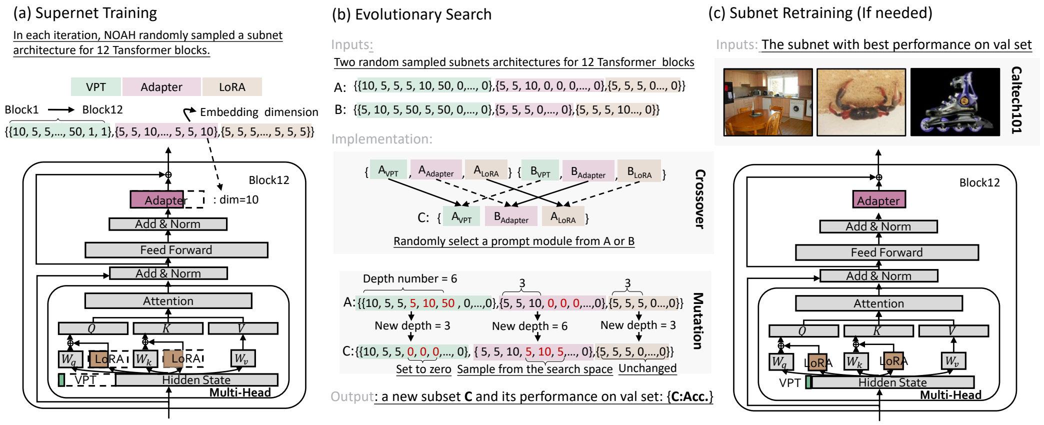


Figure 2: **Illustration of the three stages in NOAH**. For each dataset, NOAH finds the optimal designs of the prompt modules through supernet training and evolutionary search. NOAH retrains the subnet, which performs the best on the validation set, for achieving better performance if needed.

Supernet Training The supernet, as mentioned, has 12 Transformer layers, each containing the three prompt modules with full embedding dimension, *i.e.*, 100. During each forward pass, a subnet is randomly sampled from the supernet for training. Specifically, for *each* prompt module, a depth is first sampled from $\{3, 6, 9, 12\}$ to determine which layers should have the module. Then, for *each* layer within the depth range, an embedding dimension is chosen from $\{1, 5, 10, 50, 100\}$, all with a uniform probability. Note that only the prompt modules' parameters are learned while the pre-trained model is kept fixed. AutoFormer (Chen et al., 2021) allows the weights in each prompt module to be entangled during training, meaning that different weights are maximally shared, *e.g.*, in a VPT module, if 100 tokens are selected for training, the previously trained tokens, such as 50, will be reused and trained together with other 50 tokens. This way, as suggested in AutoFormer, leads to faster convergence and low memory cost. We add more details of this step in Supplementary.

Evolutionary Search After the supernet is trained, evolutionary search is conducted to obtain the optimal subnet architecture under a parameter size limit (Chen et al., 2021). Specifically, we first select K random architectures, from which the top k architectures (with the best performance) are used as parents to produce the next generation through crossover and mutation. For crossover, two candidates are randomly chosen and crossed to produce a "child" architecture. For mutation, a candidate mutates its prompt module design with a probability. See Fig. 2 for an illustration. We add more details of this step in Supplementary.

3 EXPERIMENTS

In this section, we mainly address the following questions: (i) Is NOAH better than the individual prompt modules? (ii) Can NOAH work in a few-shot setting? (iii) Are models learned by NOAH robust to domain shift? The answers are discussed in Sec. 3.1, 3.2 and 3.3, respectively. We also conduct some analyses in Sec. 3.4 to have a deeper understanding of NOAH, such as what a subnet looks like and whether it is transferable beyond the dataset in which the architecture was found.

Baselines The main competitors are Adapter (Houlsby et al., 2019), LoRA (Hu et al., 2021) and VPT (Jia et al., 2022). Among them, only VPT is specifically designed for vision models while the other two are originally developed for language models. We also compare two common fine-tuning methods on the VTAB-1k benchmark: full tuning (Full) and linear probing (Linear). Full simply tunes the entire model parameters whereas Linear freezes the pre-trained part and only adjusts the newly added linear classification layer.[1] It is worth mentioning that Full has been considered as a strong baseline in existing studies (Jia et al., 2022; Houlsby et al., 2019).

---

[1] Note that all methods have a new linear classification layer to learn.

Table 1: **Full results on the VTAB-1k benchmark**. The first block contains conventional tuning methods while the second block contains parameter-efficient tuning methods, which is the main focus in this paper. NOAH achieves the best overall performance, which is 1% higher on average than the individual prompt modules.

		Natural							Specialized				Structured								
	# param (M)	Cifar100	Caltech101	DTD	Flower102	Pets	SVHN	Sun397	Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DMLab	KITTI-Dist	dSpr-Loc	dSpr-Ori	sNORB-Azim	sNORB-Ele	Average
Full	85.8	**68.9**	**87.7**	**64.3**	87.2	**86.9**	**87.4**	38.8	**79.7**	**95.7**	**84.2**	73.9	**56.3**	**58.6**	**41.7**	**65.5**	**57.5**	**46.7**	**25.7**	**29.1**	**68.9**
Linear	0.04	64.4	85.0	63.2	**97.0**	86.3	36.6	**51.0**	78.5	87.5	68.5	**74.0**	34.3	30.6	33.2	55.4	12.5	20.0	9.6	19.2	57.6
VPT	0.64	**78.8**	90.8	65.8	98.0	88.3	78.1	49.6	81.8	**96.1**	83.4	68.4	68.5	60.0	46.5	72.8	73.6	47.9	**32.9**	37.8	72.0
Adapter	0.16	69.2	90.1	68.0	98.8	89.9	82.8	**54.3**	84.0	94.9	81.9	75.5	80.9	65.3	48.6	78.3	74.8	**48.5**	29.9	41.6	73.9
LoRA	0.29	67.1	91.4	69.4	98.8	**90.4**	85.3	54.0	**84.9**	95.3	**84.4**	73.6	**82.9**	**69.2**	49.8	78.5	75.7	47.1	31.0	44.0	74.5
NOAH	0.43	69.6	**92.7**	**70.2**	**99.1**	**90.4**	**86.1**	53.7	84.4	95.4	83.9	**75.8**	82.8	68.9	**49.9**	**81.7**	**81.8**	48.3	32.8	**44.2**	**75.5**

Implementation Details We keep the training parameters identical across all experiments throughout this paper. ViT-B/16 (Dosovitskiy et al., 2020) pre-trained on ImageNet-22K (Deng et al., 2009) is used as the base model, which is strong enough so the results are fair and convincing. The supernet for NOAH is trained for 500 epochs and the ultimate subnet is trained for 100 epochs— note that "subnet" means the prompt modules/architectures. Since the AutoFormer (Chen et al., 2021) algorithm allows a subnet to be used without retraining, we demonstrate later that the subnet found by NOAH without retraining is also comparable to the retrained one. The evolutionary search in NOAH takes 5 epochs in total and each step of random pick/crossover/mutation produces 50 new subnets. The probability for crossover and mutation is set to 0.2, which follows AutoFormer (Chen et al., 2021). The individual prompt modules, *i.e.*, Adapter (Houlsby et al., 2019), LoRA (Hu et al., 2021) and VPT (Jia et al., 2022), are constructed using the best recipes suggested by the original papers (also trained for 100 epochs; see the Supplementary for more details). The parameter sizes for Adapter, LoRA and VPT are 0.16M, 0.29M and 0.64M, respectively. For fair comparison, we set the upper-limit of parameter size of the final subnet in NOAH to 0.64M so the resulting size would be comparable to the baselines. More implementation details including image augmentation and other hyper-parameters are provided in the Supplementary.

3.1 EXPERIMENTS ON VTAB-1K

Datasets We choose the VTAB-1k (Zhai et al., 2019) benchmark to evaluate the transfer learning performance of our approach. VTAB-1k consists of 19 vision datasets, which are clustered into three groups: Natural, Specialized and Structured. The Natural group contains natural images that are captured by standard cameras and cover a broad spectrum of concepts including generic, fine-grained and abstract objects. The Specialized group contains images captured by specialist equipment for remote sensing (like aerial images) and medical purposes. The Structured group is designed specifically for scene structure understanding, such as object counting, depth prediction and orientation prediction. Each dataset in VTAB-1k contains 1,000 labeled examples, which are split into a `train` (80%) and a `val` (20%) set (the latter is used for hyper-parameter tuning), while the test data comes from the original test set. The final model used for evaluation is trained using the full 1,000 examples in each dataset. Top-1 classification accuracy is used as the performance measure.

Results Table 1 presents the full results on the VTAB-1k benchmark. A high-level summary is shown earlier in Fig. 1(b). The average performance within each group is summarized in Fig. 3. We have the following observations.

Observation 1: Overall, NOAH is the best parameter-efficient tuning method. First and foremost, we demonstrate that searching for the optimal combination of the individual prompt modules works the best. This is evidenced by the 1% average gain over the strongest prompt module, *i.e.*, LoRA. Given the diversity of the benchmark, the 1% average gain can be considered to be significant. It is also worth mentioning that Adapter was previously proved to be the best-performing prompt module in NLP (Mao et al., 2021), but in our study for computer vision tasks, LoRA takes over the seat. This further confirms that search is a better option than hand-engineering in practice.

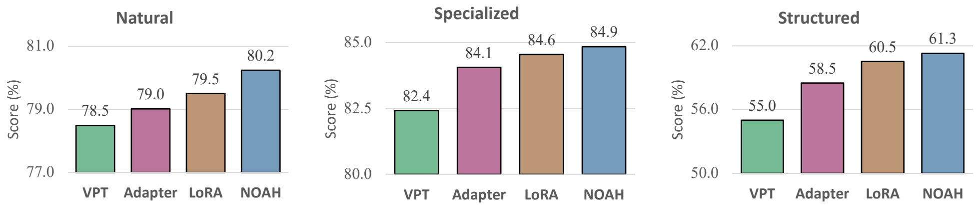


Figure 3: **Group-wise average results on VTAB-1k**. NOAH performs the best in the Natural and Structured groups while its performance in the Specialized group is similar to that of LoRA—but NOAH does not require a manual search over the architecture and hyper-parameters.

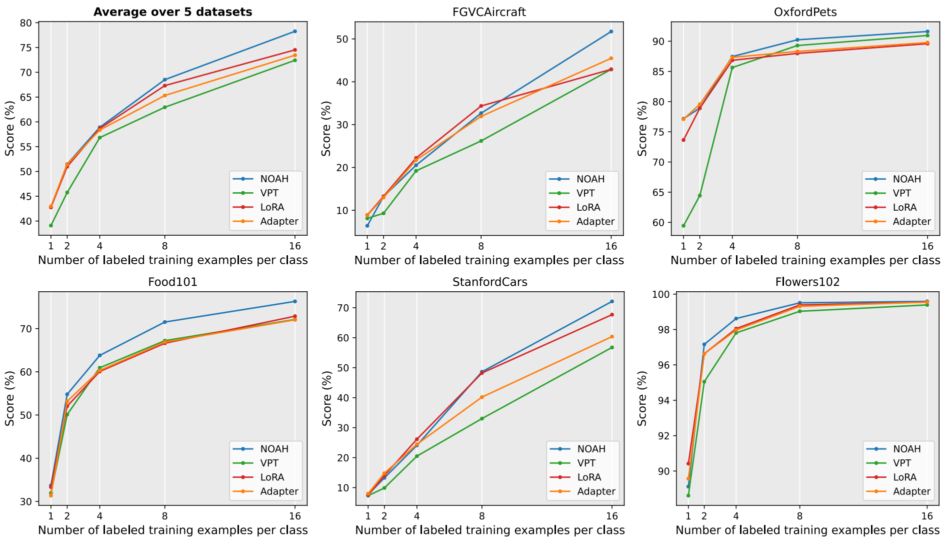


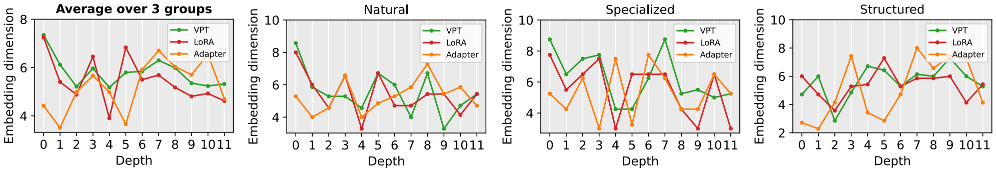
Figure 4: **Results of few-shot learning on five fine-grained visual recognition datasets**. NOAH beats the individual modules on average.

Observation 2: NOAH slightly dims in the Specialized group. The results suggest that NOAH's weakness seems to be in the Specialized tasks where the individual modules achieve the on-par performance: NOAH's results are not too far from those of the competitors, *e.g.*, NOAH's 84.9 vs LoRA's 84.6 on average. And while NOAH is superior on a Retinopathy, it lags on the other datasets, especially on the EuroSAT. Since the individual modules require a manual search over architecture and hyper-parameters, NOAH is more compelling.

3.2 EXPERIMENTS ON FEW-SHOT LEARNING

Datasets We choose five fine-grained visual recognition datasets, which include Food101 (Bossard et al., 2014), OxfordFlowers102 (Nilsback & Zisserman, 2006), StandfordCars (Krause et al., 2013), OxfordPets (Parkhi et al., 2012), and FGVCAircraft (Maji et al., 2013). The categories in these datasets cover a wide range of visual concepts closely related to our daily life: food, plant, vehicle and animal. We follow existing studies (Zhou et al., 2022b; Radford et al., 2021) to evaluate on 1, 2, 4, 8 and 16 shots, which are sufficient for observing the trend.

Results The results are summarized in Fig. 4. In terms of the average performance, we can observe that: (i) In the low-data regime like 1 or 2 shots, NOAH, LoRA and Adapter perform similarly but VPT largely lags behind; (ii) NOAH shows clear dominance when more shots become available, *e.g.*, with 16 shots the gap between NOAH and the runner-up is around 2%. By looking at the individual graphs, we can see that none of the individual prompt modules performs consistently well on all datasets, which, again, justifies that search is better than hand-engineering.



(a) (b) (c) (d)

Figure 5: **Average subnets (architectures) for the three groups in VTAB-1k**. Adapter and LoRA tend to live in deep layers while VPT is found nearly in all depths. The demands for VPT (indicated by the embedding dimension) differ in different groups. The co-existence of the three modules, especially in deep layers, serves as strong evidence of their complementarity, and such a synergy is difficult to obtain by hand-engineering.

3.3 EXPERIMENTS ON DOMAIN GENERALIZATION

Datasets Since domain shift is ubiquitous in real-world applications (Zhou et al., 2021a), we are interested to know how our search-based approach compares with the individual prompt modules in terms of domain generalization ability. Following prior studies (Zhou et al., 2022b;a), we first train a model on ImageNet (Deng et al., 2009) (using 16 shots per category) and then directly test it on four other variants of ImageNet that undergo different types of domain shift. Specifically, the test datasets include (i) ImageNetV2 (Recht et al., 2019), which is collected from different sources than ImageNet but following the same collection protocol, (ii) ImageNet-Sketch (Wang et al., 2019), which is composed of sketch images of the same 1,000 classes in ImageNet, (iii) ImageNet-A (Hendrycks et al., 2021b), which contains adversarially-filtered images, (iv) ImageNet-R (Hendrycks et al., 2021a), which is a rendition of ImageNet. Both ImageNet-A and -R have 200 classes derived from a subset of ImageNet's 1000 classes. All results are averaged over three random seeds.

Table 2: **Results on domain generalization**. NOAH is significantly better than the individual prompt modules on the four domain-shifted datasets.

	Source	**Target**			
	ImageNet	-V2	-Sketch	-A	-R
Adapter	70.5	59.1	16.4	5.5	22.1
VPT	70.5	58.0	18.3	4.6	23.2
LoRA	70.8	59.3	20.0	6.9	23.3
NOAH	**71.5**	**66.1**	**24.8**	**11.9**	**28.5**

Results Table 2 compares NOAH with the three individual prompt modules. On ImageNet, which is the source dataset, the gap between NOAH and the individual modules is small, which is about 1%. However, on the four test datasets, NOAH demonstrates significantly stronger robustness than the baselines: over 6.8%, 4.8%, 5% and 5.2% improvements on -V2, -Sketch, -A and -R, respectively. The results, together with those from previous subsections, justify that our search-based approach is superior to the individual prompt modules.

3.4 FURTHER ANALYSIS

Architecture of Subnet A key question to answer is: how does NOAH's subnet, *i.e.*, the ultimate architecture, look like. To make the results convincing, we visualize the average architecture found within each group of VTAB-1k, as well as the global average over all datasets, in Fig. 5. The x-axis represents the network depth while the y-axis represents the embedding dimension. An intriguing observation is that Adapter and LoRA, *across all groups* (Fig. 5(a)), mainly appear in deep layers with the embedding dimension larger than four and reduced to zero in some shallow layers. In contrast, VPT can be found nearly in all depths (layers), but the dimensions vary significantly in different groups, which indicates different demands for VPT. For instance, in the Natural group (Fig. 5(b)), shallow layers need more VPT modules; but in the Structured group (Fig. 5(d)), deep layers need more VPT modules. Moreover, the co-existence of the three modules, especially in deep layers, suggests that they are complementary to each other—*such a synergy is difficult to obtain by manual design*. In summary, the observed high variances in the module designs strongly indicate that search

Table 3: **Computational Cost**. We illustrate the computational cost of NOAH and other individual prompt modules. VPT, Adapter*, and LoRA* on each dataset are the best ones over grid search of search space. The computational cost for them is, therefore $1.2 * 6 = 7.2$. Param. of VPT, Adapter*, and LoRA* are the average architecture parameters of their best architecture on each dataset.

	Time (GPU hour/dataset)				Param. (M)	Top-1 Accuracy			
	Supernet	Search	Retrain	Total		Caltech101	EuroSAT	KITTI	Average
VPT	-	-	-	7.2	0.8	90.8	**96.1**	72.8	86.6
Adapter	-	-	-	1.2	0.2	90.1	94.9	78.3	87.7
Adapter*	-	-	-	7.2	1.5	91.9	95.7	80.6	89.4
LoRA	-	-	-	1.2	0.3	91.4	95.3	78.5	88.4
LoRA*	-	-	-	7.2	3.0	92.0	95.8	80.5	89.4
NOAH	3.1	2.0	1.2	6.3	0.5	**92.7**	95.4	**81.7**	**89.9**

is much more efficient than hand-engineering when it comes to developing parameter-efficient tuning methods.

Computational Cost In Table. 3, we provide the GPU hour/dataset to compare the computation cost of different methods and their variants. To gain more insights, we show their performance on three datasets selected from VTAB-Natural, VTAB-Specified, and VTAB-Structural. We observe that (i) compared to the VPT, Adapter, and LoRA, NOAH achieves at least 1.5% average accuracy gain on these three datasets at manageable computational overhead (1.2 v.s. 6.3). We would like to emphasize how to accelerate NOAH further is worthwhile future work. (ii) NOAH achieves the best performance (89.9%) with the lowest storage cost (0.5MB), compared to the best individual prompt modules obtained by manually searching. Specifically, we conduct a grid search on the Adapter, LoRA, following the grid search strategy of VPT (Jia et al., 2022), obtaining Adapter* and LoRA*. Under a similar computational budget (7.2 v.s. 6.3), the performance of Adapter* and LoRA* still lags behind NOAH by 0.5% with 3 and 6 times more parameters than NOAH. It demonstrates the superiority of the neural prompt search over manual search.

With vs Without Retraining Thanks to the weight entanglement strategy in AutoFormer (Chen et al., 2021), the subnet extracted from the supernet can be directly deployed for use without retraining. To verify if such a rule also applies to NOAH, we compare the subnets with and without retraining on VTAB-1k. The results averaged over each group are shown in Table 4 where we observe that NOAH without retraining (denoted as *inherited*) is still competitive: the inherited version still outperforms the VPT and Adapter. The results suggest that the retraining cost can be safely removed without incurring any significant loss.

Table 4: **With vs without retraining NOAH's subnets**. The results show that there is no significant difference between them, suggesting that retraining can be safely removed in practice if the compute resource is limited.

	Nat.	Spe.	Str.	Average
VPT	78.5	82.4	55.0	72.0
Adapter	79.0	84.1	58.5	73.9
LoRA	79.5	84.6	60.5	74.5
NOAH (*inherited*)	80.0	84.7	60.3	75.1
NOAH (*retrained*)	**80.2**	**84.9**	**61.3**	**75.5**

4 RELATED WORK

Parameter-Efficient Tuning A recent trend in transfer learning is to develop parameter-efficient tuning methods (Jia et al., 2022; Houlsby et al., 2019; Hu et al., 2021; He et al., 2021; Zhong et al., 2021; Zhou et al., 2022b), which is spurred by the rapid increase in model size. Existing methods can be generally divided into two groups. The first group fine-tunes a small portion of the internal parameters, such as biases (Zaken et al., 2021). The second group adds tiny learnable modules like Adapter (Houlsby et al., 2019) or LoRA, which is more relevant to our research and thus the focus here. Adapter (Houlsby et al., 2019) and LoRA (Hu et al., 2021) essentially share similar architectures—both look like a bottleneck—but are installed at different places: Adapter is often installed at the output of a block while LoRA is treated as residuals to the projection matrices in a

Transformer (Vaswani et al., 2017) block. It is worth noting that these methods are first studied in natural language processing (NLP) since pre-trained language models (Devlin et al., 2018; Brown et al., 2020) typically have an enormous parameter size that reaches the billion level. Another popular design in NLP is prompt learning (Zhong et al., 2021; Li & Liang, 2021; Lester et al., 2021), which turns some text prompt tokens into learnable vectors. Such an idea has recently been applied to vision-language models (Zhou et al., 2022b;a;b; Lu et al., 2022) and is also the source of inspiration for the recently proposed VPT (Jia et al., 2022), which adds learnable "pixels" to the input of ViT (Dosovitskiy et al., 2020). However, as discussed in recent works (Chen et al., 2022; Ding et al., 2022) in NLP and from our observation in computer vision, the optimal parameter-efficient tuning methods may differ as to the model and dataset change.

More relevant to our work are those trying to unify different parameter-efficient tuning methods (He et al., 2021; Mao et al., 2021). He et al. (2021) build a connection between Adapter and prompt learning and cast the problem into the learning of a modification vector, which leads to a unified view and a stronger baseline. UNIPELT (Mao et al., 2021) is another unified framework, which subsumes several prompt modules in a block and learns a set of gating functions to selectively activate them. Our work differs from these studies in two crucial ways: (i) we target *computer vision* problems whereas the previous studies (He et al., 2021; Mao et al., 2021) focus on NLP; (ii) we unify prompt modules from the NAS perspective with a much more *fine-grained* control over the model hyper-parameters (*e.g.*, token length and embedding dimension). This allows our model to be deployed in a resource-constrained environment. In the future, we plan to apply our approach to NLP.

Neural Architecture Search Neural architecture search (NAS) consists of two crucial components: search space and search algorithm. A search space can subsume various designs of how neurons are connected (Zoph et al., 2018), diverse combinations of model hyper-parameters (Zoph & Le, 2016), or different arrangements of specific modules like normalization layers (Zhou et al., 2021b). When it comes to the search algorithm part, the community has witnessed significant advances: from costly methods like reinforcement learning (Zoph & Le, 2016) or evolutionary search (Real et al., 2017) to more efficient ones based on weight-sharing (Pham et al., 2018) or differentiable optimization (Liu et al., 2018). The most relevant work to ours is AutoFormer (Chen et al., 2021), which is a NAS method focusing on Transformer (Vaswani et al., 2017) models. AutoFormer features a weight entanglement strategy to allow different subnets sampled from a big supernet to share weights among each other. Our work leverages AutoFormer to solve the problem of engineering parameter-efficient tuning methods, which we hope can inspire future work to address efficient transfer learning.

5 DISCUSSION, LIMITATION AND FUTURE WORK

Our research presents timely studies on how some recently-proposed parameter-efficient tuning methods, or prompt modules, fare in computer vision problems. Our studies expose a critical issue that, for any specific downstream dataset, hand-designing an optimal prompt module is extremely challenging. More importantly, we for the first time solve the problem from a NAS perspective and demonstrate the potential of our search-based approach in terms of downstream transfer learning performance, the ability to work in low-data regimes, and robustness to domain shift, which is ubiquitous in real-world data (Zhou et al., 2021a).

Our studies also unveil some intriguing phenomena. In particular, we find that the ultimate subnet exhibits different architectural patterns for the three prompt modules across datasets of different natures. Since neural networks' features, as often suggested (Zeiler & Fergus, 2014), progress from low-level visual primitives in bottom layers to high-level abstractions in top layers, the aforementioned findings entail that different prompt modules work best for features at different levels. We hope such findings and insights can inspire future work on designing more advanced prompt modules.

In terms of limitations, NOAH requires additional training for the supernet, which inevitably increases the development cost. Moreover, as suggested by the few-shot learning results, NOAH's advantages become clearer when more labeled images are available. In other words, NOAH would require more labels to unleash its full power in practice. For future work, we plan to dig deeper into the mechanisms behind NOAH for better interpretation of the intriguing results and apply NOAH to broader application domains beyond computer vision, such as NLP.

REFERENCES

Charles Beattie, Joel Z Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, et al. Deepmind lab. *arXiv preprint arXiv:1612.03801*, 2016. 15

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision (ECCV)*, pp. 446–461. Springer, 2014. 6

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems (NeuIPS)*, 33:1877–1901, 2020. 1, 9

Guanzheng Chen, Fangyu Liu, Zaiqiao Meng, and Shangsong Liang. Revisiting parameter-efficient tuning: Are we really there yet? *arXiv preprint arXiv:2202.07962*, 2022. 9

Minghao Chen, Houwen Peng, Jianlong Fu, and Haibin Ling. Autoformer: Searching transformers for visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12270–12280, 2021. 3, 4, 5, 8, 9, 13, 14

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255. Ieee, 2009. 5, 7, 14

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1, 9

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint arXiv:2203.06904*, 2022. 9

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2, 5, 9

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 15

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*, 2021. 8, 9

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. 1

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8340–8349, 2021a. 7

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15262–15271, 2021b. 7

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning (ICML)*, pp. 2790–2799. PMLR, 2019. 1, 2, 3, 4, 5, 8, 14

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 1, 2, 3, 4, 5, 8, 14

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. *arXiv preprint arXiv:2203.12119*, 2022. 1, 2, 3, 4, 5, 8, 9, 14, 15

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2901–2910, 2017. 1, 15

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013. 6

Yann LeCun, Fu Jie Huang, and Leon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pp. II–104. IEEE, 2004. 1

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 9

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 9

Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018. 9

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 14

Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 9

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 6

Yuning Mao, Lambert Mathias, Rui Hou, Amjad Almahairi, Hao Ma, Jiawei Han, Wen-tau Yih, and Madian Khabsa. Unipelt: A unified framework for parameter-efficient language model tuning. *arXiv preprint arXiv:2110.07577*, 2021. 5, 9

M-E Nilsback and Andrew Zisserman. A visual vocabulary for flower classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pp. 1447–1454. IEEE, 2006. 6

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3498–3505. IEEE, 2012. 6

Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. Efficient neural architecture search via parameters sharing. In *International conference on machine learning (ICML)*, pp. 4095–4104. PMLR, 2018. 9

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pp. 8748–8763. PMLR, 2021. 6

Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Jie Tan, Quoc V Le, and Alexey Kurakin. Large-scale evolution of image classifiers. In *International Conference on Machine Learning (ICML)*, pp. 2902–2911. PMLR, 2017. 9

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning (ICML)*, pp. 5389–5400. PMLR, 2019. 7

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022. 1

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems (NeuIPS)*, 30, 2017. 1, 9

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems (NeuIPS)*, 32, 2019. 7

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*, 2021. 8

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision (ECCV)*, pp. 818–833. Springer, 2014. 9

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019. 1, 2, 5, 14

Zexuan Zhong, Dan Friedman, and Danqi Chen. Factual probing is [mask]: Learning vs. learning to recall. *arXiv preprint arXiv:2104.05240*, 2021. 8, 9

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *arXiv preprint arXiv:2103.02503*, 2021a. 7, 9

Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Learning generalisable omni-scale representations for person re-identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021b. 9

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022a. 7, 9

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*, 2022b. 6, 7, 8, 9

Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016. 9

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8697–8710, 2018. 9

A BACKGROUND

A.1 AUTOFORMER

We present the detailed steps of the AutoFormer algorithm as follows (we use AutoFormer-base as an example). As described below, AutoFormer searches for the optimal architecture for ViT. NOAH adopts AutoFormer for the parameter-efficient tuning task by (i) designing the specific search space for this task (please refer to Section 2.2 for details). (ii) using "weight entangled" to sample subnet weights. Notable, other NAS approaches are also available and can be a future work to extend NOAH, but designing a specific search space, which is general for NAS approaches, is novel for parameter-efficient tuning tasks.

Search Space AutoFormer aims to find the optimal architecture for the Vision Transformer (ViT). The search space of AutoFormer includes different variant factors of the ViT architecture. Specifically, the search space includes the number of layer $\in \{1, 2, 3\}$; the number of multi-attention head $\in \{8, 10, 1\}$; the embedding dimension of multi-attention block $\in \{528, 624, 48\}$; the reduction ratio of MLP block $\in \{3, 4, 0.5\}$; and the embedding dimension of Q-K-V $\in \{512, 640, 64\}$.

Supernet Training The supernet is initialized with the largest architecture in the search space: the number of layer $= 16$; the number of multi-attention head $= 10$; the embedding dimension of multi-attention block $= 624$; the reduction ratio of MLP block $= 0.5$; the embedding dimension of Q-K-V $= 512$. In each training step, AutoFormer randomly samples a subnet for parameter training. First, it samples the value of the number of layers, says 14, which means the subnet only has top-14 layers of the supernet. Then for each layer, it samples values from other factors, say $\{8, 48, 3, 64\}$ for layer 1, which means layer 1 has 8 multi-attention heads; the embedding dimension of the multi-attention block is 48, etc. AutoFormer proposed "weight entangled" to sample subnet weights. Specifically, as the full embedding dimension of the multi-attention block is 624, if the sampled value of the embedding dimension is 48, the embedding of the multi-attention block of the subset is the top-48 of embedding of the multi-attention block of the supernet. Model inference and loss backward at the current step are based on the sampled subnet.

B DETAILS OF NOAH

B.1 SUPERNET TRAINING

We detail the supernet training formally as follows. We first uniformly sample 5 subnet models from the pre-defined search space/model size. We then select the attentive subnet model M as mentioned before, and optimize its corresponding weights W in each training iteration. We represent the structure and weight of this subnet as:

$$\begin{aligned} W &= \left\{W^0, W^i, \cdots, W^l\right\}, \\ M &= \left\{M^0, M^i, \cdots, M^l\right\}, \end{aligned} \tag{5}$$

where i indicates the $i+1$ block of the Transformer with $l+1$ blocks. Referred to the Eq. 234 in the main paper, we denotes the adapter weights as $W(\text{A})$, $W(\text{A}) = \left\{W^{down}, W^{up}\right\}$, the LoRA weights as $W(\text{L})$, $W(\text{L}) = \left\{W_q^{down}, W_q^{up}, W_k^{down}, W_k^{up}\right\}$, the VPT weight as $W(\text{V})$, $W(\text{V}) = \text{P}$. Hence M^i and W^i are denoted as follows:

$$\begin{aligned} W^i &= \left\{W^i(\text{V}), W^i(\text{A}), W^i(\text{L})\right\}, \\ M^i &= \left\{M^i(\text{V}), M^i(\text{A}), M^i(\text{L})\right\}. \end{aligned} \tag{6}$$

There are multiple choices of each prompt module during the prompt search. Hence, $W^i(\cdot)$ and $M^i(\cdot)$ are selected from a set of s candidates that equal the number of choices of the embedding dimension, which is formulated as:

$$\begin{aligned} W^i(\cdot) &\in \left\{W_0^i(\cdot), W_j^i(\cdot), W_k^i(\cdot), \cdots, W_s^i(\cdot)\right\}, \\ M^i(\cdot) &\in \left\{M_0^i(\cdot), M_j^i(\cdot), M_k^i(\cdot), \cdots, M_s^i(\cdot)\right\}, \end{aligned} \tag{7}$$

where $M_j^i(\cdot)$ is a candidate prompt module, *i.e.* Adapter, LoRA or VPT, in the search space and $W_j^i(\cdot)$ is its weights.

In this work, we use the weight entangle strategy Chen et al. (2021) to sample and update the $W_j^i(\cdot)$. Specifically, for any two candidates prompt module $M_j^i(\cdot)$ and $M_k^i(\cdot)$ with their weight $W_j^i(\cdot)$ and $W_k^i(\cdot)$, we have:

$$W_j^i(\cdot) \subseteq W_k^i(\cdot), j \leq k. \tag{8}$$

Weight entangle strategy makes the weight updates of $W_j^i(\cdot)$ and $W_k^i(\cdot)$ entangled with each other.

B.2 EVOLUTIONARY SEARCH

After training the supernet, we conduct the evolutionary search to obtain the optimal subnet under the parameter size limit Chen et al. (2021). In each search epoch, we have three steps. The mutation and crossover steps are illustrated in Fig. 2. We maintain a subnet pool $\mathcal{P}$, which is initialized as ϕ, to store the subnet candidates.

Random sampling. We uniformly produce n legal subnets $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_n\}$. Legal subnets are under the pre-defined parameter limit. Then we renew $\mathcal{P}$, as $\mathcal{P} = \mathcal{P} \cup \mathcal{M}$

Mutation. We sample n subnets from the newest version $\mathcal{P}$, and mutate each module by randomly mutating each prompt module with a probability P_m, producing n new subnets. Then we renew $\mathcal{P} = \mathcal{P} \cup \mathcal{M}$

Crossover. We sample $2n$ subnets from $\mathcal{P}$ in this step. Then, every two subnets are randomly combined, forming n pairs. For a pair, we crossover prompt modules located at different subnets but belonging to the same Transformer layer and the same type, producing two new subnets. Then we renew $\mathcal{P} = \mathcal{P} \cup \mathcal{M}$

After each epoch, we renew $\mathcal{P}$ as the subnets that achieve the top-N validation performance.

C IMPLEMENTATION DETAIL

Augmentation For the VTAB-1k (Zhai et al., 2019), we follows its default augmentation settings, implementing the resizing and normalization for input images. Specifically, we resize a input image to 224 × 224, followed by normalizing it with ImageNet (Deng et al., 2009) means and standard deviation. For few-shot learning and domain generalization experiments, we implement color-jitters with the factor as 0.4, and RandAugmentation with magnitude equals 9, magnitude standard deviation equals 0.5.

Hyperparameters We consistently set the embedding dimension of Adapter (Houlsby et al., 2019) and LoRA (Hu et al., 2021) as 8. We set prompt length of VPT (Jia et al., 2022) following the paper. For few-shot learning (FS) and domain generalization (DG) experiments, we consistently set the VPT prompt length as 8. We use AdamW optimizer (Loshchilov & Hutter, 2017) with the cosine scheduler. The weight decay equals 1e$-$3, warm-up epochs equals 10, and the batch size equals 64. Other hyperparameters are shown in below.

Table 5: **Hyperparameters**. Others includes few-shot learning and domain generalization.

	Learning Rate		Dimension	Depth
	VTAB	Others		
VPT	1e-3	5e-3	{1, 5, 10, 50, 100}	{3, 6, 9, 12}
Adapter	1e-3	5e-3		
LoRA	1e-3	5e-3		
NOAH-supernet	5e-4	5e-4		
NOAH-subnet	1e-3	5e-3		

D EXPERIMENTS

Transferability of Subnet As discussed previously, the subnet (*i.e.*, architecture) found for different datasets differs dramatically. Here we study whether, or in what circumstances, the subnet found from one dataset can be transferred to another. To this end, we train NOAH on ImageNet and apply the ultimate subnet to the VTAB-1k benchmark where the model is retrained and evaluated. To measure transferability, we compare the ImageNet subnet with the dataset-specific subnets on VTAB-1k. Fig. 6 shows the comparisons. Overall, the gap between the ImageNet subnet and the 19 dataset-specific subnets on VTAB-1k is below 3%, meaning that NOAH has fair transferability. By digging deeper

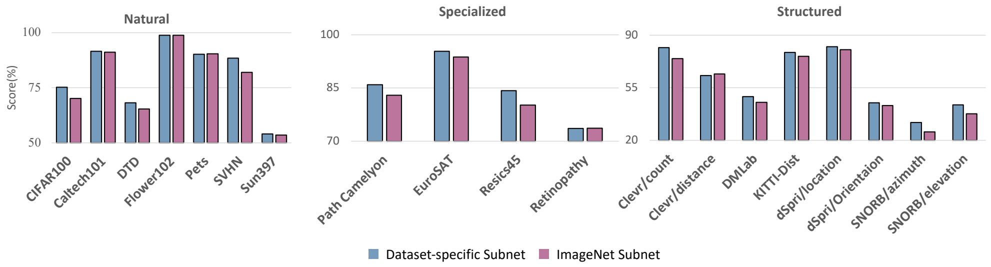
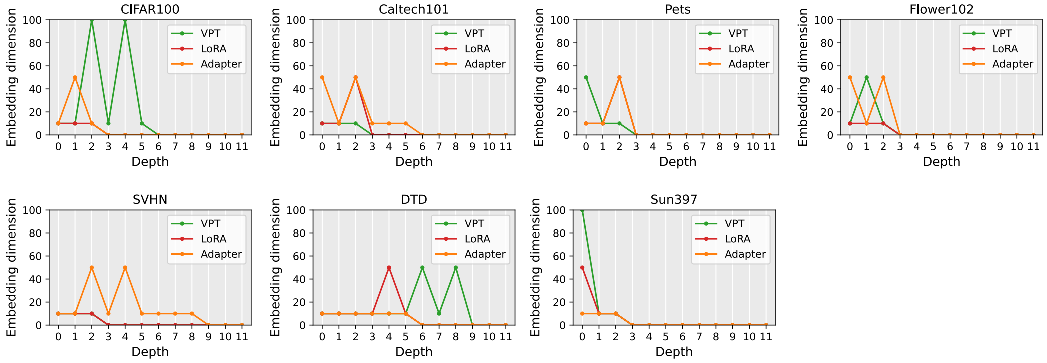


Figure 6: **Evaluation on the transferability of subnets**. Dataset-specific subnet means the architecture is found from the target dataset. ImageNet subnet means the architecture is found from ImageNet and transferred to the target dataset. All target datasets come from VTAB-1k. In general, better transferability is achieved when the source and target datasets are closer, and vice versa.

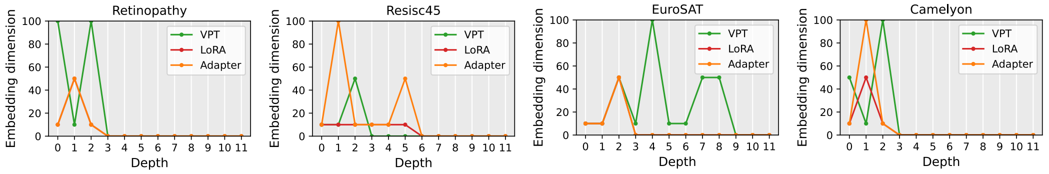
into the results, we find that the transfer gap is smaller when the source (*i.e.*, ImageNet) and target datasets are closer, and vice versa. For instance, the gaps in the Natural group are less than 1%, which make sense because the ImageNet images and those from the Natural group share similar visual concepts, such as generic objects, flowers and animals.

D.1 ARCHITECTURE OF SUBNET

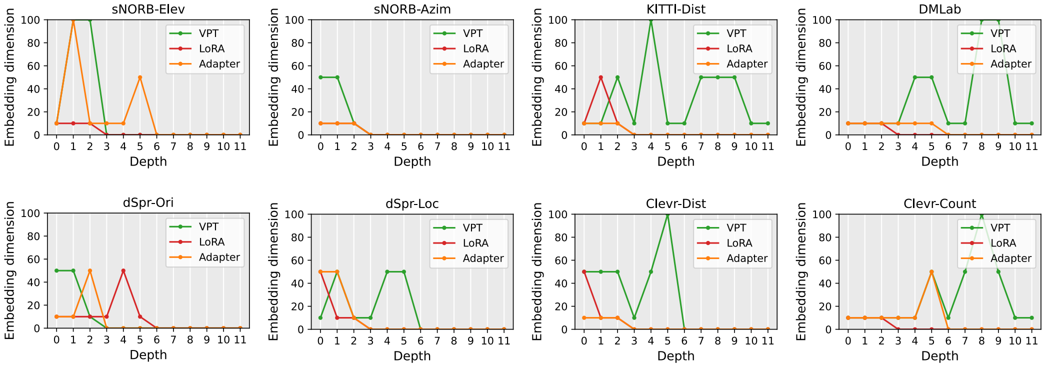
We illustrate the subnets architectures of each dataset in Figure 7. Across all datasets, Adapter and LoRA are usually inserted in the shallow layers. The embedding dimension and depth of VPT are various in different datasets. Specifically, in most datasets of VTAB-natural, VPT has small embedding dimensions, *i.e.*, less than or equal to 50, and lies in shallow depth (layers). In contrast, in the KITTI-Dist (Geiger et al., 2013), DMLab (Beattie et al., 2016), Clevr-Dist (Johnson et al., 2017), and Clevr-Count (Johnson et al., 2017) datasets of VTAB-structured, VPT plays an important role in deeper layers with larger embedding dimensions. We note that the variation of searched VPT by different groups coincides with that of *manually* designed VPT in (Jia et al., 2022), *i.e.*, VPT has a smaller embedding dimension for VTAB-natural and a larger embedding dimension for VTAB-structured, which indicates the superiority of NOAH that can automatically search the optimal subnet without carefully custom designs.



(a) Natural



(b) Specialized



(c) Structured

Figure 7: **Illustration of the subnets architectures for the datasets in VTAB-1k**. Subnet architectures show different characteristic in different groups.