Visual Tuning

BRUCE X.B. YU[*], Zhejiang University-University of Illinois Urbana-Champaign Institute, Zhejiang University, China

JIANLONG CHANG[*], Huawei, China

HAIXIN WANG, National Engineering Research Center for Software Engineering, Peking University, China

LINGBO LIU, Peng Cheng Laboratory, China

SHIJIE WANG, Huawei, China

ZHIYU WANG, Huawei, China

JUNFAN LIN, Peng Cheng Laboratory, China

LINGXI XIE, Huawei, China

HAOJIE LI, College of Computer Science and Engineering, Shandong University of Science and Technology, China

ZHOUCHEN LIN, National Key Lab of General AI, School of Intelligence Science and Technology, Peking University, China and Peng Cheng Laboratory, China

QI TIAN[†], Huawei, China

CHANG WEN CHEN[†], Department of Computing, The Hong Kong Polytechnic University, Hong Kong

Fine-tuning visual models has been widely shown promising performance on many downstream visual tasks. With the surprising development of pre-trained visual foundation models, visual tuning jumped out of the standard modus operandi that fine-tunes the whole pre-trained model or just the fully connected layer. Instead, recent advances can achieve superior performance than full-tuning the whole pre-trained parameters by updating far fewer parameters, enabling edge devices and downstream applications to reuse the increasingly large foundation models deployed on the cloud. With the aim of helping researchers get the full picture and future directions of visual tuning, this survey characterizes a large and thoughtful selection of recent works, providing a systematic and comprehensive overview of existing work and models. Specifically, it provides a detailed background of visual tuning and categorizes recent visual tuning techniques into five groups: fine-tuning, prompt tuning, adapter tuning, parameter tuning, and remapping tuning. Meanwhile, it offers some exciting research directions for prospective pre-training and various interactions in visual tuning.

CCS Concepts: • **Computing methodologies → Computer vision tasks**; *Image and video acquisition*; **Computer vision representations**; **Computer vision problems**.

[*]Both authors contributed equally to this research.

[†]Corresponding author(s).

Authors' addresses: Bruce X.B. Yu, Zhejiang University-University of Illinois Urbana-Champaign Institute, Zhejiang University, China; Jianlong Chang, Huawei, China; Haixin Wang, National Engineering Research Center for Software Engineering, Peking University, China; Lingbo Liu, Peng Cheng Laboratory, China; Shijie Wang, Huawei, China; Zhiyu Wang, Huawei, China; Junfan Lin, Peng Cheng Laboratory, China; Lingxi Xie, Huawei, China; Haojie Li, College of Computer Science and Engineering, Shandong University of Science and Technology, China; Zhouchen Lin, National Key Lab of General AI, School of Intelligence Science and Technology, Peking University, China and Peng Cheng Laboratory, China; Qi Tian, Huawei, China, tian.qi1@huawei.com; Chang Wen Chen, Department of Computing, The Hong Kong Polytechnic University, Hong Kong, changwen.chen@polyu.edu.hk.

ACM Comput. Surv., Vol. 1, No. 1, Article . Publication date: April 2024.

https://mc.manuscriptcentral.com/csur

1 INTRODUCTION

Since the wide usage of Transformer models [72, 105] and the emergence of large foundation models
[10, 294], the paradigm of deep learning in vision intelligence has been experiencing the hype of
adapting downstream tasks to the foundation models. The astonishing performance of the recent
Visual ChatGPT [244] is enabled with myriad computation resources in the pre-training process
[12], and human feedback during the tuning process. The pre-trained foundation model (i.e., GPT-3)
shows strong capability but entails large storage space, around 800GB, to store 175B parameters
[12], which makes it expensive to retrain independent model copies for different downstream tasks.
Foundation models are expected to continue to scale up, and how to reuse the foundation model
via parameter-efficient transfer learning (PETL) methods (e.g., prompt, prefix, adapter, etc.) quickly
becomes a research hype. In the past two years, taking the inspiration of PETL methods in natural
language processing (NLP) [136, 219, 294], numerous visual tuning techniques have been proposed
for adapting downstream tasks to pre-trained vision or visual-language models.

   In the era of increasingly large models, vision models have been scaled up from EfficientNet-based
[177] (480M parameters) to Transformer-based [266] (2, 100M parameters) and even larger scales
such as 22B parameters [43] and 562B [50]. For such large models, PETL methods aim to make
good reuse of the shared parameter weights (usually interpreted as the knowledge of large models)
deployed on the cloud to save storage overhead and to empower edge devices such as autonomous
vehicles, drones, and robots that are intensive in computing and battery resources [273]. This
practice is different from the modus operandi of transfer learning that either fully fine-tunes the
whole model or just fine-tunes the task head (e.g., the last fully connected layer) [301].

   Given the emergence of increasingly large models (i.e., foundation models), we are in a new
paradigm of visual tuning that is beyond tuning the entire model or the task head. How to effectively
reuse the knowledge thereof with PETL methods, leading to less memory usage and higher inference
speed is a hot topic in various vision tasks [130, 212, 213]. Starting with a detailed background in
Section 2, this paper provides an in-depth review of recent tuning advances in the vision domain,
categorizing them into five common types and elaborating their current technical state with
discussions in Section 3. Last but not the least, we provide insights into future research directions
that hold significant promise in Section 4, followed by a conclusion. To the best of our knowledge,
this is the first comprehensive survey on visual tuning, which bears great importance for researchers
to understand the mechanisms and trends of this practice.

2 BACKGROUND

In the early days, machine learning methods relied on feature engineering such as SIFT [104], BRIEF
[20], and ORB [193] to handle specific tasks, which is later on dominated by the deep learning
paradigm [116] since the introduction of ImageNet [45]. Deep learning models [80, 112, 214] pre-
trained on ImageNet are able to benefit various downstream vision tasks such as image recognition,
object detection, and image segmentation via fine-tuning. Fine-tuning is the second step of typical
transfer learning, which makes use of the knowledge acquired from the source domain to facilitate
the learning process of the target domain [216, 301]. Given the promise of large-scale pre-trained
models, visual tuning techniques beyond fine-tuning have attracted increasing research interest,
leading to the visual tuning paradigm as illustrated in Fig. 1. This section will elaborate on the
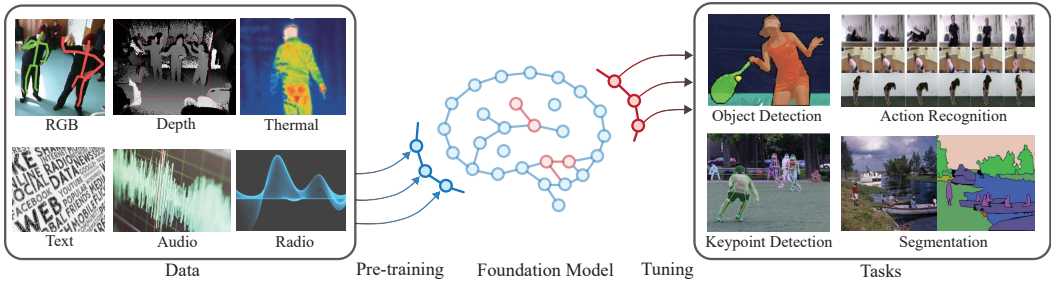
Fig. 1. Illustration of visual tuning. A pre-trained foundation model can accumulate knowledge via various pre-training techniques by scaling up in terms of model size, data modalities, tasks, etc. Given the pre-trained model, the focus of this survey is visual tuning, showing how to effectively reuse the knowledge of the pre-trained models by concerning important aspects such as tuned parameters, generalization ability, data efficacy, training memory, and inference memory, etc.

background of visual tuning from five perspectives: theory, definition, model architecture, model pre-training, and model tuning.

2.1 Theories

In the 1990s, the machine learning community largely ignored neural networks and backpropagation due to concerns about overfitting and the potential for poor local minima. However, in the recent era of deep learning, these concerns have been greatly alleviated via advancements in theories and empirical experiences [116]. In this section, we present the fundamental theories that underpin the current state of visual tuning, exploring these theories from three distinct perspectives as follows.

2.1.1 Biological Perspective. Like the origin convolutional neural network (CNN) architecture was inspired by the receptive fields in the visual cortex [91], learning models can be inspired or motivated by biological and neuroscience discoveries. Particularly, researchers are working on enabling computer vision to have capabilities that are similar to human vision. First, human vision can efficiently process huge amounts of continuous visual streams. Regarding the intrinsic mechanism of this ability, classical biological findings suggest that humans perceive real-world scenes by contextualizing information from local parts (such as small edges) as a whole (i.e., subjective contours), which are respectively handled by cortical areas V1 and V2 [159]. It is also suggested that human vision is embodied and developed in interactive ecological environments [65]. This motivates researchers to work on effective solutions concerning the aspects such as accuracy and efficiency. Second, humans are good at generalizing visual understanding to unseen brand-new scenes by reasoning their physical and geometric properties [114]. This motivates emerged foundation models to be tested via increasingly challenging setups such as zero-shot learning, continual learning, multi-task learning, etc.

2.1.2 Model Perspective. Taking inspiration from human vision, the recently emerged paradigm of tuning large vision models aims to effectively reuse the knowledge in the large pre-trained model in an efficient way regarding computation and data. Generative pre-trained large language models such as GPT-3 show significant continual performance improvements when the model size is scaling up from 0.1B to 175B parameters [12]. This observation is known as the *scaling up law*: larger pre-trained model will benefit downstream tasks, which shows insights that adapting from a larger knowledge base can lead to better performance for downstream tasks. This *scaling up law* has also been proved in recent literature [43, 137]. Sung *et al.* [212] elaborated on the reason for the

Table 1. Notations used in the paper.

Symbol	Definition
m	Number of domains
P	Distribution
$\mathcal{D}$	Domain
S	Source domain
T	Target domain
$\mathcal{X}$	Feature space
$\mathcal{Y}$	Label space
X	Instance set
Y	Label set
N	Number of samples in X
a	Learnable vectors
M	Number of prompts
$\mathcal{T}$	Tokens of visual inputs
Z	Feature propagate through network
l	Neural network layer
k	Input size of a convolutional layer
d	Output size of a convolutional layer
K	Kernel size
G	Group size
r	Dimension of low rank

reduced training memory of PETL techniques from the perspective of backpropagation and further reduced their training memory by skipping the gradient traversal through the frozen backbone, which steps further on the analysis of existing PETL technique regarding training and inference memories.

2.1.3 Statistical Perspective. Machine learning models are restricted by some statistical assumptions such as independent and identically distributed, the law of large numbers, central limit theorem, etc. [44], making practitioners conduct regularization techniques, collect large-scale datasets, and normalize the input data, respectively. In the era of large models, breaking these statistical boundaries becomes imaginable with encouraging recent progress (surveyed in Section 3), which intrinsically improves models' generalization ability to out-of-distribution or long-tail data with less training data (from few-shot to zero-shot learning) and tunable parameters. To guide tuning with statistical rules, there are some works proposed based on measurable domain bound. For instance, Ye *et al.* [263] proposed the concept of expansion function, quantifying regularization or bound restrictions as the "variation" between the source and target domains, and the "informativeness" of a feature. Liu and Zhang [138] also attempted to measure the domain gap by using the test error. Zhang *et al.* [286] proposed to use the margin loss to replace the 0-1 loss for domain adaptation. The margin loss is expected to relax the restriction and provide a more informative generalization bound. Nilesh *et al.* [223] defined task diversity from a statistical perspective, providing generalization upper bounds of sample complexity for multi-task transfer learning.

2.2 Notation and Definition

In order to understand efficient fine-tuning, let's start by defining domains, tasks, transfer learning, and other notations. A joint distribution $\mathcal{X} \times \mathcal{Y}$ can be expressed as $P(X, Y)$ (i.e., P_{XY}), where $\mathcal{X}$

and $\mathcal{Y}$ represent its corresponding feature space and label space, respectively. (X and Y represent the observed instance set and its corresponding label set.) Given P_{XY}, we refer $P(X)$ (i.e., P_X) as the marginal distribution on X, $P_{Y|X}$ the posterior distribution of Y, and $P_{X|Y}$ the class-conditional distribution of X given Y.

Definition 1 (Domain): A domain $\mathcal{D} = \{\mathcal{X}, P(X)\}$ is defined by its feature space $\mathcal{X}$ and a marginal distribution $P(X)$, where X denotes an instance set defined as $X = \{x|x_i \in \mathcal{X}, i = 1, ..., n\}$. A domain can be with or without labeling information.

Definition 2 (Task): A task can be denoted as $\mathcal{T} = \{\mathcal{Y}, f\}$, where $\mathcal{Y}$ and f represent a label space and a decision function, respectively. For the classification task of a source domain $\mathcal{T}^S$, the goal is usually to predict the conditional distribution of instances, which can be denoted as $f(x_j) = \{P(y_k|x_j)|y_k \in \mathcal{Y}, k = 1, ..., |\mathcal{Y}|\}$. In this case, the task $\mathcal{T}^S$ can be regarded as forming a typical source domain $\mathcal{D}^S$ with labeling information, being denoted as $(\mathcal{D}^S, \mathcal{T}^S) = \{(x, y)|x_i \in \mathcal{X}^S, y_i \in \mathcal{Y}^S, i = 1, ..., n^S\}$.

Definition 3 (Transfer learning): Given $m^S \in \mathbb{N}^+$ source domain(s) and $m^T \in \mathbb{N}^+$ target domain(s), their corresponding task(s) can be denoted as $\{(\mathcal{D}^{S_i}, \mathcal{T}^{S_i})|i = 1, ..., m^S\}$ and $\{(\mathcal{D}^{T_j}, \mathcal{T}^{T_j})|j = 1, ..., m^T\}$, respectively. Transfer learning aims to improve the performance of decision functions f^{T_j} on the target domain(s) by making good use of the knowledge learned from the source domain(s).

Definition 4 (Parameter efficient fine-tuning): Given $m^S \in \mathbb{N}^+$ source domain(s) $\{(\mathcal{D}^{S_i}, \mathcal{T}^{S_i})|i = 1, ..., m^S\}$ and $m^T \in \mathbb{N}^+$ target domain(s) $\{(\mathcal{D}^{T_j}, \mathcal{T}^{T_j})|j = 1, ..., m^T\}$ defined in transfer learning, the goal of efficient fine-tuning $\{(\mathcal{D}^{T_j}, \mathcal{Y}^{T_j}, f^{T_j}, f^{S_i})|i = 1, ..., m^S, j = 1, ..., m^T\}$ is to improve the performances of f^{T_j} by reusing $f^{S_i}|i = 1, ..., m^S$ learned from their corresponding tasks denoted as P_{XY}. In particular, the parameters of f^{S_i} need to be frozen or tuned with a small portion. While f^{T_j} denotes an extra small amount of model parameters that can be easily deployed on edge devices. In practice, for supervised or self-supervised pre-training, f^{S_i} can be learned from $P_{Y|X}$ and P_X, respectively. This definition is an extension of typical transfer learning [301], which covers multisource efficient fine-tuning.

2.3 Model Architecture

Pre-trained foundation models for vision have been surveyed in [294], which develops from CNN- and GAN-based models to recent Transformer-based models. We recommend readers refer to [294] for the detailed pre-training strategies. This section briefly introduces these representative models' basic structures: CNN-based, Transformer-based, and CNN+Transformer.

CNN is one of the most popular deep learning models such as AlexNet [112], VGGNet [205], Inception [215], ResNet [80], EfficientNet [217], etc., which has been surveyed time to time [2, 124]. EfficientNet is lightweight yet can achieve comparable performance to Transformer-based models via pre-trained initialization on various visual tasks such as image classification [269] and video understanding [14]. Except for 2D CNN, a couple of 3D CNN models such as C3D [222], I3D [21], S3D [250], and X3D [59] have been introduced for video understanding tasks. In addition, graph convolutional network [257] has also been proposed for tasks such as exercise evaluation [13] and pose estimation [265].

The typical architecture of a Transformer model is structured with several basic Transformer layers. Each layer can be made of a varied number of Transformer blocks composed of a multi-head self-attention (Attention) module and a fully connected feed-forward network (FFN) implemented with a 2-layer multilayer perceptron (MLP). Layer normalization (LN) and residual connection are respectively performed before and after both FFN and Attention modules. Building upon the basic Transformer model, Transformer has been dominating increasing tasks [72, 105]. Early Transformer models for vision are Vision Transformer (ViT) [49], Data-efficient image transformer (DeiT) [221],

while their representative variations are TNT [73], T2T [271], PVT [237], Swin-Vit [146], Video Swin Transformer [147], CPVT [39].

Transformer models are well known for their ability to capture long-range dependencies of input data. Whereas CNNs might be better at representing local features. Models combining Transformer and CNN can achieve better performance. Twins-SVT [38] proposed to add a positional encoding module implemented via a 2D depth-wise convolution 2D in between the Transformer encoders, and designed global and local attention modules to improve the model's representation ability, leading to improved performance on image-based tasks with slightly more model parameters. Representative methods combining CNN and Transformer are Shuffle [90], CMT [71], VOLO [272], etc. Although they can achieve superior performance, how they can be used for visual tuning seems under-explored.

2.4 Model Pre-training

Pre-training methods can be roughly grouped into supervised and self-supervised ones. Early vision models were pre-trained via supervised learning on large-scale datasets such as ImageNet [45], JFT-300M [209], Kinetics [103], etc. Since fine-tuning models pre-trained with supervised learning, larger-scale pre-training has been conducted recently. For example, Gato [188] uses multi-task learning with the supervision of varied tasks to enable the large model to acquire more knowledge for the adaptation of downstream tasks. Multi-label learning is used to pre-train a pure vision model that reaches 22B parameters [43], showing fantastic performance on downstream visual tasks.
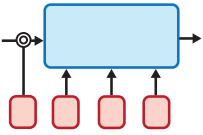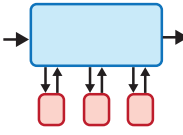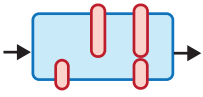
In the regime of supervised pre-training, the non-trivial annotation cost imposes a practical obstacle to scaling up the benefit of transfer learning. Alternatively, self-supervised learning on unannotated data can also make the models richer and potentially more useful [10]. The paradigm of fine-tuning models pre-trained via self-supervision brings the possibility of learning knowledge from unannotated data at a larger scale, which is enabled by advanced computing power, the Transformer model, and more data. Models pre-trained with self-supervised learning are termed "foundation models" by Bommasani *et al.* [10]. Recent notable examples include MAE [79, 220] in vision; CLIP [183], ALIGN [93], Florence [270], BEiT [236], Gato [188], CoCa [266], SWAG [207], etc. for visual-language models. More recently, generative models such as NeRF [161] and Diffusion [173] have also been fine-tuned for better image or video generation such as Latent-NeRF [160], DreamBooth [194], and Tune-a-video [248]. Taking the initial success in NLP, this paradigm has started showing success in vision and various other realms such as climate science [163], protein design [229], intelligent transportation [201], etc. Bommasani *et al.* [10] identified the key significance of foundation models as *emergence* regarding capability and *homogenization* regarding model, modality, tasks, and domains.

2.5 Model Tuning

Given the knowledge learned via pre-trained models, downstream tasks can greatly benefit from them. Early modus operandis of fine-tuning includes updating the whole parameters of the pre-trained model and tuning the task head only (e.g., fully connected layer). With the popularity of large language models such as GPT-3 [12] pre-trained via meta-learning in an unsupervised manner, enabling them to handle a broad set of skills (the inner loop termed "in-context learning"). Given the ability of multiple skills, the current leading paradigm in NLP is to adapt downstream tasks to the large language models, entering the learning paradigm of "pre-train, prompt, predict" from "pre-train, fine-tune" [10].

On the one hand, a couple of recent works [167, 275] achieve promising performances on vision downstream tasks by fine-tuning visual-language models. However, according to the results in

Table 2. A comprehensive review and classification of visual tuning methods. Red and blue parts are tunable and frozen parameters, respectively.

Category	Description	Method
Fine-tuning	All parameters in the pre-trained model are updated in the tuning process. This method is by far regarded as an effective practice to achieve state-of-the-art performance on many vision benchmark datasets. However, when vision models continue to scale up, this fine-tuning method becomes less practicable due to the storage and training overhead.	**CNN**: VGGNet [205], Inception [215], ResNet [80], EfficientNet [217], C3D [222], I3D [21], S3D [250], X3D [59] **Transformer**: ViT [49], DeiT [221], TNT [73], T2T [271], PVT [237], Swin-Vit [146], Video Swin Transformer [147], CPVT [39] **CNN and Transformer**: Shuffle [90], CMT [71], VOLO [272]
Prompt Tuning	Prompt tuning unifies all downstream tasks into pre-trained tasks via designing a specific template to fully exploit the capabilities of foundation models. Prompt tuning usually learns few parameters and keeps pre-trained models frozen. In addition, the core mechanism of the vision prompts aims to exploit the potential of the upstream pre-trained model, so that the upstream pre-trained model can perform the downstream task as well as possible with some or fewer labeled data.	**Vision-driven Prompt**: VPT [94], S-Prompting[239], DePT [64], ZegCLIP [298], ACT [48], PViT [83], TeCoA [156], EVP [247], ProSFDA [89], APT [11], PAT [267], LPT [47], PointCLIP [282], P2P [242], PromptGen [245], NOAH [288], PGN [148], FPTrans [278], FRPT [235], RePro [62], ViLD [68], LION [231] **Language-driven Prompt**: CoOp [296], SubPT [153], MPA [28], ZegOT [109], X-CLIP [164], ProGrad [299], Berg et. al [8], PTP [285], LANIT [170], SgVA-CLIP [175], LASP [17], DualCoOp [210], PLOT [25], CPL [82], DeFo [230], GALIP [218], CoCoOp [295], PointCLIP V2 [300] **Vision-language Prompt**: UPT [275], DPT [252], MaPLe [106], MVLPT [200], MetaPrompt [292], TPT [203]
Adapter Tuning	Adapter tuning is a class of techniques that inserts additional trainable parameters into a pre-trained model frozen to facilitate learning for downstream tasks. The advantage of this method is its lightweight nature and ease of plug-and-play insertion into the middle of a pre-trained network, making it widely applicable in many visual tasks.	**Sequential Adapter**: Res-adapt [186], EPM [187], DAN [192], LST [212], Conv-Adapter [26], Polyhistor [145], Pro-tuning [165], AMixer [185], Fit [204], TINA [158], RepAdapter [150], BDTL [123], ViTDet [122], Florence [270], SND [233], MK-Adapter [280], ADA [55], AIM [261], ST-Adapter [166], PEA [199], CAOA [224], HA [108], CLIP-Adapter [63], Tip-Adapter [281], BALLAD [154], MAGMA [53], VL-Adapter [213], Hierarchical3D [169], HyperPELT [290], SVL-Adapter [168], LAVISH [129], CrossModal-Adapter [95], MV-Adapter [277] **Parallel Adapter**: ViT-Adapter [36], PESF-KD [184], AdaptMLP [31], Convpass [98], AMA [268], UniAdapter [149] **Mix Adapter**: Consolidator [75], ETT [253], PATT [81], PALT [227], TVG [202], VQT [225]
Parameter Tuning	Parameter tuning aims to directly modify the model parameters (i.e., weight and bias). They can be grouped into three categories: bias part, weight part, and both. Common modification schemes can be addition, decomposition, or without extra parameters (i.e., directly tune part of parameters). Representative methods are bias tuning, LoRA, and Compacter.	**Bias Part**: Bitfit [274], Side Adapter [255], AdapterBias [60], DP-BiTFiT [15] **Weight Part**: LoRA [87], MoSA [111], DyLoRA [227] DnA [96], Compacter [102], KAdaptation [81], PHM [276], PHNNs [67], TARP [85], FacT [99], KronA [52], DLDR [121], Aurora [232] **Weight and Bias**: SSF [125]
Remapping Tuning	Remapping-based tuning is a novel approach that involves transferring the learned knowledge of a pre-existing model to a new downstream model. This technique has shown promising results in improving the performance of downstream models and can be categorized into three different types according to the use of the pre-trained model.	**Knowledge Distillation**: KD [84], Fitnet [191], Student [27], DFA [69], AdaIN [259], Normalized KD [254], Heterogeneous KD [172], DeiT [221], Manifold KD [76], Paraphrasing KD [107], RKD [171], AKDNet [141], SemCKD [23], HKD [297], Review [30], DKD [291] **Weight Remapping**: Net2Net [32], EAS [18], N2N Learning [5], NASH [54], Path-level EAS [19], FNA [57], FNA++ [58] **Architecture Remapping**: DARTS [131], DATA [22], DATA-GS [283], P-DARTS [34], DARTS+ [126], SGAS [118], SNAS [251], MiLeNAS [77], DARTS- [40]

[167] and [81], fine-tuning visual-language models do not lead to results as good as fine-tuning supervised pre-trained vision models. In addition, pure vision models are also increasingly large (reach 22B parameters) [43] and gain great advances recently [29, 117, 188] with varied pre-training strategies [105, 294]. As such, it needs further investigation on proper pre-training techniques and fine-tuning techniques for vision downstream tasks.

3 VISUAL TUNING

To the best of our knowledge, there is no survey that systematically summarizes the recent state of visual tuning from the technical perspective. He *et al.* [78] analyzed different PETL tuning methods such as prompt-tuning, prefix-tuning, and adapters in the NLP domain, showing they are intrinsically similar (i.e., they bring a certain amount of tunable parameters for adaptation). Taking parameter efficient transfer learning methods in NLP into consideration, we group visual tuning methods into five categories: fine-tuning, prompt tuning, adapter tuning, parameter tuning, and remapping tuning (see Table 2) according to their structures and motivations. In the remainder of this section, we introduce the five groups of tuning techniques with discussions of their advantages and disadvantages.

3.1 Fine-tuning

We use fine-tuning to denote the standard practice of transfer learning, which either tunes the whole parameters of pre-trained models or just tunes the task head. Many state-of-the-art methods adopted this practice to achieve impressive performance on vision benchmarks such as ImageNet [45], Kinetics [103], COCO [128], NTU RGB+D 120 [132], Human3.6M [92], etc. Tuning the whole pre-trained model intrinsically initiates the learning process of the downstream tasks via the learned model weights. While tuning the task head treats the pre-trained model as a feature extractor.

The full fine-tuning strategy comes with obstacles for adapting large models to downstream tasks. First, it requires one to update and store separate model parameters for different downstream tasks, which can be expensive and infeasible when the foundation models become increasingly large. Second, it relies on high-quality downstream data and can hardly adapt to unseen scenarios that have large distribution shift [113], which is unlike the learning process of humans who can learn from few samples and generalize well to new circumstances. This issue has been researched in directions such as zero-shot learning, few-shot learning, and continual learning [120]. Alternatively, fine-tuning the downstream task head can avoid updating the entire backbone model, but it usually leads to unsatisfactory experimental performance.

3.2 Prompt Tuning

Prompt-based learning is first introduced in NLP to efficiently adapt downstream language tasks to foundation models. Unlike the traditional "pre-training, fine-tuning" paradigm which initializes the weight parameters of pre-trained model and optimizes these parameters under the guidance of downstream task-specific loss functions, prompt-based learning leverages textual prompts to reformulate various downstream tasks as the original pre-trained task. Inspired by prompt techniques in NLP, prompt tuning is also introduced into the computer vision field. Specifically, vision prompt tuning could be divided into three groups, *i.e.*, vision-driven prompt, language-driven prompt, and vision-language prompt.

3.2.1 Vision-driven Prompt. Vision-driven prompt tuning [11, 47, 89, 148, 247, 267, 288, 298] has become a popular parameter-efficient way to transfer the remarkable generalization ability of pre-trained vision models to various downstream tasks. The research efforts of vision-driven prompt strategies can be roughly categorized into two groups, *i.e.*, modifying inputs directly,
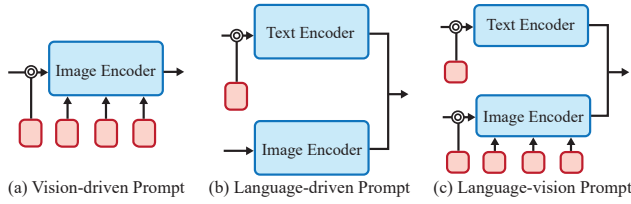
Fig. 2. Three different types of prompt methods. The red and blue parts are tunable and frozen parameters, respectively.

and designing vision prompt sub-networks to produce vision prompts. Studies of the first family [48, 64, 83, 94, 156, 208, 239] usually tend to directly modify inputs, *e.g.*, *adding a set of learnable parameters into input images*, which aims to modify the input distribution and further makes downstream tasks close to the solved task during the original pre-training, as shown in Fig. 2(a). Formally, the mathematical formulation can be described as:

$$\mathcal{P}_V = [\mathcal{T}, a_1, \cdots, a_i, \cdots, a_n], \tag{1}$$

where $\mathcal{P}_V$ indicates the vision-driven prompts, $\mathcal{T}$ denotes the embeddings of local images or tokens outputted by Transformer, and a_i is the i-th learnable vector.

Existing extensive works utilize the above principle to design vision prompts to instruct frozen vision pre-trained models to various downstream tasks. Concretely, VPT [94] plugs solely a few learnable parameters and regards these parameters as a part of input tokens of Transformer, which steers pre-trained vision models to perform various downstream tasks. Similar with VPT, DePT [64] also introduces learnable visual prompts into the vision Transformer and only optimizes these source-initialized prompts while keeping the vision Transformer frozen during adaptation. In addition, PViT [83] designs task-specific prompts by introducing a small set of specialized parameters to adopt a shared video transformer backbone to perform synthetic scene tasks and a real video downstream task. LPT [47] optimizes shared prompts to explore the general features across the entire long-tailed dataset, and group-specific prompts to endow the fine-grained discrimination ability into frozen pre-trained vision models.

As proven by the above works, prompt learning enables pre-trained visual models to adapt to a variety of visual tasks in natural scenarios. However, prompt learning still has great potential in transferring visual knowledge of pre-trained vision models trained in natural scenarios to downstream tasks that have large domain gaps. A recent study has extended vision prompt from natural scene understanding to diverse vision tasks with huge domain discrepancies, such as point cloud analysis [242, 282], image generation [245] and even speech understanding [110]. Concretely, PointCLIP [282] converts the raw points into scatter depth maps by projecting them onto predefined image planes, termed as vision prompt, which effectively transfers the remarkable ability of CLIP model. In addition, PointCLIP also narrows the modality discrepancies between unordered point clouds and the visual images, thus producing a unique insight for processing vision tasks with significant domain gaps using prompt technology. P2P [242] proposes the geometry-preserved projection and geometry-aware coloring operations to translate point cloud data into colorful images, which are regarded as vision prompt and further adapt the pre-trained vision model for various point cloud analysis tasks. These works show that vision-driven prompts can transfer pre-trained vision models from natural scenarios to various downstream tasks even with domain discrepancies.

Excitingly, the above work takes only a simple manner (e.g., adding extra parameters into inputs) to construct visual prompts but makes great progress on transferring the remarkable discrimination

and generalization of pre-trained vision models. To further investigate and dig into the effectiveness of vision prompt, the other family of approaches [61, 62, 68, 148, 235, 278, 288] also tend to *design a sub-network to construct vision prompts*, as shown in Fig. 2(c). Specifically, the vision-driven prompts $\mathcal{P}_V$ can be denoted as:

$$\mathcal{P}_V = \Phi(X, \boldsymbol{\theta}), \tag{2}$$

where $\Phi(,)$ denotes the designed sub-network to produce vision prompts $\mathcal{P}_V$, $\boldsymbol{\theta}$ is the learnable parameters in $\Phi(,)$, and X is the input image. For instance, NOAH [288] combines adapter, prompt, and LoRA. It utilizes the neural architecture search algorithm to learn the down-sampled dimension of adapters, the down-projection dimension of LoRA, and the learnable token length of prompts, leading to better parameter efficiency and performance trade-off. PGN [148] learns to produce input-dependent prompts via selectively sampling input images from a commonly learned library of tokens. FRPT [235] explicitly zooms the discriminative regions of input images via designing a lightweight sampling network to obtain the vision prompts. RePro [62] localizes objects from videos as vision prompts utilizing a tracklet detector and further learns the correlation between subjects and objects according to the learned vision prompts. ViLD [68] generates multiple regions of interest based on the region proposal network, regarded as vision prompts, to align their visual embeddings and textual embeddings for open-vocabulary object detection. These works can produce appropriate prompts according to downstream tasks, thus effectively exploring the remarkable generalization and discrimination ability of pre-trained vision models. More importantly, compared to introducing learnable parameters directly, they can improve the interpretability of these vision prompts such as directly modifying the pixels.

3.2.2   Language-driven Prompt. Recently, large-scale vision-language models are pre-trained by extensive image-text pairs and focus on open-world visual concepts. Following this ideology of prompt learning in NLP, most existing works tend to transfer large-scale vision-language models into various downstream vision tasks via designing appropriate language-driven prompts [8, 127, 153, 170, 175, 218, 285, 295, 300]. As shown in Fig. 2(b), most works, such as CoOp [296], firstly extract unified context or class-specific context of visual images as language-driven prompts to adapt frozen pre-trained vision-language models to diverse vision tasks. Formally, the language-driven prompts can be formulated as below:

$$\mathcal{P}_T = [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_i, \cdots, \boldsymbol{a}_n, < class >], \tag{3}$$

where $\mathcal{P}_T$ denotes the language-driven prompts, a_i represents the i-th learnable vectors, the number of learnable vectors is n, and $< class >$ is the class embeddings. Extensive existing works have focused widely on this line of language-driven prompt and utilize this prompt analogous to the designed in CoOp to adapt various downstream tasks, *e.g.*, domain adaption [28], semantic segmentation [109], video understanding [100, 164], and few-shot learning [299]. In addition, recent methods [17, 25, 82, 196, 210, 230] extend the original language-driven prompt and thus design multiple complementary language-driven prompts to better mine the task-specific knowledge from pre-trained vision-language models. The multiple complementary language-driven prompts $\mathcal{P}_{MT}$ can be represented as

$$\mathcal{P}_{MT} = [\mathcal{P}_T^1, \cdots, \mathcal{P}_T^i, \cdots, \mathcal{P}_T^M]. \tag{4}$$

For instance, PLOT [25] learns multiple comprehensive prompts to capture different attributes of classes and aligns visual embeddings and multiple textual embeddings via optimizing the optimal transport distances between multiple prompts.

3.2.3   Vision-language Prompt. Vision-driven and language-driven prompts have been explored to simultaneously modify the vision and text inputs for pre-trained vision-language models, thus

transferring the discrimination and generalization ability of pre-trained vision-language models thanks to effectively aligning visual and textual embeddings [106, 200, 203, 234, 252, 275, 292]. For an instance, UPT [275] designs a shared prompt network to produce the vision prompt and text prompt, thus narrowing the gap between visual representations and textual embeddings. DPT [252] simultaneously optimizes the visual and textual prompts from the vision and text input perspectives, which aims to modify the textual classifier and visual representations of pre-trained vision-language models. TPT [203] introduces learnable text prompts with random vectors and category names and designs vision prompts generated by cropping input images randomly. These methods can transfer the pre-trained vision-language models to various downstream tasks from the perspective of text and vision inputs.

3.2.4　Discussion. It is well known that the quantity of labeled data largely determines the upper limit of the vision algorithm. Vision prompt learning usually focuses on solving the problem of few-shot or zero-shot learning, which allows the model to perform relatively well even without labeled data. Moreover, visual prompt learning integrates all subsequent tasks into pre-training tasks by creating a distinct template. Through this approach, data from downstream tasks are converted into new inputs to leverage the inherent capacities of pre-trained models. In other words, the core mechanism of the vision prompts aims to harness the capabilities of the upstream pre-trained model, allowing it to excel in downstream tasks even with minimal reliance on annotated data.

However, prompt-based tuning also suffers from some limitations, vitally sacrificing the applicability in the real world. Firstly, a significant challenge facing prompt tuning is how to construct or highlight effective visual cues of inputs and seamlessly integrate them with downstream tasks. This necessitates a profound understanding and solid technical expertise in both the fields of original pre-training tasks and downstream tasks. Additionally, the prompt-based tuning approach still demands substantial computational resources for model training and optimization, inevitably leading to increased training time and costs. Lastly, despite prompt tuning showcasing notable performance improvements in many tasks, its generalizability requires further exploration and validation when facing huge domain differences between original pretraining tasks and downstream ones.

Despite these limitations, prompt tuning will assume an increasingly pivotal role in the realm of artificial intelligence. We posit that exploring three specific avenues could mitigate some of the current limitations. Firstly, prompt tuning tends to transparent and interpretable adjustments to the input prompts. This transparency enables researchers and practitioners to understand and validate the model's decision-making process, performing better against different data distributions or interferences. Furthermore, researchers and users can tailor prompts to steer the model's attention towards specific features or classes of interest, making the model more usable for various applications and tasks. Prompt tuning contributes to the usability of deep learning models by facilitating model interpretability and controllability. Lastly, prompt tuning tends to promote consistency in model performance by enabling standardized methodologies for adjusting prompts across different datasets and tasks. This consistency ensures that models behave predictably and reliably in various scenarios, enhancing their overall usability and applicability.

3.3　Adapter Tuning

Adapter-based methods are a class of techniques that involves additional trainable parameters into a pre-trained model that has been frozen to facilitate learning for downstream tasks. In the NLP domain, adapters were first introduced by Houlsby *et al.* [86] as a means of achieving PETL. However, efficient adaptation, particularly in the field of computer vision, has received comparatively little attention. Initial efforts to develop adaptive methods for computer vision have included incremental learning methods [192] and domain adaptation methods [186, 187]. Subsequently, adapters have

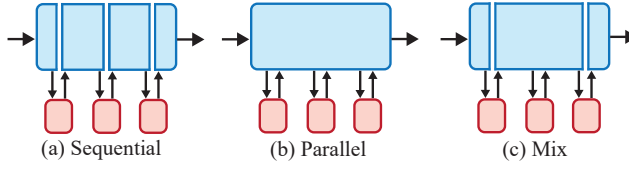(a) Sequential   (b) Parallel   (c) Mix

Fig. 3. Three different types of adapter methods. Red and blue parts are tunable and frozen parameters, respectively.

garnered interest across domains and have been successfully applied in the computer vision field. Adapters provide a lightweight alternative to extensive model fine-tuning.

In this section, we have sorted out the existing vision-related adapter-based tuning methods, which can be roughly divided into three ideas, *i.e.*, sequential adapter, parallel adapter, and mix adapter one by one as follows.

3.3.1 Sequential Adapter. Sequential adapter refers to the technique of inserting parameters into a sequential forward network shown in Fig. 3(a), which typically includes a linear down projection, a non-linear activation function, an up projection, and a residual connection. This approach is commonly applied after the multi-head attention layer and/or the feed-forward layer to enhance model performance. In particular, given a d-dimensional input feature map $Z^{(l)}$, the number of parameters of adapter can be adjusted by a hyperparameter d_{bottle} ($d_{\text{bottle}} \ll d$). The sequential adapter module first uses a down-projection (*i.e.*, downsampling) with $\boldsymbol{W}_{\text{down}} \in \mathbb{R}^{d \times d_{\text{bottle}}}$ to project the feature to the lower-dimensional representation, followed by a ReLU activation function and an up-projection (*i.e.*, upsampling) with $\boldsymbol{W}_{\text{up}} \in \mathbb{R}^{d_{\text{bottle}} \times d}$. The above formulation can be written as:

$$\hat{Z}^{(l)} = \text{ReLU}(\text{LN}(Z^{(l)})\,\boldsymbol{W}_{\text{down}})\,\boldsymbol{W}_{\text{up}}, \tag{5}$$

where $\hat{Z}^{(l)}$ denotes the optimized features outputted by the sequential adapter.

In sequential adapter strategies, research can be categorized into two groups: inserting residual blocks directly, and using parameter optimization techniques to minimize adapter size. Studies of the first group [26, 186, 192, 212] emerged early without large-scale models. Res-adapt [186] involves a customized deep network with adapter residual modules to adapt to different visual domains in real-time. DAN [192] converges to comparable or higher performance with a fraction (typically 13%) of the parameters of standard fine-tuning after Res-adapt. Recent work [212] introduces LST, which trains a separate ladder network using intermediate activations and shortcut connections to improve accuracy and reduce computational complexity. Additionally, Conv-Adapter[26] investigates feasible solutions to learn task-specific knowledge by adapting intermediate features of each residual block using four variants.

EPM [187] suggests using universal parametric neural network families with limited parameters, while Polyhistor [145] decomposes a hyper-network into separate hyper-networks and factorizes adapter weight matrices. Additionally, Pro-tuning enriches the feature space with multiple prompt blocks [165], while AMixer captures long and short-term dependencies without self-attention [185]. Shysheya et al. propose Fit [204], which scales and shifts activations and uses a Naive Bayes final layer classifier for image classification. Marouf et al. introduce TINA [158], which iteratively reduces adapter size using a scoring function compared to neuron importance, improving overall model efficiency. Finally, Luo et al. propose RepAdapter [150], which uses re-parameterization of sparse structure to approach nearby projection weights, reducing model parameters while maintaining effectiveness and lightweight nature.

Adapters have become a popular technique for foundation tasks where the pre-training task is often image classification. However, other tasks such as high-level vision tasks [55, 122, 123, 134, 152, 270, 280], low-level vision tasks [224, 233], video understanding [166, 261, 270], and robotic control [199] all require designs that are tailored to their specific architecture, in order to efficiently transfer learned parameters and achieve good performance through PETL. In addition to these task differences, recent research has proposed innovative ways to utilize adapters in different applications. Recent research proposes innovative ways to use adapters, such as BDTL and ViTDet [122, 123] adjusting a plain backbone with minimal adaptation for object detection, and Florence [270] incorporating universal visual-language representations for a wide range of tasks such as retrieval, classification, object detection, visual question answering, and action recognition. SND [233] uses a dynamic stacked network for image restoration, MK-Adapter [280] blends predictions for few-shot classification, and ADA [55] performs continual learning. AIM [261] and ST-Adapter [166] equip models with spatio-temporal reasoning for video understanding. PEA [199] addresses robotic manipulation limitations, and CAOA [224] optimizes image compression with adapters.

In the field of multi-modal learning, with the development of large-scale cross-modal pre-trained models, *i.e.*, CLIP [183] and ALIGN [93], adapter technique has been widely adopted, using a design analogous to the one mentioned above, to adapt various downstream tasks for efficient fine-tuning [53, 63, 95, 129, 154, 168, 213, 277, 281, 290] with excellent results. HA [108] recommends general recipes for efficient multi-modal transfer learning. CLIP-Adapter [63] uses residual-style feature blending with an additional bottleneck adapter, while Tip-Adapter [281] enhances few-shot capability without backpropagation during training. MAGMA [53] combines visual and textual inputs for generative language models, and BALLAD [154] augments representations for long-tailed vision language learning. Hierarchical3D [169] integrates multi-modal content into a textual summarizer, while VL-Adapter [213] adjusts pre-trained models with sequential adapter layers for cross-modal domains. HyperPELT [290] fine-tunes small modules using a shared hyper-network, while CrossModal-Adapter and MV-Adapter [95, 277] allow early cross-modal interactions. SVL-Adapter [168] combines vision-language pre-training and self-supervised representation learning, and LAVISH [129] migrates adapters for pre-trained ViTs to audio-visual tasks. These approaches demonstrate the versatility of adapters and their potential for various applications beyond traditional classification tasks in multi-modal learning.

3.3.2 Parallel Adapter. Parallel adapter [31, 36, 98, 135, 149, 184, 268] has been proposed as a variant of the classic sequential adapter architecture shown in Fig. 3(b). Here, activations are passed via the module layer in parallel to the adapted sub-layer (i.e. feed-forward or attention layer), as opposed to the established, sequential, order of computations. The parallel adapter module also uses a down-projection (*i.e.*, downsampling) with $W_{\text{down}} \in \mathbb{R}^{d \times d_{\text{bottle}}}$ to project the feature to the lower-dimensional representation, followed by a ReLU activation function, and an up-projection (a.k.a. (*i.e.*, upsampling)) with $W_{\text{up}} \in \mathbb{R}^{d_{\text{bottle}} \times d}$ in parallel. Formally, the process of parallel adapter can be described:

$$\hat{Z}^{(l)} = \text{ReLU}(\text{LN}(Z^{(l)}) W_{\text{down}}) W_{\text{up}} + \text{LN}(Z^{(l)}), \tag{6}$$

where $\hat{Z}^{(l)}$ denotes the optimized features outputted by the parallel adapter.

The simplest application of adapters is to insert a module in parallel. ViT-Adapter [36] introduces image-related biases by a pre-training-free adapter, while PESF-KD [184] updates only the adapter for soft labels. AdaptMLP [31] adapts to large video action recognition using two parallel branches. Convpass [98] uses trainable convolutional blocks to improve inductive bias. AMA [268] restores 2D structure for each modality, and UniAdapter [149] unifies uni-modal and multi-modal adapters with partial weight sharing. These approaches demonstrate the versatility of adapter modules in various applications.

3.3.3   Mix Adapter. Mix adapter [75, 202, 225, 246, 253, 264] introduces new parameters in different positions with mixed architecture demonstrated in Fig. 3(c), *i.e.*, the multi-head attention blocks in each Transformer layer.

PATT [264] explores efficient parameter techniques for video-based downstream tasks with a prefix-tuning module. ETT [253] uses attentive prefix tuning and domain residual adapters for few-shot learning. PALT [246] prunes adapters based on the lottery ticket hypothesis. VQT [225] aggregates intermediate features for parameter and memory-efficient transfer learning. Consolidator [75] structures tunable parts for efficient transfer learning with group-wise convolution. TVG [202] compares pre-trained models and adapters for video grounding tasks. These approaches demonstrate the versatility of efficient adapter techniques in various applications.

3.3.4   Discussion. Adapter-based methods represent a popular PETL approach within vision and multi-modal learning, emphasizing the modification of a small set of parameters within a frozen backbone to address downstream tasks. This not only economizes on computational expense but also introduces a high degree of modularity. Such modularity facilitates the swift adaptation of pre-trained models to new tasks without necessitating significant architectural overhauls. Meanwhile, by focusing adaptation efforts on a concise set of parameters, adapter-based techniques maintain the integrity of the original model's learned representations, thereby enhancing the model's generalization capabilities across various tasks. Moreover, Adapters introduce variability through methods like projecting down and up with intermediate non-linear layers, offering a range of model adjustments not typically available through direct parameter tuning.

However, adapter tuning has its limitations when compared with other methods. On one hand, adapter tuning lacks of interpretable semantic meaning compared with prompt tuning. On the other hand, it can be slightly less parameter efficient than parameter tuning such as LoRA. Regarding the comparison with remapping methods, adapter tuning is faced with the challenge of where to insert parameters (such as Transformer models' attention and feed-forward modules, between the Transformer layers or blocks, etc.). Existing adapter tuning methods seem to have no consistent rule but just insert parameters to specific layers.

We posit that exploring two specific avenues could mitigate some of the current limitations. Firstly, introducing more efficient operations could broaden the applicability of adapter-based methods across various communities. Not all layers of a foundation model may require adapters; a unified rule, akin to the scaling principles used in foundation models, could dictate their strategic implementation, enhancing efficiency. Secondly, more adapter architecture can be studied. For instance, in the realm of NLP, there exist adapter architectures that exhibit promising performance in adapting to new tasks [176], which can be leveraged and applied to visual tuning. Furthermore, emerging integration techniques will likely enable adapters to achieve improved performance in practical applications.

3.4   Parameter Tuning

An effective parameter-based tuning involves directly modifying the parameters (either weights or biases) of the pre-trained model in a more aggressive manner. Given a specific layer, it can have its weight-term multiplied to the feature map and a bias-term added to the feature map. As shown in Fig. 4, this section introduces parameter-based methods based on which part of the parameters are tuned: weight part, bias part, and both. Techniques can be grouped into addition and decomposition. Existing works also termed this technique as reparameterization-based methods [46, 150].

Given a neural network layer with parameters (k, d, K, G), where $k = C_{\text{in}}$ is the input channel, $d = C_{\text{out}}$ is the output channel, K is the kernel size, and G is the group size. When $G = 1$, we will
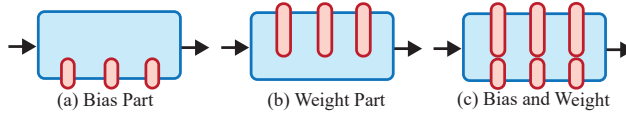
(a) Bias Part   (b) Weight Part   (c) Bias and Weight

Fig. 4. Three types of parameter tuning. Red and blue parts are tunable and frozen parameters, respectively.

have $W \in \mathbb{R}^{d \times k \times K}$ and $b \in \mathbb{R}^d$. Then a typical neural convolutional operation can be denoted as:

$$Z^{(l)} = Z^{(l-1)} W + b, \tag{7}$$

where $Z^{(l)}$ and $Z^{(l-1)}$ denote the output and input features at the l-th neural network layer. The group parameter G can be used to control the connections between inputs and outputs, leading to the weight becoming $W \in \mathbb{R}^{d \times \frac{k}{G} \times K}$. For ease of explanation, we do not consider the kernel size and feature size but focus on the variable size. In the remainder of this section, parameter tuning methods are introduced based on three groups: bias part, weight part, and both.

3.4.1 Bias Part. Bitfit [274] is also known as side-tuning, which only tunes the bias part of the pre-trained model (see Fig. 4[a]) and can be represented as:

$$Z^{(l)} = Z^{(l-1)} W + b, \tag{8}$$

where the weight parameters W are frozen, and the bias b contains the parameters optimized in the tuning process. Avoiding change in the bias of the pre-trained model, AdapterBias [60] targets the bias term at the MLP layer by using a linear layer L with weight ($\alpha \in \mathbb{R}^d$) and a tunable vector $v \in \mathbb{R}^r$, which can be calculated as:

$$Z^{(l)} = Z^{(l-1)} W + b + v \otimes \alpha. \tag{9}$$

Xu *et al.* [255] introduced side-tuning as two branches: one for predicting mask proposals, and the other for predicting attention bias which is applied in the CLIP model for semantic segmentation. It adds a bias term to the results of the Softmax layer of the attention module. Differentially Private Bias-Term Fine-Tuning (DP-BiTFiT) [15] proposed a differentially private version of bias-tuning. DP-BiTFiT used the optimizer DP-SGD to make the bias term private: first aggregate bias gradient norms across all layers, then use it to compute clipping factor, add Gaussian noise to the sum of clipped gradients, and descend on bias term. DP-BiTFiT basically changed the way for optimizing the bias term, which achieves comparable performance with bias-tuning. DP-BiTFiT's implementation is worth noting as it does not calculate the gradients for the pre-trained weights, which helps to save over 60% training time.

Namazifar *et al.* [162] studied the role bias-term of Transformer for NLP tasks. From the mathematical perspective with empirical verification, it concludes that the bias term of the key linear transformation is redundant and can be omitted without any impact on the attention module. Moreover, the bias term of the value linear transformation has a more prominent role than that of the bias term of the query linear transformation.

3.4.2 Weight Part. Fig. 4(b) shows models that tune the weight part of some layers. Given the parameter of a neural network layer with weight $W \in \mathbb{R}^{d \times k}$, LoRA [87] learns parameters $W_{\text{down}} \in \mathbb{R}^{d \times r}$ and $W_{\text{up}} \in \mathbb{R}^{r \times k}$ on top of W, which can be denoted as:

$$Z^{(l)} = Z^{(l-1)} + Z^{(l-1)} (W + W_{\text{down}} W_{\text{up}}). \tag{10}$$

The LoRA structure has been applied to an encoder-decoder model called motion style adapters (MoSA) [111]. MoSA uses a lightweight LoRA structure for adapting the motion style (e.g., pedestrians) from a source domain with sufficient labeled data to a target domain (e.g., cyclists). DyLoRA

[227] proposes to truncate the parameters of rank to multiple parts (*i.e.*, ranks) and optimize them separately sequentially without relying on a search mechanism.

Decomposition-and-Alignment (DnA) [96] uses GreBsmo (replaced with SVD in implementation) to decompose the weight matrix $W \in \mathbb{R}^{d \times k}$ to a low-rank form: $W = UV + S$, $U \in \mathbb{R}^{d \times r}$ is the "alignable" part, $V \in \mathbb{R}^{r \times k}$ is the "fixed support" from the pre-trained model, $S \in \mathbb{R}^{d \times k}$ is the residual term. Two additional variables ΔU and ΔS are added to the "alignable" of the decomposed W, which can be denoted as:

$$Z^{(l)} = Z^{(l-1)}((U + \Delta U)V + S + \Delta S). \tag{11}$$

DnA remains needs to use SVD to implement the GreBsmo algorithm, bringing additional complexity to the iterative optimization process.

Compacter [102], KAdaptation [81], and Aurora [232] use Kronecker products to decompose weight parameter to a $W = \sum_{i=1}^{n} A_i \otimes B_i$, $A_i \in \mathbb{R}^{n \times n}$, $B_i \in \mathbb{R}^{\frac{k}{n} \times \frac{d}{n}}$ and tune one part of the decomposed term B_i with a low rank formed parameters $B_i = u_i v_i$, $u_i \in \mathbb{R}^{\frac{k}{n} \times r}$, $v_i \in \mathbb{R}^{r \times \frac{d}{n}}$, wihch can be represented as:

$$Z^{(l)} = Z^{(l-1)}(\sum_{i=1}^{n} A_i \otimes u_i v_i). \tag{12}$$

The decomposition method using the Kronecker product is also named as a parameterized hyper-complex multiplication/convolutional (PHM/PHC) layer [67, 276], being applied for varied tasks such as vision and audio tasks. PHM [276] inspires [3] to form a tunable weight with three terms z_i, s_i, and A_i, being added to the pre-trained weight for PETL of NLP tasks. FacT [99] considers two decomposition methods Fact-TT and Fact-TK, using Kronecker product and a multilinear generalization of the SVD (i.e., the Trucker model) [42], respectively. Fact-TK generally performs better than Fact-TT with slightly more parameters than Fact-TT across 19 image-based tasks, which is far fewer parameters than the basic LoRA method. Dynamic Linear Dimensionality Reduction (DLDR) [121] claims that only optimizing the low-dimensional subspace of a large model can achieve comparable performance. DLDR used SVD to decompose the weight to find the tuned subspace, which achieves comparable performance by training a small number of epochs.

RepAdapter [150] is built based on LoRA structure and introduced a group-wise transformation [151] method to reparameterize the weight term. RepAdapter also interpreted its group-wise divided LoRA layers as a reparameterization process. RepAdapter [150] aims to reduce the inference time and seamlessly integrate the RepAdapter into most giant vision models via structural reparameterization.

Similarly in NLP, task-adaptive reparameterization (TARP) [85] uses the Kronecker product as a dynamic low-rank decomposition for the MLP module for domain adaptation. Kronecker Adapter (KronA) [52] also introduces the Kronecker product to improve the limited representation power low-rank representation for NLP tasks.

3.4.3 Weight and Bias. As illustrated in Fig. 4(c), some methods modify parameters of both weight and bias parts. Scale and Shift the deep Features (SSF) [125] works towards the weights and bias terms by using two vectors $\gamma \in \mathbb{R}^d$ and $\beta \in \mathbb{R}^d$, which can be represented as:

$$Z^{(l)} = \gamma(WZ^{(l-1)} + b) + \beta, \tag{13}$$

where are respectively interpreted as scale and shift factors. Note that both γ and β are learnable vectors, which can be much smaller than matrix variables of LoRA or decomposed forms of DnA, Compacter, and KAdaptation.
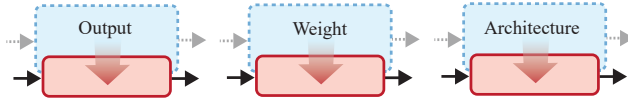
Fig. 5. Three different types of remapping tuning methods. Red and blue parts are tunable and frozen parameters, respectively.

3.4.4 Discussion. Differences from the prompt-based and adapter-based methods, parameter-based tuning can use fewer parameters to achieve a similar effect of adaptation. On the tested image-based tasks [125] can even outperform adapter and VPT. SSF [125] introduces the bias-tuning technique to the weight variable via dot product. According to the analysis of SSF, it intrinsically modifies both the weight and bias variables, which is interpreted as follows:

$$Z^{(l)} = \boldsymbol{\gamma}(\boldsymbol{W}Z^{(l-1)} + \boldsymbol{b}) + \boldsymbol{\gamma} = (\boldsymbol{\gamma} \odot \boldsymbol{W})Z^{(l-1)} + \boldsymbol{\gamma} \odot \boldsymbol{b} + \boldsymbol{\gamma}, \tag{14}$$

where $\odot$ is dot product. Given the varied techniques available, Mao *et al.* [157] unified these methods with a gate mechanism. [97] uses pruning techniques to drop the activations during back-propagation, leading to sparse activations. This track of techniques will be further introduced in Section 3.5. In addition to Transformer-based structures, LoRA Winograd convolution [181] aims to use the LoRA mechanism to prune the 3D CNN backbone model (e.g., C3D and R3D-18) for accelerating the Winograd operation [115] with less trainable parameters.

Although parameter-based tuning can be less expensive from the perspective of tuned parameters, sometimes they will underperform the former two methods (i.e., prompt tuning and adapter tuning). This might be because fewer parameters can reduce the adaptation ability to a target domain with a large domain gap. Another limitation of existing parameter-based tuning methods is the lack of exploring based on the semantics of pre-trained models, leading to insufficient explainability. By far, most methods are tested on Transformer-based structures, but there remains exploration of their effect on CNN-based structures.

In the future, there will be continual exploration along this track of technique for more progressive parameter efficiency via further factorization on pre-trained models' weight or bias terms. Meanwhile, visual semantics are expected to be considered based on different types of pre-trained models (i.e., foundation models pre-trained with varied levels of vision tasks: low-level, middle-level, and high-level). It can also be combined with other tuning techniques for better interpretability. In addition, existing methods can also be expanded to CNN-based methods that have superior performance over corresponding Transformer-based methods on some specific tasks.

3.5 Remapping Tuning

Instead of directly fine-tuning or processing the pre-existing model, remapping-based tuning is a category of techniques that transfer the knowledge learned by a pre-trained model to a new downstream model. Based on how to utilize the pre-trained model, *i.e.*, output, weight, and network architecture (see Fig. 5), we discuss three forms of knowledge transfer in the following categories: knowledge distillation-based remapping, weight-based remapping, and architecture-based remapping.

3.5.1 Knowledge Distillation. Knowledge distillation aims to regularize the downstream model by enforcing it to mimic the output of pre-trained models. Note that, the output typically refers to the final response or intermediate features. In the meantime, knowledge distillation is also an important model compression technique. In this section, we do not involve other model compression

techniques such as network pruning [66, 155, 243], since they are not typically motivated to transfer knowledge from teacher network to student network.

The fundamental idea of knowledge distillation is to transfer the learned knowledge from a large pre-trained teacher model into a small student model by learning the network output or intermediate features of the teacher. Typically, knowledge is distilled from the teacher model to the student model using a soft target distribution for each case. The probability q_i of each case can be formulated as:

$$q_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}, \tag{15}$$

where z is the output logit of the teacher networks and T is the temperature of the distillation process.

To our best knowledge, the work of [16] first introduces knowledge distillation to extract knowledge from a pre-existing model. They trained a compressed model with the pseudo data produced by an ensemble of shallow networks while no significant loss occurs in performance. This idea has been expanded to compress the deep and wide networks into shallower ones in [6]. Hinton *et al.* [84] introduced the teacher-student knowledge distillation framework, where the student network is penalized based on the softened class distribution output of the teacher network.

One of the characteristics of deep neural networks is to obtain increasingly expressive power by learning hierarchical feature representations, as pointed out in [7]. Based on this theory, both the final response and the intermediate feature maps of the teacher network can be employed as the target for training the student model. To substantially exploit the information of intermediate layers, Fitnets [191] introduces intermediate-level hints of the teacher to facilitate training the student. It enforces the intermediate feature alignment between the teacher and student networks via the teacher's intermediate feature maps as hints. Subsequently, a rich line of work is devoted to aligning the features indirectly [27, 69, 76, 107, 172, 221, 254, 259]. Concretely, Kim *et al.* [107] developed a factor transfer method that employs paraphrased intermediate features of the teacher as a factor, rendering the knowledge of the teacher network more understandable for the student network. Inspired by neural architecture search (NAS) [131], Guan *et al.* [69] developed a two-stage distillation approach that adopts the differentiable search strategy to simultaneously improve the efficiency and the effectiveness of knowledge distillation. Xu *et al.* [254] developed a feature-normalized distillation method by introducing a sample-specific correction factor for the replacement of the temperature, with the goal of suppressing the impact of noise resulting from the one-hot label. Passalis *et al.* [172] modeled the information flow of the teacher's multiple intermediate layers and then trained a student model to match this information flow. To realize knowledge transfer for vision transformers, Touvron *et al.* [221] introduced a token-based distillation strategy termed DeiT, which enforces the student transformer to directly reproduce the label estimated by the pre-trained teacher network using a distillation token. Hao *et al.* [76] introduced a manifold distillation approach for vision transformers by substantially utilizing patch-level information.

There are also some extensions to further explore knowledge transfer patterns. Park *et al.* [171] proposed a relational knowledge distillation scheme for mutual relations transfer of outputs instead of individual outputs. As a generalization of vanilla knowledge distillation, they introduced distance-wise and angle-wise distillation losses to sufficiently extract the structural relations in data examples. Liu *et al.* [141] proposed an architecture-aware knowledge distillation approach termed AKD, with the goal of finding the optimal student networks for distilling a given teacher network. Chen *et al.* [23] studied the semantics of intermediate layers and employed an attention mechanism to automatically assign the soft layer association between teacher and student networks, which can reduce the impact of over-regularization during the training process. Zhou *et al.* [297] proposed a

holistic knowledge distillation with graph neural networks, where the holistic knowledge contains individual knowledge and relational knowledge [139, 174]. To integrate two knowledge and refine their correlations, graph neural networks are adopted to learn holistic knowledge to provide supervision for the student network by aggregating feature representation from correlated data examples. Chen *et al.* [30] proposed a residual distillation framework termed Review to effectively learn informative features from multi-level information in the teacher network. Review utilizes multiple layers in the teacher to guide the training for one layer in the student with great performance gains. Zhao *et al.* [291] modeled the traditional knowledge distillation loss into target class knowledge distillation and non-target class knowledge distillation, then dived into their effects. Based on the observations, they found that the traditional knowledge distillation loss is a highly entangled formulation and introduced a decoupled method to facilitate the knowledge distillation.

For application, most of the above methods focus on image classification. Furthermore, knowledge distillation also demonstrates promising results in more vision tasks, such as object detection [70, 119, 279, 293], image segmentation [140, 143, 241, 258], person re-identification [179, 189], super-resolution [4, 284], depth estimation [88, 240], and crowd counting [133].

3.5.2 Weight Remapping. Rather than relying on the teacher's output as supervision to train the student, weight remapping directly transfers the model weights from the teacher network to the student ones. Specifically, assume that a teacher network is a function $f(x; \theta)$ parameterized by θ, where x is the network's input. Weight remapping for student network g is to reassemble a new set of parameters θ' from the existing parameters of θ, such that:

$$\forall x, f(x; \theta) = g(x; \theta'). \tag{16}$$

Net2Net [32] is a pioneering effort that rapidly transfers the knowledge stored in a pre-existing network into another network by remapping the weight of a pre-existing teacher network to the student. Subsequently, EAS [18] introduces the concept of weight remapping into neural architecture search by exploring the search space according to a pre-existing network and reusing its weights. To tackle variable-length architecture and consider the entire input architecture, EAS employs a bidirectional recurrent network [198] as the encoder network. In this way, the previously trained network can be further exploited to efficiently explore the architecture space and greatly accelerate the training process of the new network. To efficiently compress the teacher network for knowledge transfer, Ashoket *et al.* [5] proposed a reinforcement learning-based approach termed N2N learning, which models the conversion from the teacher network into a student network as a Markov Decision Process (MDP). N2N learning formulates the process of knowledge transfer as a two-stage action selection. In the first stage, a recurrent policy network selects a sequence of actions including keeping or removing layers of the large teacher network. In the second stage, another policy network performs further reduction in each remaining layer to meet the attenuate configuration.

Furthermore, some interesting weight remapping methods take the network path topology into consideration instead of merely adding or removing network layers. Elsken *et al.* [54] introduced a hill climbing-based approach named NASH, which can automatically search the optimal student architecture. By using a series of alternative network morphisms, NASH can train the child networks with a short optimization process by cosine annealing. At each training step, NASH searches for the optimal architectures by a simple hill-climbing strategy [195]. Path-level EAS [19] enforces the meta-controller to change the topology of network connection paths while using function-preserving transformation operations to remap weights. To achieve this, Path-level EAS develops a bidirectional tree-structured meta-controller based on reinforcement learning, in order to enrich the architecture space to the generalization of multi-branch structures. And Yang *et al.* [262] proposed

to customize networks efficiently via reassembling various pre-trained network blocks subject to downstream constraints.

Previous object detection and semantic segmentation approaches use the network weights pre-trained on image classification for performance gains. However, one of the major challenges is that ImageNet pre-training typically requires highly large computation costs. To address this issue, Fang *et al.* [57] introduced a fast neural network adaptation approach dubbed FNA, making a pre-trained network adapt to a new task by modifying the network such as depth and kernels. In this way, FNA can expand NAS techniques to object detection and semantic segmentation with negligible computation costs. Technically, FNA first designs a seed network by selecting a manually designed network pre-trained on ImageNet such as [197] and then enlarges it to a super network. By applying the weight remapping technique, the seed network is used to assign new model parameters. The follow-up work FNA++ [58] extends the weight remapping of FNA to one more task (*i.e.*, human pose estimation) and more network architectures including ResNet [80] and NAS networks with diverse widths, depths, and kernel sizes.

3.5.3 Architecture Remapping. Architecture remapping refers to the knowledge transfer about network architecture from a pre-existing model. To our best knowledge, this line of work is mainly used in weight-sharing neural network search (NAS). Formally, $\mathcal{F}$ denotes the architecture and $\boldsymbol{\omega}$ denotes the weight of $\mathcal{F}$. The goal of NAS is to find the optimal architecture $\mathcal{F}^*$ that produces the best performance on the test set:

$$\mathcal{F}^* = \arg \max_{\mathcal{F}} \text{Eval}(\{\mathcal{F}, \boldsymbol{\omega}\}; \mathcal{D}_{\text{test}}). \qquad (17)$$

Specifically, this type of NAS formulates the search space into an over-parameterized super-network, *e.g.*, modeling the search space as multiple repeatable cells [41, 131, 178]. When transferring the searched architecture to downstream tasks, direct architecture transfer, which stacks several searched cells to form a downstream model and then retrains it on the downstream data, is the current mainstream scheme. Canonical examples include DARTS [131] and its variants [34, 40, 77, 118, 251, 256]. Direct architecture transfer has shown impressive results on downstream tasks.

3.5.4 Discussion. Different from traditional transfer learning approaches, remapping tuning focuses on training a new downstream model isolated from the pre-existing model. Thus remapping tuning methods own their exclusive advantages. In this line of work, knowledge distillation involves training a smaller student model to mimic the output or intermediate features of a teacher model. This method is advantageous for efficient model compression as well as flexible student architecture designs. Weight remapping directly transfers model weights from a teacher network to a student network. This approach is beneficial for speed and efficiency, as transferring weights can be faster than retraining a new model from scratch. Architecture remapping focuses on transferring network architecture knowledge, often used in weight-sharing neural network search. This method enables the transferability of architectures discovered in one task to other tasks, accelerating the development of new models for various applications.

While remapping tuning offers many advantages, there are also some limitations for different approaches. For instance, knowledge distillation may lead to a loss of information from the teacher model. It can also be sensitive to hyperparameters, such as temperature and weighting of different losses. Weight remapping is a simple and effective solution without manually adding constraints. However, this type of work typically struggles to obtain a lightweight student network compared with knowledge distillation. Architecture remapping faces challenges in designing an effective search space for NAS, which can be complex and time-consuming. Additionally, NAS methods often

require significant computational resources to explore and evaluate a large number of candidate architectures, increasing the overall computational cost.

Their advantages and challenges also provide valuable insights for future advancements. For knowledge distillation, beyond learning from the pre-trained model's output, its high flexibility suggests the potential to incorporate grounded information from the downstream tasks for distillation. For example, combining the pre-trained model's output with a physics simulation could help guide the knowledge distillation process, resulting in an accurate and efficient student model for the downstream tasks [127]. Regarding weight remapping, a potential improvement is to combine with knowledge distillation to reduce the model size. As for architecture remapping, exploring stable and reusable modules from multiple pre-existing models could largely reduce the complexity of search space design, e.g., earlier layers of CNN are often reused for extracting lower-level visual features. This would help to flexibly and efficiently integrate various domain-specific models, based on the semantic associations among different domains.

4 VISUAL TUNING FUTURE

To date, the state of visual intelligence forms a transfer learning paradigm of pre-training and tuning, showing great promising performance on numerous benchmarks. Vision contributes a large portion of knowledge acquisition of human intelligence. However, due to the high dimensionality of vision data, the intelligence of machine vision suffers from a relatively small data scale in comparison to that of NLP and remains far behind the general human vision. The future promise of intelligent vision will be expanded beyond the competed benchmark datasets, realizing transformative impacts on more domains via a multidisciplinary coevolutionary process. On one hand, we expect that future pre-training techniques play the role of knowledge acquisition and storage in a "collection-labeling-training-feedback" cycle system. While future tuning is around how to make use of the learned knowledge through more diversified interactions beyond the prompts around conversational systems. Along the way to further understand the mechanisms of deep neural network models and even the human brain, we discuss the future works of vision from perspectives of pre-training and tuning techniques.

4.1 Advanced Pre-training

Previous works use supervised or self-supervised methods to guide models to learn representations of our visual and visual-text world. The supervised pre-training method is a mainstream practice of the traditional transfer learning paradigm [216, 287, 301], While self-supervised pre-training scales pre-trained models up to foundation models (introduced in Section 2.4). Although encouraging progress has been made, as data continues to accumulate, we expect that future pre-training techniques will be able to constantly scale up the model size and improve the capabilities of foundation models. Here, we discuss the future directions of model pre-training from three perspectives: data, models, and optimization.

4.1.1 Data. Quality data are the nourishment of foundation models. To realize the promise of future foundation models, it is expected to acquire more fundamental knowledge from the open-world multi-modal data with characteristics as follows:

- Increasing scales: Concerning the data volume, large vision models that learn knowledge from large-scale datasets are empirically proven effective for adapting to downstream tasks via tuning techniques. However, compared with human vision, existing large-scale vision datasets remain far from the amount of data that humans learn from. On the contrary, the situation in NLP can be different as large language models can be regarded as having a wide knowledge of the Internet, making them more capable in some NLP tasks, *i.e.*, chatGPT. To

ACM Comput. Surv., Vol. 1, No. 1, Article . Publication date: April 2024.

https://mc.manuscriptcentral.com/csur

scale up the data volume, multi-modal data (e.g., image, video, audio, and text), multi-source data (e.g., Internet, generative models such as NeRF [161] and Diffusion [190, 260]), and multi-sensor data (e.g., different types of cameras, biomarkers, and ambient sensors) can be considered for training large models.

- High quality: Before arbitrarily collecting large-scale data, determining what data and how much data are essential concerns. The newly collected data can be redundant or noisy, respectively leading to limited or even negative effects on the model. Chen *et al.* [33] introduced the diversity rule at the level of feature representation. However, there remains a lack of investigation into the quality at the data level for existing benchmark datasets, giving rise to research on topics such as out-of-distribution generalization and tolerance to noise (see Section 2.1.3). Further investigating measurable factors of data quality (e.g., 16 dimensions summarized in [56]) and their corresponding consequences on the large model can bring a large impact on the machine intelligence community. Findings will guide evidence-oriented data collection and effectively reduce the expensive labeling cost.
- Security and privacy are always the priority throughout the life cycle of the data, especially for domains such as healthcare and finance when interacting with large models on the cloud [24]. Issues around cloud computing can be grouped into four aspects: 1) users' control over the data, 2) authorized replication, 3) legal requirements, and 4) cloud subcontractors' processing [211]. Protective actions can be taken at the data level to prevent attacks such as re-identification, dataset reconstruction, and tracing [101].

4.1.2 Models. Given the multimodal, multi-source, multi-sensor data, vision large pre-trained models are expected to continuously accumulate knowledge from the new data in an interpretable and secure mechanism.

- Theoretical support: Training models with theoretical support from statistical and biological perspectives can make them more interpretable, explainable, and improvable. In the regime of large models, there are a certain number of recent works motivated by some theoretical definitions from the statistical perspective [138, 223, 263, 286], by which a generalization bound are used to promise efficient knowledge transfer. Except for the statistical aspect, biological and neuroscience discoveries also benefit the development of deep neural networks, which can provide more insights and inspire new ideas for future large vision models. Recent works [65, 114] discussed in Section 2.1.1 are mainly delayed from one to another as they are intended to explain each other's observed domain empirical realities instead of truly inspiring new ideas. Basic neural network connections are inspired by how brain neuron works, but it has not yet been known exactly how the human brain learns new knowledge. As such, it is also not clear if the knowledge acquired by existing large models via back-propagation can be effective. On one hand, we humans are sentient beings and acquire knowledge via multiple sensations: vision, sound, haptic, taste, etc. On the other hand, the human brain can be very efficient to activate just a small portion of neurons to complete a task, while existing foundation models do not. By far, the built intelligence is different from the most intelligent and efficient machine in the world (i.e., the human brain). Understanding the brain can be the next turning point (i.e., artificial general intelligence), which brings serious ethical issues.
- Continuously updating: As introduced in Section 2.2, a model can be featured with its domains and tasks with their feature space and label space. We expect that foundation models will scale up not only at the parameter aspect but also at aspects of domains and tasks. For a single domain, it can have multiple tasks. Models such as Gato [188] and Flamingo [1] are pre-trained with multiple tasks, where the former covers vision tasks while the latter even covers both NLP and vision tasks. For a single task, a classification task needs to handle

novel unseen classes, which is defined by an active learning paradigm (i.e., lifelong learning or continual learning). In contrast to batch learning where all training data is available at once, continual learning represents a family of methods that accumulate knowledge and learn continuously with data available in sequential order [182]. The future stronger vision and visual-language models will bring a more profound impact on other domains via the multidisciplinary coevolutionary process.

- Security: Aside from privacy issues at the data level, large foundation models (i.e., at the model level) can also be vulnerable to attacks. Foundation models allow users to easily plug and unplug via APIs, which raises security and privacy concerns. such as adversarial attacks and model-inversion. Kaissis *et al.* [101] introduced the advantages of federated learning and provided an outlook for future works. Although federated learning can mitigate data-level privacy issues, it can be vulnerable to adversarial attack [226]. Around privacy-preserving AI, the adversarial attack will attract more research in the near future.

4.1.3 Optimization. Current foundation models are generally optimized with back-propagation and reinforcement learning with human feedback. Optimization itself can be relying on hardware devices, hyperparameter configuration, and algorithms as follows:

- Hardware: Recent large models are trained with GPUs, which is unaffordable for general researchers or small companies. Fortunately, the newly released NVIDIA Hopper H100 GPU [37] supports FP8 format for accelerating compute-intensive Transformer models (around 9 times faster than previous A100 GPU for training), making trillion-parameter models within the reach of all researchers. While the inference speedup of H100 compared with A100 can be 30 times faster, making tuning a promising direction.
- Hyperparameter configuration: In machine learning, hyperparameters such as initial learning rate, batch size, and task-specific parameters often considerably impact performance. To avoid the manual process of trial-and-error, hyperparameter optimization is a sub-field of automated machine learning, which aims to identify a well-performing combination of hyper-parameters. Simple techniques are grid or random search. Recent advances in hyperparameter optimization are evolution strategies, Bayesian optimization, Hyperband, etc. [9].
- Algorithm: The combination of back-propagation and stochastic gradient descent remains the mainstream algorithm to make foundation models optimize towards some statistical goals (e.g., the probability that a picture is identified as a cat). Meanwhile, reinforcement learning with human feedback brings more raw human opinions, which can be some kind of human-machine interactions that align the pre-trained large models to more specific human desired tasks.

4.2 Tuning Techinques

As introduced in Section 3, recent developments in visual tuning techniques can be regarded as orig-inating from the prompt tuning of the NLP domain and working towards the PETL direction. Then a couple of adapter methods are proposed, showing better performance than visual prompt methods but lacking interpretability. (It is expected that visual tuning techniques will be implemented on more existing benchmarks, their reformed versions, and the emerging brand-new benchmarks, which will not be listed in this survey. Readers are recommended to refer to the benchmarks of their targeting domains.) The bias-tuning and LoRA methods further reduced the number of parameters, leading to direct parameter tuning methods via addition or decomposition. More recent works are grouped as remapping tuning, among which NAS-based methods [74, 249] show an even more aggressive PETL manner. These techniques provide exciting research foundations for developing future prompts, leading to better use of language and visual knowledge stored in large models

via guidance and interaction, respectively. We discuss three core progressive interaction aspects: interpretable prompt, conversational guidance, and diversified interactions, that researchers will discern explosive development as follows.

4.2.1 Interpretable Prompt. Prompt engineering will work from intuitive design to more understandable and interpretable directions. Existing text or visual prompts are more like implicit guidance at the high level, describing what is the downstream visual task. As introduced in Section 3.2, many works attempted to learn prompts to facilitate visual downstream tasks. Despite some progress, they suffered from poor interpretability, i.e., it remains difficult to understand what prompts the network has learned. For example, some works (e.g., VPT) learn unordered token-based prompts, which can not be visualized into an understandable prompt. Chen *et al.* [35] attempted to learn understandable prompts. Regarding other tuning techniques such as adapter, parameter-based, and remapping ones, they are also faced with the interpretability issue, as they intrinsically aim to reduce the number of tuned parameters for adapting the downstream tasks to the large model. Hence, future research should answer questions such as what are good text and vision prompts, and how to evaluate them throughout the learning pipeline (from the input side to the output side); the relationship between vision and text prompts, and in what situations visual and text prompts can be mutually replaced; How to design explicit, consistent, and logical prompts that enable a large model to adapt efficiently.

4.2.2 Conversational Guidance. We observe the development of visual tuning will lead to new jobs such as prompt engineers who have expertise in providing guidance to large-scale visual-language models such as Sora [144]. Multi-round conversational systems can provide a natural platform that guides models to adapt toward the desired task goals [244]. It is generally expected that vision models will homogenize with language models [1, 93, 183, 206, 207, 244, 266]. However, due to the fact that "a picture is worth a thousand words", the development of visual tuning is somehow behind the success of large language models (detailed in Section 2.1.2). Specifically, concerning data complexity and scenario diversity, the industrial applications of the vision domain (not common application scenarios such as autonomous vehicles, transportation recommendation [142], whether prediction [163], protein design [229], etc.) are highly demanding for customization based on the specific task requirements. Given a tumor detection task, a prompt engineer will select or design good segmentation samples in multiple rounds of conversation with the large models to improve some core steps of the task by referring to various agents [51] or tools [180] and eventually achieve acceptable results for production.

4.2.3 Diversified Interactions. In addition to the interaction in a conversational system with text and visual prompts, interactions in vision can be more diversified. Humans can gradually build up the evaluation standard themselves and then practice (i.e., learn or train for a model) towards higher standards. Existing self-learning models have not set up mechanisms with progressively improving goals. We expect, in the long run, universal AI or strong AI in a specific domain will evolve in the form of prompts, guidance, and diversified interactions. In recent works [238, 289], a segmentation sample is also used as a prompt to tell the model what task will be performed. Currently, interactions in image synthesis use text prompts and sketch images [190], enabling everyone to become a visual content creator. These visual interactions can be regarded as a kind of visual prompt based on the image. Image can represent a limited part of visual interaction scenarios, which can be regarded as tasks with static viewpoints but provide basic conditions for more plentiful visual interaction. Current image-based interactions via prompts are also known as in-context learning, which aims to mimic the efficient visual understanding of the human brain and intrinsically narrows the searching space of foundation models [238]. In addition to the simple aspect of navigating large

models to downstream tasks, there are more diversified interaction scenarios that provide plenty of egocentric visual interactions such as robots, drones, and bionic robot dogs [50, 228]. These video data provide interactive ecological environments that enable the development of human vision mentioned in Section 2.1.1. Although tuning foundation models pre-trained via self-supervised learning indicates a promising future direction, future visual interactions will rely on advanced pre-training techniques (knowledge accumulation techniques) that are beyond currently tested ones that are based on generating masked pixels or contrastive learning. Promising long-term directions that enable diversified interactions can involve emerging technologies such as brain-computer interface, quantum computing, event cameras, etc. This will lead to new generalization capabilities on top of the future "collection-labeling-training-feedback" cycle system.

5 CONCLUSION

This survey summarized visual tuning techniques, particularly focusing on the recent state of visual tuning in the coming regime of large models. Starting from fine-tuning, existing states of prompt tuning, adapter tuning, parameter tuning, and remapping tuning are systematically investigated and compared based on a comprehensive understanding of their technical details. Based on the expected emerging large models, future visual tuning directions are discussed from perspectives of prompt, guidance, interaction, and optimization. We hope this first survey on the latest state of visual tuning will offer a new perspective to researchers in the era of large models, facilitating their research based on a better understanding of the current state and grasping the future core research challenges.

6 ACKNOWLEDGMENTS

REFERENCES

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Preprint at https://arxiv.org/abs/2204.14198* (2022).

[2] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. 2017. Understanding of a convolutional neural network. In *Proc. ICET*. IEEE, 1–6.

[3] Reinald Kim Amplayo, Kang Min Yoo, and Sang-Woo Lee. 2022. Attribute Injection for Pretrained Language Models: A New Benchmark and An Efficient Method. In *Proc. COLING*. 1051–1064.

[4] Simone Angarano, Francesco Salvetti, Mauro Martini, and Marcello Chiaberge. 2022. Generative Adversarial Super-Resolution at the Edge with Knowledge Distillation. *Preprint at https://arxiv.org/abs/2209.03355* (2022).

[5] Anubhav Ashok, Nicholas Rhinehart, Fares Beainy, and Kris M Kitani. 2018. N2n learning: Network to network compression via policy gradient reinforcement learning. In *Proc. ICLR*.

[6] Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep?. In *Proc. NeurIPS*, Vol. 27.

[7] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Trans. Patt. Anal. Mach. Intell.* 35, 8 (2013), 1798–1828.

[8] Hugo Berg, Siobhan Mackenzie Hall, Yash Bhalgat, Wonsuk Yang, Hannah Rose Kirk, Aleksandar Shtedritski, and Max Bain. 2022. A prompt array keeps the bias away: Debiasing vision-language models with adversarial learning. *Preprint at https://arxiv.org/abs/2203.11933* (2022).

[9] Bernd Bischl, Martin Binder, Michel Lang, Tobias Pielok, Jakob Richter, Stefan Coors, Janek Thomas, Theresa Ullmann, Marc Becker, Anne-Laure Boulesteix, et al. 2021. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* (2021), e1484.

[10] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models.

ACM Comput. Surv., Vol. 1, No. 1, Article . Publication date: April 2024.

https://mc.manuscriptcentral.com/csur

Preprint at https://arxiv.org/abs/2108.07258 (2021).

[11] Benjamin Bowman, Alessandro Achille, Luca Zancato, Matthew Trager, Pramuditha Perera, Giovanni Paolini, and Stefano Soatto. 2023. \A-la-carte Prompt Tuning (APT): Combining Distinct Data Via Composable Prompting. *Preprint at https://arxiv.org/abs/2302.07994* (2023).

[12] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Proc. NeurIPS*, Vol. 33. 1877–1901.

[13] XB Bruce, Yan Liu, Keith CC Chan, and Chang Wen Chen. 2024. EGCN++: A New Fusion Strategy for Ensemble Learning in Skeleton-Based Rehabilitation Exercise Assessment. *IEEE Trans. Patt. Anal. Mach. Intell.* 01 (2024), 1–16.

[14] XB Bruce, Yan Liu, Xiang Zhang, Sheng-hua Zhong, and Keith CC Chan. 2022. Mmnet: A model-based multimodal network for human action recognition in rgb-d videos. *IEEE Trans. Patt. Anal. Mach. Intell.* (2022).

[15] Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. 2022. Differentially Private Bias-Term only Fine-tuning of Foundation Models. *Preprint at https://arxiv.org/abs/2210.00036* (2022).

[16] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *ACM SIGKDD*. 535–541.

[17] Adrian Bulat and Georgios Tzimiropoulos. 2022. Language-Aware Soft Prompting for Vision & Language Foundation Models. *Preprint at https://arxiv.org/abs/2210.01115* (2022).

[18] Han Cai, Tianyao Chen, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Efficient architecture search by network transformation. In *Proc. AAAI*, Vol. 32.

[19] Han Cai, Jiacheng Yang, Weinan Zhang, Song Han, and Yong Yu. 2018. Path-level network transformation for efficient architecture search. In *Proc. ICML*. PMLR, 678–687.

[20] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. 2010. Brief: Binary robust independent elementary features. In *Proc. ECCV*. Springer, 778–792.

[21] Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. CVPR*. 6299–6308.

[22] Jianlong Chang, Xinbang Zhang, Yiwen Guo, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. 2019. DATA: Differentiable ArchiTecture Approximation. In *Proc. NeurIPS*. 874–884.

[23] Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. 2021. Cross-layer distillation with semantic calibration. In *Proc. AAAI*, Vol. 35. 7028–7036.

[24] Deyan Chen and Hong Zhao. 2012. Data security and privacy protection issues in cloud computing. In *2012 international conference on computer science and electronics engineering*, Vol. 1. IEEE, 647–651.

[25] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. 2022. Prompt Learning with Optimal Transport for Vision-Language Models. *Preprint at https://arxiv.org/abs/2210.01253* (2022).

[26] Hao Chen, Ran Tao, Han Zhang, Yidong Wang, Wei Ye, Jindong Wang, Guosheng Hu, and Marios Savvides. 2022. Conv-Adapter: Exploring Parameter Efficient Transfer Learning for ConvNets. *Preprint at https://arxiv.org/abs/2208.07463* (2022).

[27] Hanting Chen, Yunhe Wang, Chang Xu, Chao Xu, and Dacheng Tao. 2020. Learning student networks via feature embedding. *IEEE Trans. Neur. Netw. Learn. Syst.* 32, 1 (2020), 25–35.

[28] Haoran Chen, Zuxuan Wu, and Yu-Gang Jiang. 2022. Multi-Prompt Alignment for Multi-source Unsupervised Domain Adaptation. *Preprint at https://arxiv.org/abs/2209.15210* (2022).

[29] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. 2020. Generative pretraining from pixels. In *Proc. ICML*. PMLR, 1691–1703.

[30] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. 2021. Distilling knowledge via knowledge review. In *Proc. CVPR*. 5008–5017.

[31] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. 2022. AdaptFormer: Adapting Vision Transformers for Scalable Visual Recognition. *Preprint at https://arxiv.org/abs/2205.13535* (2022).

[32] Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. 2016. Net2net: Accelerating learning via knowledge transfer. In *Proc. ICLR*.

[33] Tianlong Chen, Zhenyu Zhang, Yu Cheng, Ahmed Awadallah, and Zhangyang Wang. 2022. The principle of diversity: Training stronger vision transformers calls for reducing all levels of redundancy. In *Proc. CVPR*. 12020–12030.

[34] Xin Chen, Lingxi Xie, Jun Wu, and Qi Tian. 2019. Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. In *Proc. CVPR*. 1294–1303.

[35] Xiang Chen, Ningyu Zhang, Lei Li, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Good Visual Guidance Makes A Better Extractor: Hierarchical Visual Prefix for Multimodal Entity and Relation Extraction. *North American Chapter of the Association for Computational Linguistics* (2022).

[36] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. 2022. Vision transformer adapter for dense predictions. *Preprint at https://arxiv.org/abs/2205.08534* (2022).

[37] Jack Choquette. 2023. NVIDIA Hopper H100 GPU: Scaling Performance. *IEEE Micro* (2023).

[38] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. 2021. Twins: Revisiting the design of spatial attention in vision transformers. In *Proc. NeurIPS*, Vol. 34. 9355–9366.

[39] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. 2021. Conditional positional encodings for vision transformers. *Preprint at https://arxiv.org/abs/2102.10882* (2021).

[40] Xiangxiang Chu, Xiaoxing Wang, Bo Zhang, Shun Lu, Xiaolin Wei, and Junchi Yan. 2021. Darts-: robustly stepping out of performance collapse without indicators. In *Proc. ICLR*.

[41] Xiangxiang Chu, Bo Zhang, and Ruijun Xu. 2021. Fairnas: Rethinking evaluation fairness of weight sharing neural architecture search. In *Proc. CVPR*. 12239–12248.

[42] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. 2000. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications* 21, 4 (2000), 1253–1278.

[43] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. 2023. Scaling Vision Transformers to 22 Billion Parameters. *Preprint at https://arxiv.org/abs/2302.05442* (2023).

[44] Frederik Michel Dekking, Cornelis Kraaikamp, Hendrik Paul Lopuhaä, and Ludolf Erwin Meester. 2005. *A Modern Introduction to Probability and Statistics: Understanding why and how*. Vol. 488. Springer.

[45] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 CVPR*. IEEE, 248–255.

[46] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence* (2023), 1–16.

[47] Bowen Dong, Pan Zhou, Shuicheng Yan, and Wangmeng Zuo. 2022. LPT: Long-tailed Prompt Tuning for Image Classification. *Preprint at https://arxiv.org/abs/2210.01033* (2022).

[48] Runpei Dong, Zekun Qi, Linfeng Zhang, Junbo Zhang, Jianjian Sun, Zheng Ge, Li Yi, and Kaisheng Ma. 2022. Autoencoders as Cross-Modal Teachers: Can Pretrained 2D Image Transformers Help 3D Representation Learning? *Preprint at https://arxiv.org/abs/2212.08320* (2022).

[49] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. ICLR*.

[50] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. 2023. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378* (2023).

[51] Zane Durante, Qiuyuan Huang, Naoki Wake, Ran Gong, Jae Sung Park, Bidipta Sarkar, Rohan Taori, Yusuke Noda, Demetri Terzopoulos, Yejin Choi, et al. 2024. Agent ai: Surveying the horizons of multimodal interaction. *arXiv preprint arXiv:2401.03568* (2024).

[52] Ali Edalati, Marzieh Tahaei, Ivan Kobyzev, Vahid Partovi Nia, James J Clark, and Mehdi Rezagholizadeh. 2022. KronA: Parameter Efficient Tuning with Kronecker Adapter. *Preprint at https://arxiv.org/abs/2212.10650* (2022).

[53] Constantin Eichenberg, Sidney Black, Samuel Weinbach, Letitia Parcalabescu, and Anette Frank. 2021. MAGMA–Multimodal Augmentation of Generative Models through Adapter-based Finetuning. *Preprint at https://arxiv.org/abs/2112.05253* (2021).

[54] Thomas Elsken, Jan-Hendrik Metzen, and Frank Hutter. 2018. Simple and efficient architecture search for convolutional neural networks. In *Proc. ICLR, Workshop Track*.

[55] Beyza Ermis, Giovanni Zappella, Martin Wistuba, Aditya Rawal, and Cédric Archambeau. 2022. Continual Learning With Transformers for Image Classification. In *Proc. CVPR Workshops*. 3774–3781.

[56] Philip Evans. 2006. Scaling and assessment of data quality. *Acta Crystallographica Section D: Biological Crystallography* 62, 1 (2006), 72–82.

[57] Jiemin Fang, Yuzhu Sun, Kangjian Peng, Qian Zhang, Yuan Li, Wenyu Liu, and Xinggang Wang. 2020. Fast neural network adaptation via parameter remapping and architecture search. In *Proc. ICLR*.

[58] Jiemin Fang, Yuzhu Sun, Qian Zhang, Kangjian Peng, Yuan Li, Wenyu Liu, and Xinggang Wang. 2020. FNA++: Fast network adaptation via parameter remapping and architecture search. *IEEE Trans. Patt. Anal. Mach. Intell.* 43, 9 (2020), 2990–3004.

[59] Christoph Feichtenhofer. 2020. X3d: Expanding architectures for efficient video recognition. In *Proc. CVPR*. 203–213.

[60] Chin-Lun Fu, Zih-Ching Chen, Yun-Ru Lee, and Hung-yi Lee. 2022. AdapterBias: Parameter-efficient Token-dependent Representation Shift for Adapters in NLP Tasks. *Preprint at https://arxiv.org/abs/2205.00305* (2022).

[61] Yulu Gan, Xianzheng Ma, Yihang Lou, Yan Bai, Renrui Zhang, Nian Shi, and Lin Luo. 2022. Decorate the Newcomers: Visual Domain Prompt for Continual Test Time Adaptation. *Preprint at https://arxiv.org/abs/2212.04145* (2022).

[62] Kaifeng Gao, Long Chen, Hanwang Zhang, Jun Xiao, and Qianru Sun. 2023. Compositional Prompt Tuning with Motion Cues for Open-vocabulary Video Relation Detection. *Preprint at https://arxiv.org/abs/2302.00268* (2023).

ACM Comput. Surv., Vol. 1, No. 1, Article . Publication date: April 2024.

https://mc.manuscriptcentral.com/csur

[63] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. 2021. Clip-adapter: Better vision-language models with feature adapters. *Preprint at https://arxiv.org/abs/2110.04544* (2021).

[64] Yunhe Gao, Xingjian Shi, Yi Zhu, Hao Wang, Zhiqiang Tang, Xiong Zhou, Mu Li, and Dimitris N Metaxas. 2022. Visual Prompt Tuning for Test-time Domain Adaptation. *Preprint at https://arxiv.org/abs/2210.04831* (2022).

[65] James J Gibson. 2014. *The ecological approach to visual perception: classic edition.* Psychology press.

[66] Yunchao Gong, Liu Liu, Ming Yang, and Lubomir Bourdev. 2014. Compressing deep convolutional networks using vector quantization. *Preprint at https://arxiv.org/abs/1412.6115* (2014).

[67] Eleonora Grassucci, Aston Zhang, and Danilo Comminiello. 2022. PHNNs: Lightweight neural networks via parameterized hypercomplex convolutions. *IEEE Trans. Neur. Netw. Learn. Syst.* (2022).

[68] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. 2021. Open-vocabulary object detection via vision and language knowledge distillation. *Preprint at https://arxiv.org/abs/2104.13921* (2021).

[69] Yushuo Guan, Pengyu Zhao, Bingxuan Wang, Yuanxing Zhang, Cong Yao, Kaigui Bian, and Jian Tang. 2020. Differentiable feature aggregation search for knowledge distillation. In *ECCV 16.* Springer, 469–484.

[70] Jianyuan Guo, Kai Han, Yunhe Wang, Han Wu, Xinghao Chen, Chunjing Xu, and Chang Xu. 2021. Distilling object detectors via decoupled features. In *Proc. CVPR.* 2154–2164.

[71] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. 2022. Cmt: Convolutional neural networks meet vision transformers. In *Proc. CVPR.* 12175–12185.

[72] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. 2022. A survey on vision transformer. *IEEE Trans. Patt. Anal. Mach. Intell.* (2022).

[73] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. 2021. Transformer in transformer. In *Proc. NeurIPS,* Vol. 34. 15908–15919.

[74] Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. 2021. Dynamic neural networks: A survey. *IEEE Trans. Patt. Anal. Mach. Intell.* (2021).

[75] Tianxiang Hao, Hui Chen, Yuchen Guo, and Guiguang Ding. 2023. Consolidator: Mergable Adapter with Group Connections for Visual Adaptation. In *Proc. ICLR.* https://openreview.net/forum?id=J_Cja7cpgW

[76] Zhiwei Hao, Jianyuan Guo, Ding Jia, Kai Han, Yehui Tang, Chao Zhang, Han Hu, and Yunhe Wang. 2022. Learning Efficient Vision Transformers via Fine-Grained Manifold Distillation. In *Proc. NeurIPS.*

[77] Chaoyang He, Haishan Ye, Li Shen, and Tong Zhang. 2020. Milenas: Efficient neural architecture search via mixed-level reformulation. In *Proc. CVPR.* 11993–12002.

[78] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a Unified View of Parameter-Efficient Transfer Learning. In *Proc. ICLR.* https://openreview.net/forum?id=0RDcd5Axok

[79] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proc. CVPR.* 16000–16009.

[80] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proc. CVPR.* 770–778.

[81] Xuehai He, Chunyuan Li, Pengchuan Zhang, Jianwei Yang, and Xin Eric Wang. 2022. Parameter-efficient fine-tuning for vision transformers. *Preprint at https://arxiv.org/abs/2203.16329* (2022).

[82] Xuehai He, Diji Yang, Weixi Feng, Tsu-Jui Fu, Arjun Akula, Varun Jampani, Pradyumna Narayana, Sugato Basu, William Yang Wang, and Xin Eric Wang. 2022. CPL: Counterfactual Prompt Learning for Vision and Language Models. *Preprint at https://arxiv.org/abs/2210.10362* (2022).

[83] Roei Herzig, Ofir Abramovich, Elad Ben-Avraham, Assaf Arbelle, Leonid Karlinsky, Ariel Shamir, Trevor Darrell, and Amir Globerson. 2022. PromptonomyViT: Multi-Task Prompt Learning Improves Video Transformers using Synthetic Scene Data. *Preprint at https://arxiv.org/abs/2212.04821* (2022).

[84] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *Preprint at https://arxiv.org/abs/1503.02531* (2015).

[85] Zejiang Hou, Julian Salazar, and George Polovets. 2022. Meta-learning the difference: Preparing large language models for efficient adaptation. *Transactions of the Association for Computational Linguistics* 10 (2022), 1249–1265.

[86] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proc. ICML.* PMLR, 2790–2799.

[87] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *Proc. ICLR.* https://openreview.net/forum?id=nZeVKeeFYf9

[88] Junjie Hu, Chenyou Fan, Hualie Jiang, Xiyue Guo, Yuan Gao, Xiangyong Lu, and Tin Lun Lam. 2021. Boosting light-weight depth estimation via knowledge distillation. *Preprint at https://arxiv.org/abs/2105.06143* (2021).

[89] Shishuai Hu, Zehui Liao, and Yong Xia. 2022. ProSFDA: Prompt Learning based Source-free Domain Adaptation for Medical Image Segmentation. *Preprint at https://arxiv.org/abs/2211.11514* (2022).

[90] Zilong Huang, Youcheng Ben, Guozhong Luo, Pei Cheng, Gang Yu, and Bin Fu. 2021. Shuffle transformer: Rethinking spatial shuffle for vision transformer. *Preprint at https://arxiv.org/abs/2106.03650* (2021).

[91] David H Hubel and Torsten N Wiesel. 1959. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology* 148, 3 (1959), 574.

[92] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 2013. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Trans. Patt. Anal. Mach. Intell.* 36, 7 (2013), 1325–1339.

[93] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proc. ICML*. PMLR, 4904–4916.

[94] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. 2022. Visual prompt tuning. In *Proc. ECCV*. Springer, 709–727.

[95] Haojun Jiang, Jianke Zhang, Rui Huang, Chunjiang Ge, Zanlin Ni, Jiwen Lu, Jie Zhou, Shiji Song, and Gao Huang. 2022. Cross-Modal Adapter for Text-Video Retrieval. *Preprint at https://arxiv.org/abs/2211.09623* (2022).

[96] Ziyu Jiang, Tianlong Chen, Xuxi Chen, Yu Cheng, Luowei Zhou, Lu Yuan, Ahmed Awadallah, and Zhangyang Wang. 2022. DnA: Improving Few-Shot Transfer Learning with Low-Rank Decomposition and Alignment. In *Proc. ECCV*. Springer, 239–256.

[97] Ziyu Jiang, Xuxi Chen, Xueqin Huang, Xianzhi Du, Denny Zhou, and Zhangyang Wang. 2022. Back Razor: Memory-Efficient Transfer Learning by Self-Sparsified Backpropagation. In *Proc. NeurIPS*.

[98] Shibo Jie and Zhi-Hong Deng. 2022. Convolutional bypasses are better vision transformer adapters. *Preprint at https://arxiv.org/abs/2207.07039* (2022).

[99] Shibo Jie and Zhi-Hong Deng. 2022. FacT: Factor-Tuning for Lightweight Adaptation on Vision Transformer. *Preprint at https://arxiv.org/abs/2212.03145* (2022).

[100] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. 2022. Prompting visual-language models for efficient video understanding. In *Proc. ECCV*. Springer, 105–124.

[101] Georgios A Kaissis, Marcus R Makowski, Daniel Rückert, and Rickmer F Braren. 2020. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence* 2, 6 (2020), 305–311.

[102] Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. 2021. Compacter: Efficient low-rank hypercomplex adapter layers. In *Proc. NeurIPS*, Vol. 34. 1022–1035.

[103] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. 2017. The kinetics human action video dataset. *Preprint at https://arxiv.org/abs/1705.06950* (2017).

[104] Yan Ke and Rahul Sukthankar. 2004. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proc. CVPR*, Vol. 2. IEEE, II–II.

[105] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. 2022. Transformers in vision: A survey. *ACM computing surveys (CSUR)* 54, 10s (2022), 1–41.

[106] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. 2022. Maple: Multi-modal prompt learning. *Preprint at https://arxiv.org/abs/2210.03117* (2022).

[107] Jangho Kim, SeongUk Park, and Nojun Kwak. 2018. Paraphrasing complex network: Network compression via factor transfer. In *Proc. NeurIPS*, Vol. 31.

[108] Konwoo Kim, Michael Laskin, Igor Mordatch, and Deepak Pathak. 2021. How to Adapt Your Large-Scale Vision-and-Language Model. *Preprint at https://openreview.net/forum?id=EhwEUb2ynIa* (2021).

[109] Kwanyoung Kim, Yujin Oh, and Jong Chul Ye. 2023. ZegOT: Zero-shot Segmentation Through Optimal Transport of Text Prompts. *Preprint at https://arxiv.org/abs/2301.12171* (2023).

[110] Minsu Kim, Hyung-Il Kim, and Yong Man Ro. 2023. Prompt Tuning of Deep Neural Networks for Speaker-adaptive Visual Speech Recognition. *Preprint at https://arxiv.org/abs/2302.08102* (2023).

[111] Parth Kothari, Danya Li, Yuejiang Liu, and Alexandre Alahi. 2022. Motion style transfer: Modular low-rank adaptation for deep motion forecasting. *Preprint at https://arxiv.org/abs/2211.03165* (2022).

[112] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60, 6 (2017), 84–90.

[113] Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. 2021. Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution. In *Proc. ICLR*.

[114] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. *Science* 350, 6266 (2015), 1332–1338.

[115] Andrew Lavin and Scott Gray. 2016. Fast algorithms for convolutional neural networks. In *Proc. CVPR*. 4013–4021.

[116] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.

ACM Comput. Surv., Vol. 1, No. 1, Article . Publication date: April 2024.

https://mc.manuscriptcentral.com/csur

[117] Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. 2022. Efficient self-supervised vision transformers for representation learning. In *Proc. ICLR*.

[118] Guohao Li, Guocheng Qian, Itzel C Delgadillo, Matthias Muller, Ali Thabet, and Bernard Ghanem. 2020. Sgas: Sequential greedy architecture search. In *Proc. CVPR*. 1620–1630.

[119] Quanquan Li, Shengying Jin, and Junjie Yan. 2017. Mimicking very efficient network for object detection. In *Proc. CVPR*. 6356–6364.

[120] Tianjiao Li, Qiuhong Ke, Hossein Rahmani, Rui En Ho, Henghui Ding, and Jun Liu. 2021. Else-net: Elastic semantic network for continual action recognition from skeleton data. In *Proc. CVPR*. 13434–13443.

[121] Tao Li, Lei Tan, Zhehao Huang, Qinghua Tao, Yipeng Liu, and Xiaolin Huang. 2022. Low dimensional trajectory hypothesis is true: Dnns can be trained in tiny subspaces. *IEEE Trans. Patt. Anal. Mach. Intell.* (2022).

[122] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. 2022. Exploring plain vision transformer backbones for object detection. In *Proc. ECCV*. Springer, 280–296.

[123] Yanghao Li, Saining Xie, Xinlei Chen, Piotr Dollar, Kaiming He, and Ross Girshick. 2021. Benchmarking detection transfer learning with vision transformers. *Preprint at https://arxiv.org/abs/2111.11429* (2021).

[124] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. 2021. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE Trans. Neur. Netw. Learn. Syst.* (2021).

[125] Dongze Lian, Zhou Daquan, Jiashi Feng, and Xinchao Wang. 2022. Scaling & Shifting Your Features: A New Baseline for Efficient Model Tuning. In *Proc. NeurIPS*, Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (Eds.). https://openreview.net/forum?id=XtyeppctGgc

[126] Hanwen Liang, Shifeng Zhang, Jiacheng Sun, Xingqiu He, Weiran Huang, Kechen Zhuang, and Zhenguo Li. 2019. Darts+: Improved differentiable architecture search with early stopping. *Preprint at https://arxiv.org/abs/1909.06035* (2019).

[127] Junfan Lin, Jianlong Chang, Lingbo Liu, Guanbin Li, Liang Lin, Qi Tian, and Chang-wen Chen. 2023. Being Comes from Not-being: Open-vocabulary Text-to-Motion Generation with Wordless Training. In *Proc. CVPR*.

[128] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Proc. ECCV*. Springer, 740–755.

[129] Yan-Bo Lin, Yi-Lin Sung, Jie Lei, Mohit Bansal, and Gedas Bertasius. 2022. Vision Transformers are Parameter-Efficient Audio-Visual Learners. *Preprint at https://arxiv.org/abs/2212.07983* (2022).

[130] Zhaojiang Lin, Andrea Madotto, and Pascale Fung. 2020. Exploring Versatile Generative Language Model Via Parameter-Efficient Transfer Learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. 441–459.

[131] Hanxiao Liu, Karen Simonyan, and Yiming Yang. 2019. Darts: Differentiable architecture search. In *Proc. ICLR*.

[132] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. 2019. Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Trans. Patt. Anal. Mach. Intell.* 42, 10 (2019), 2684–2701.

[133] Lingbo Liu, Jiaqi Chen, Hefeng Wu, Tianshui Chen, Guanbin Li, and Liang Lin. 2020. Efficient crowd counting via structured knowledge transfer. In *ACM MM*. 2645–2654.

[134] Lingbo Liu, Jiaqi Chen, Hefeng Wu, Guanbin Li, Chenglong Li, and Liang Lin. 2021. Cross-modal collaborative representation learning and a large-scale rgbt benchmark for crowd counting. In *Proc. CVPR*. 4823–4833.

[135] Lingbo Liu, Bruce XB Yu, Jianlong Chang, Qi Tian, and Chang-Wen Chen. 2022. Prompt-matched semantic segmentation. *Preprint at https://arxiv.org/abs/2208.10159* (2022).

[136] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Preprint at https://arxiv.org/abs/2107.13586* (2021).

[137] Shiwei Liu and Zhangyang Wang. 2023. Ten Lessons We Have Learned in the New" Sparseland": A Short Handbook for Sparse Neural Network Researchers. *Preprint at https://arxiv.org/abs/2302.02596* (2023).

[138] Xiaobin Liu and Shiliang Zhang. 2022. Who is closer: A computational method for domain gap evaluation. *Patt. Recog.* 122 (2022), 108293.

[139] Yufan Liu, Jiajiong Cao, Bing Li, Chunfeng Yuan, Weiming Hu, Yangxi Li, and Yunqiang Duan. 2019. Knowledge distillation via instance relationship graph. In *Proc. CVPR*. 7096–7104.

[140] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. 2019. Structured knowledge distillation for semantic segmentation. In *Proc. CVPR*. 2604–2613.

[141] Yu Liu, Xuhui Jia, Mingxing Tan, Raviteja Vemulapalli, Yukun Zhu, Bradley Green, and Xiaogang Wang. 2020. Search to distill: Pearls are everywhere but not the eyes. In *Proc. CVPR*. 7539–7548.

[142] Yang Liu, Cheng Lyu, Zhiyuan Liu, and Jinde Cao. 2021. Exploring a large-scale multi-modal transportation recommendation system. *Transportation Research Part C: Emerging Technologies* 126 (2021), 103070.

[143] Yifan Liu, Changyong Shu, Jingdong Wang, and Chunhua Shen. 2020. Structured knowledge distillation for dense prediction. *IEEE Trans. Patt. Anal. Mach. Intell.* (2020).

[144] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, et al. 2024. Sora: A Review on Background, Technology, Limitations, and Opportunities of Large Vision Models. *arXiv preprint arXiv:2402.17177* (2024).

[145] Yen-Cheng Liu, Chih-Yao Ma, Junjiao Tian, Zijian He, and Zsolt Kira. 2022. Polyhistor: Parameter-Efficient Multi-Task Adaptation for Dense Vision Tasks. *Preprint at https://arxiv.org/abs/2210.03265* (2022).

[146] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proc. CVPR.* 10012–10022.

[147] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. 2022. Video swin transformer. In *Proc. CVPR.* 3202–3211.

[148] Jochem Loedeman, Maarten C Stol, Tengda Han, and Yuki M Asano. 2022. Prompt Generation Networks for Efficient Adaptation of Frozen Vision Transformers. *Preprint at https://arxiv.org/abs/2210.06466* (2022).

[149] Haoyu Lu, Mingyu Ding, Yuqi Huo, Guoxing Yang, Zhiwu Lu, Masayoshi Tomizuka, and Wei Zhan. 2023. UniAdapter: Unified Parameter-Efficient Transfer Learning for Cross-modal Modeling. *Preprint at https://arxiv.org/abs/2302.06605* (2023).

[150] Gen Luo, Minglang Huang, Yiyi Zhou, Xiaoshuai Sun, Guannan Jiang, Zhiyu Wang, and Rongrong Ji. 2023. Towards Efficient Visual Adaption via Structural Re-parameterization. *Preprint at https://arxiv.org/abs/2302.08106* (2023).

[151] Gen Luo, Yiyi Zhou, Xiaoshuai Sun, Yan Wang, Liujuan Cao, Yongjian Wu, Feiyue Huang, and Rongrong Ji. 2022. Towards lightweight transformer via group-wise transformation for vision-and-language tasks. *TIP* 31 (2022), 3386–3398.

[152] Xiao Luo, Haixin Wang, Daqing Wu, Chong Chen, Minghua Deng, Jianqiang Huang, and Xian-Sheng Hua. 2023. A survey on deep hashing methods. *ACM Transactions on Knowledge Discovery from Data* 17, 1 (2023), 1–50.

[153] Chengcheng Ma, Yang Liu, Jiankang Deng, Lingxi Xie, Weiming Dong, and Changsheng Xu. 2023. Understanding and Mitigating Overfitting in Prompt Tuning for Vision-Language Models. *TCSVT* (2023).

[154] Teli Ma, Shijie Geng, Mengmeng Wang, Jing Shao, Jiasen Lu, Hongsheng Li, Peng Gao, and Yu Qiao. 2021. A simple long-tailed recognition baseline via vision-language model. *Preprint at https://arxiv.org/abs/2111.14745* (2021).

[155] Zeyu Ma, Yuhang Guo, Xiao Luo, Chong Chen, Minghua Deng, Wei Cheng, and Guangming Lu. 2022. DHWP: Learning High-Quality Short Hash Codes Via Weight Pruning. In *ICASSP*. IEEE, 4783–4787.

[156] Chengzhi Mao, Scott Geng, Junfeng Yang, Xin Wang, and Carl Vondrick. 2022. Understanding Zero-Shot Adversarial Robustness for Large-Scale Models. *Preprint at https://arxiv.org/abs/2212.07016* (2022).

[157] Yuning Mao, Lambert Mathias, Rui Hou, Amjad Almahairi, Hao Ma, Jiawei Han, Wen-tau Yih, and Madian Khabsa. 2021. Unipelt: A unified framework for parameter-efficient language model tuning. *Preprint at https://arxiv.org/abs/2110.07577* (2021).

[158] Imad Eddine MAROUF, Enzo Tartaglione, and Stéphane Lathuilière. 2023. Tiny Adapters for Vision Transformers. https://openreview.net/forum?id=V0Vo9eW2nzL

[159] David Marr. 2010. *Vision: A computational investigation into the human representation and processing of visual information.* MIT press.

[160] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. 2023. Latent-nerf for shape-guided generation of 3d shapes and textures. In *Proc. CVPR.* 12663–12673.

[161] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.

[162] Mahdi Namazifar, Devamanyu Hazarika, and Dilek Hakkani-Tur. 2023. Role of Bias Terms in Dot-Product Attention. *Preprint at https://arxiv.org/abs/2302.08626* (2023).

[163] Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K Gupta, and Aditya Grover. 2023. ClimaX: A foundation model for weather and climate. *Preprint at https://arxiv.org/abs/2301.10343* (2023).

[164] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. 2022. Expanding language-image pretrained models for general video recognition. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*. Springer, 1–18.

[165] Xing Nie, Bolin Ni, Jianlong Chang, Gaomeng Meng, Chunlei Huo, Zhaoxiang Zhang, Shiming Xiang, Qi Tian, and Chunhong Pan. 2022. Pro-tuning: Unified prompt tuning for vision tasks. *Preprint at https://arxiv.org/abs/2207.14381* (2022).

[166] Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. 2022. ST-Adapter: Parameter-Efficient Image-to-Video Transfer Learning. In *Proc. NeurIPS.*

[167] Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. 2022. ST-Adapter: Parameter-Efficient Image-to-Video Transfer Learning for Action Recognition. *Preprint at https://arxiv.org/abs/2206.13559* (2022).

ACM Comput. Surv., Vol. 1, No. 1, Article . Publication date: April 2024.

https://mc.manuscriptcentral.com/csur

[168] Omiros Pantazis, Gabriel Brostow, Kate Jones, and Oisin Mac Aodha. 2022. SVL-Adapter: Self-Supervised Adapter for Vision-Language Pretrained Models. *Preprint at https://arxiv.org/abs/2210.03794* (2022).

[169] Pinelopi Papalampidi and Mirella Lapata. 2022. Hierarchical3D Adapters for Long Video-to-text Summarization. *Preprint at https://arxiv.org/abs/2210.04829* (2022).

[170] Jihye Park, Soohyun Kim, Sunwoo Kim, Jaejun Yoo, Youngjung Uh, and Seungryong Kim. 2022. LANIT: Language-Driven Image-to-Image Translation for Unlabeled Data. *Preprint at https://arxiv.org/abs/2208.14889* (2022).

[171] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. 2019. Relational knowledge distillation. In *Proc. CVPR*. 3967–3976.

[172] Nikolaos Passalis, Maria Tzelepi, and Anastasios Tefas. 2020. Heterogeneous knowledge distillation using information flow modeling. In *Proc. CVPR*. 2339–2348.

[173] William Peebles and Saining Xie. 2023. Scalable diffusion models with transformers. In *Proc. CVPR*. 4195–4205.

[174] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. 2019. Correlation congruence for knowledge distillation. In *Proc. CVPR*. 5007–5016.

[175] Fang Peng, Xiaoshan Yang, and Changsheng Xu. 2022. SgVA-CLIP: Semantic-guided Visual Adapting of Vision-Language Models for Few-shot Image Classification. *Preprint at https://arxiv.org/abs/2211.16191* (2022).

[176] Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. Adapterhub: A framework for adapting transformers. *arXiv preprint arXiv:2007.07779* (2020).

[177] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. 2021. Meta pseudo labels. In *Proc. CVPR*. 11557–11568.

[178] Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. 2018. Efficient neural architecture search via parameters sharing. In *Proc. ICML*. PMLR, 4095–4104.

[179] Angelo Porrello, Luca Bergamini, and Simone Calderara. 2020. Robust re-identification by multiple views knowledge distillation. In *Proc. ECCV*. Springer, 93–110.

[180] Yujia Qin, Shengding Hu, Yankai Lin, Weize Chen, Ning Ding, Ganqu Cui, Zheni Zeng, Yufei Huang, Chaojun Xiao, Chi Han, et al. 2023. Tool learning with foundation models. *arXiv preprint arXiv:2304.08354* (2023).

[181] Ziran Qin, Mingbao Lin, and Weiyao Lin. 2023. Low-Rank Winograd Transformation for 3D Convolutional Neural Networks. *Preprint at https://arxiv.org/abs/2301.11180* (2023).

[182] Haoxuan Qu, Hossein Rahmani, Li Xu, Bryan Williams, and Jun Liu. 2021. Recent advances of continual learning in computer vision: An overview. *Preprint at https://arxiv.org/abs/2109.11369* (2021).

[183] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *Proc. ICML*. PMLR, 8748–8763.

[184] Jun Rao, Xv Meng, Liang Ding, Shuhan Qi, and Dacheng Tao. 2022. Parameter-efficient and student-friendly knowledge distillation. *Preprint at https://arxiv.org/abs/2205.15308* (2022).

[185] Yongming Rao, Wenliang Zhao, Jie Zhou, and Jiwen Lu. 2022. AMixer: Adaptive Weight Mixing for Self-attention Free Vision Transformers. In *Proc. ECCV*. Springer, 50–67.

[186] Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Proc. NeurIPS*.

[187] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2018. Efficient parametrization of multi-domain deep neural networks. In *Proc. CVPR*. 8119–8127.

[188] 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. 2022. A generalist agent. *Preprint at https://arxiv.org/abs/2205.06175* (2022).

[189] Félix Remigereau, Djebril Mekhazni, Sajjad Abdoli, Rafael MO Cruz, Eric Granger, et al. 2022. Knowledge distillation for multi-target domain adaptation in real-time person re-identification. In *ICIP*. IEEE, 3853–3557.

[190] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proc. CVPR*. 10684–10695.

[191] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. Fitnets: Hints for thin deep nets. In *Proc. ICLR*.

[192] Amir Rosenfeld and John K Tsotsos. 2018. Incremental learning through deep adaptation. *IEEE Trans. Patt. Anal. Mach. Intell.* 42, 3 (2018), 651–663.

[193] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. 2011. ORB: An efficient alternative to SIFT or SURF. In *Proc. ICCV*. IEEE, 2564–2571.

[194] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2023. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proc. CVPR*. 22500–22510.

[195] Stuart J Russell. 2010. *Artificial intelligence a modern approach*. Pearson Education, Inc.

[196] Kuniaki Saito, Kihyuk Sohn, Xiang Zhang, Chun-Liang Li, Chen-Yu Lee, Kate Saenko, and Tomas Pfister. 2022. Prefix conditioning unifies language and label supervision. *Preprint at https://arxiv.org/abs/2206.01125* (2022).

[197] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proc. CVPR*. 4510–4520.

[198] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Trans. Sign. Proces.* 45, 11 (1997), 2673–2681.

[199] Mohit Sharma, Claudio Fantacci, Yuxiang Zhou, Skanda Koppula, Nicolas Heess, Jon Scholz, and Yusuf Aytar. 2023. Lossless Adaptation of Pretrained Vision Models For Robotic Manipulation. In *Proc. ICLR*. https://openreview.net/forum?id=5IND3TXJRb-

[200] Sheng Shen, Shijia Yang, Tianjun Zhang, Bohan Zhai, Joseph E Gonzalez, Kurt Keutzer, and Trevor Darrell. 2022. Multitask Vision-Language Prompt Tuning. *Preprint at https://arxiv.org/abs/2211.11720* (2022).

[201] Yifeng Shi, Feng Lv, Xinliang Wang, Chunlong Xia, Shaojie Li, Shujie Yang, Teng Xi, and Gang Zhang. 2023. Open-transmind: A new baseline and benchmark for 1st foundation model challenge of intelligent transportation. In *Proc. CVPR Workshop*. 6327–6334.

[202] Erica K Shimomoto, Edison Marrese-Taylor, Hiroya Takamura, Ichiro Kobayashi, Hideki Nakayama, and Yusuke Miyao. 2022. Towards Parameter-Efficient Integration of Pre-Trained Language Models In Temporal Video Grounding. *Preprint at https://arxiv.org/abs/2209.13359* (2022).

[203] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. 2022. Test-time prompt tuning for zero-shot generalization in vision-language models. *Preprint at https://arxiv.org/abs/2209.07511* (2022).

[204] Aliaksandra Shysheya, John F Bronskill, Massimiliano Patacchiola, Sebastian Nowozin, and Richard E Turner. 2023. FiT: Parameter Efficient Few-shot Transfer Learning for Personalized and Federated Image Classification. In *Proc. ICLR*. https://openreview.net/forum?id=9aokcgBVIj1

[205] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *Preprint at https://arxiv.org/abs/1409.1556* (2014).

[206] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. 2022. Flava: A foundational language and vision alignment model. In *Proc. CVPR*. 15638–15650.

[207] Mannat Singh, Laura Gustafson, Aaron Adcock, Vinicius de Freitas Reis, Bugra Gedik, Raj Prateek Kosaraju, Dhruv Mahajan, Ross Girshick, Piotr Dollár, and Laurens van der Maaten. 2022. Revisiting Weakly Supervised Pre-Training of Visual Perception Models. In *Proc. CVPR*. 804–814.

[208] Kihyuk Sohn, Yuan Hao, José Lezama, Luisa Polania, Huiwen Chang, Han Zhang, Irfan Essa, and Lu Jiang. 2022. Visual Prompt Tuning for Generative Transfer Learning. *Preprint at https://arxiv.org/abs/2210.00990* (2022).

[209] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. 2017. Revisiting unreasonable effectiveness of data in deep learning era. In *Proc. ICCV*. 843–852.

[210] Ximeng Sun, Ping Hu, and Kate Saenko. 2022. Dualcoop: Fast adaptation to multi-label recognition with limited annotations. *Preprint at https://arxiv.org/abs/2206.09541* (2022).

[211] Yunchuan Sun, Junsheng Zhang, Yongping Xiong, and Guangyu Zhu. 2014. Data security and privacy in cloud computing. *International Journal of Distributed Sensor Networks* 10, 7 (2014), 190903.

[212] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. 2022. Lst: Ladder side-tuning for parameter and memory efficient transfer learning. In *Proc. NeurIPS*.

[213] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. 2022. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proc. CVPR*. 5227–5237.

[214] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proc. CVPR*. 1–9.

[215] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proc. CVPR*. 2818–2826.

[216] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. 2018. A survey on deep transfer learning. In *International conference on artificial neural networks*. Springer, 270–279.

[217] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proc. ICML*. PMLR, 6105–6114.

[218] Ming Tao, Bing-Kun Bao, Hao Tang, and Changsheng Xu. 2023. GALIP: Generative Adversarial CLIPs for Text-to-Image Synthesis. *Preprint at https://arxiv.org/abs/2301.12959* (2023).

[219] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. Efficient transformers: A survey. *Comput. Surveys* 55, 6 (2022), 1–28.

[220] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. 2022. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Preprint at https://arxiv.org/abs/2203.12602* (2022).

[221] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2021. Training data-efficient image transformers & distillation through attention. In *Proc. ICML*. PMLR, 10347–10357.

ACM Comput. Surv., Vol. 1, No. 1, Article . Publication date: April 2024.

https://mc.manuscriptcentral.com/csur

[222] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proc. ICCV*. 4489–4497.

[223] Nilesh Tripuraneni, Michael Jordan, and Chi Jin. 2020. On the theory of transfer learning: The importance of task diversity. In *Proc. NeurIPS*, Vol. 33. 7852–7862.

[224] Koki Tsubota, Hiroaki Akutsu, and Kiyoharu Aizawa. 2023. Universal Deep Image Compression via Content-Adaptive Optimization with Adapters. In *Proc. WACV*. 2529–2538.

[225] Cheng-Hao Tu, Zheda Mai, and Wei-Lun Chao. 2022. Visual Query Tuning: Towards Effective Usage of Intermediate Representations for Parameter and Memory Efficient Transfer Learning. *Preprint at https://arxiv.org/abs/2212.03220* (2022).

[226] Dmitrii Usynin, Alexander Ziller, Marcus Makowski, Rickmer Braren, Daniel Rueckert, Ben Glocker, Georgios Kaissis, and Jonathan Passerat-Palmbach. 2021. Adversarial interference and its mitigations in privacy-preserving collaborative machine learning. *Nature Machine Intelligence* 3, 9 (2021), 749–758.

[227] Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. 2022. DyLoRA: Parameter Efficient Tuning of Pre-trained Models using Dynamic Search-Free Low-Rank Adaptation. *Preprint at https://arxiv.org/abs/2210.07558* (2022).

[228] Sai Vemprala, Rogerio Bonatti, Arthur Bucker, and Ashish Kapoor. 2023. Chatgpt for robotics: Design principles and model abilities. (2023).

[229] Robert Verkuil, Ori Kabeli, Yilun Du, Basile IM Wicky, Lukas F Milles, Justas Dauparas, David Baker, Sergey Ovchin-nikov, Tom Sercu, and Alexander Rives. 2022. Language models generalize beyond natural proteins. *bioRxiv* (2022), 2022–12.

[230] Feng Wang, Manling Li, Xudong Lin, Hairong Lv, Alexander G Schwing, and Heng Ji. 2022. Learning to Decompose Visual Features with Latent Textual Prompts. *Preprint at https://arxiv.org/abs/2210.04287* (2022).

[231] Haixin Wang, Jianlong Chang, Xiao Luo, Jinan Sun, Zhouchen Lin, and Qi Tian. 2023. LION: Implicit Vision Prompt Tuning. *Preprint at https://arxiv.org/abs/2303.09992* (2023).

[232] Haixin Wang, Xinlong Yang, Jianlong Chang, Dian Jin, Jinan Sun, Shikun Zhang, Xiao Luo, and Qi Tian. 2023. Mode Approximation Makes Good Vision-Language Prompts. *arXiv preprint arXiv:2305.08381* (2023).

[233] Haixin Wang, Tianhao Zhang, Muzhi Yu, Jinan Sun, Wei Ye, Chen Wang, and Shikun Zhang. 2020. Stacking networks dynamically for image restoration based on the Plug-and-Play framework. In *Proc. ECCV*. Springer, 446–462.

[234] Shijie Wang, Jianlong Chang, Haojie Li, Zhihui Wang, Wanli Ouyang, and Qi Tian. 2023. Open-Set Fine-Grained Retrieval via Prompting Vision-Language Evaluator. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 19381–19391.

[235] Shijie Wang, Jianlong Chang, Zhihui Wang, Haojie Li, Wanli Ouyang, and Qi Tian. 2022. Fine-grained Retrieval Prompt Tuning. *Preprint at https://arxiv.org/abs/2207.14465* (2022).

[236] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. 2022. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *Preprint at https://arxiv.org/abs/2208.10442* (2022).

[237] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. 2021. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proc. CVPR*. 568–578.

[238] Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. 2022. Images Speak in Images: A Generalist Painter for In-Context Visual Learning. *Preprint at https://arxiv.org/abs/2212.02499* (2022).

[239] Yabin Wang, Zhiwu Huang, and Xiaopeng Hong. 2022. S-Prompts Learning with Pre-trained Transformers: An Occam's Razor for Domain Incremental Learning. *Preprint at https://arxiv.org/abs/2207.12819* (2022).

[240] Yiran Wang, Xingyi Li, Min Shi, Ke Xian, and Zhiguo Cao. 2021. Knowledge distillation for fast and accurate monocular depth estimation on mobile devices. In *Proc. CVPR*. 2457–2465.

[241] Yukang Wang, Wei Zhou, Tao Jiang, Xiang Bai, and Yongchao Xu. 2020. Intra-class feature variation distillation for semantic segmentation. In *Proc. ECCV*. Springer, 346–362.

[242] Ziyi Wang, Xumin Yu, Yongming Rao, Jie Zhou, and Jiwen Lu. 2022. P2p: Tuning pre-trained image models for point cloud analysis with point-to-pixel prompting. *Preprint at https://arxiv.org/abs/2208.02812* (2022).

[243] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. 2016. Learning structured sparsity in deep neural networks. In *Proc. NeurIPS*, Vol. 29.

[244] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. 2023. Visual ChatGPT: Talking, Drawing and Editing with Visual Foundation Models. *Preprint at https://arxiv.org/abs/2303.04671* (2023).

[245] Chen Henry Wu, Saman Motamed, Shaunak Srivastava, and Fernando De la Torre. 2022. Generative visual prompt: Unifying distributional control of pre-trained generative models. In *Proc. NeurIPS*.

[246] Jiarun Wu and Qingliang Chen. 2022. Pruning Adapters with Lottery Ticket. *Algorithms* 15, 2 (2022), 63.

[247] Junyang Wu, Xianhang Li, Chen Wei, Huiyu Wang, Alan Yuille, Yuyin Zhou, and Cihang Xie. 2022. Unleashing the Power of Visual Prompting At the Pixel Level. *Preprint at https://arxiv.org/abs/2212.10556* (2022).

[248] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. 2023. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proc. CVPR*. 7623–7633.

[249] Lingxi Xie, Xin Chen, Kaifeng Bi, Longhui Wei, Yuhui Xu, Lanfei Wang, Zhengsu Chen, An Xiao, Jianlong Chang, Xiaopeng Zhang, et al. 2021. Weight-sharing neural architecture search: A battle to shrink the optimization gap. *ACM Computing Surveys (CSUR)* 54, 9 (2021), 1–37.

[250] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. 2018. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proc. ECCV*. 305–321.

[251] Sirui Xie, Hehui Zheng, Chunxiao Liu, and Liang Lin. 2019. SNAS: stochastic neural architecture search. In *Proc. ICLR*.

[252] Yinghui Xing, Qirui Wu, De Cheng, Shizhou Zhang, Guoqiang Liang, and Yanning Zhang. 2022. Class-aware visual prompt tuning for vision-language pre-trained model. *Preprint at https://arxiv.org/abs/2208.08340* (2022).

[253] Chengming Xu, Siqian Yang, Yabiao Wang, Zhanxiong Wang, Yanwei Fu, and Xiangyang Xue. 2023. Exploring Efficient Few-shot Adaptation for Vision Transformers. *Preprint at https://arxiv.org/abs/2301.02419* (2023).

[254] Kunran Xu, Lai Rui, Yishi Li, and Lin Gu. 2020. Feature normalized knowledge distillation for image classification. In *Proc. ECCV*. Springer, 664–680.

[255] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. 2023. Side Adapter Network for Open-Vocabulary Semantic Segmentation. *Preprint at https://arxiv.org/abs/2302.12242* (2023).

[256] Yuhui Xu, Lingxi Xie, Xiaopeng Zhang, Xin Chen, Guo-Jun Qi, Qi Tian, and Hongkai Xiong. 2020. Pc-darts: Partial channel connections for memory-efficient architecture search. In *Proc. ICLR*.

[257] Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proc. AAAI*, Vol. 32.

[258] Chuanguang Yang, Helong Zhou, Zhulin An, Xue Jiang, Yongjun Xu, and Qian Zhang. 2022. Cross-image relational knowledge distillation for semantic segmentation. In *Proc. CVPR*. 12319–12328.

[259] Jing Yang, Brais Martinez, Adrian Bulat, and Georgios Tzimiropoulos. 2020. Knowledge distillation via adaptive instance normalization. *Preprint at https://arxiv.org/abs/2003.04289* (2020).

[260] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. 2023. Diffusion models: A comprehensive survey of methods and applications. *Comput. Surveys* 56, 4 (2023), 1–39.

[261] Taojiannan Yang, Yi Zhu, Yusheng Xie, Aston Zhang, Chen Chen, and Mu Li. 2023. AIM: Adapting Image Models for Efficient Video Action Recognition. In *Proc. ICLR*. https://openreview.net/forum?id=CIoSZ_HKHS7

[262] Xingyi Yang, Daquan Zhou, Songhua Liu, Jingwen Ye, and Xinchao Wang. 2022. Deep model reassembly. In *Proc. NeurIPS*, Vol. 35. 25739–25753.

[263] Haotian Ye, Chuanlong Xie, Tianle Cai, Ruichen Li, Zhenguo Li, and Liwei Wang. 2021. Towards a theoretical framework of out-of-distribution generalization. In *Proc. NeurIPS*, Vol. 34. 23519–23531.

[264] Bruce XB Yu, Jianlong Chang, Lingbo Liu, Qi Tian, and Chang Wen Chen. 2022. Towards a Unified View on Visual Parameter-Efficient Transfer Learning. *Preprint at https://arxiv.org/abs/2210.00788* (2022).

[265] Bruce XB Yu, Zhi Zhang, Yongxu Liu, Sheng-hua Zhong, Yan Liu, and Chang Wen Chen. 2023. Gla-gcn: Global-local adaptive graph convolutional network for 3d human pose estimation from monocular video. In *Proc. ICCV*. 8818–8829.

[266] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models. *Preprint at https://arxiv.org/abs/2205.01917* (2022).

[267] Shengming Yu, Zhaopeng Dou, and Shengjin Wang. 2023. Prompting and Tuning: A Two-Stage Unsupervised Domain Adaptive Person Re-identification Method on Vision Transformer Backbone. *Tsinghua Science and Technology* 28, 4 (2023), 799–810.

[268] Zitong Yu, Rizhao Cai, Yawen Cui, Xin Liu, Yongjian Hu, and Alex Kot. 2023. Rethinking Vision Transformer and Masked Autoencoder in Multimodal Face Anti-Spoofing. *Preprint at https://arxiv.org/abs/2302.05744* (2023).

[269] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. 2021. Incorporating convolution designs into visual transformers. In *Proc. CVPR*. 579–588.

[270] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. 2021. Florence: A new foundation model for computer vision. *Preprint at https://arxiv.org/abs/2111.11432* (2021).

[271] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. 2021. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proc. CVPR*. 558–567.

[272] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. 2022. Volo: Vision outlooker for visual recognition. *IEEE Trans. Patt. Anal. Mach. Intell.* (2022).

[273] Sha Yuan, Hanyu Zhao, Shuai Zhao, Jiahong Leng, Yangxiao Liang, Xiaozhi Wang, Jifan Yu, Xin Lv, Zhou Shao, Jiaao He, et al. 2022. A Roadmap for Big Model. *Preprint at https://arxiv.org/abs/2203.14101* (2022).

ACM Comput. Surv., Vol. 1, No. 1, Article . Publication date: April 2024.

https://mc.manuscriptcentral.com/csur

[274] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models. In *Proc. ACL (Volume 2: Short Papers)*. 1–9.

[275] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. 2022. Unified vision and language prompt learning. *Preprint at https://arxiv.org/abs/2210.07225* (2022).

[276] Aston Zhang, Yi Tay, SHUAI Zhang, Alvin Chan, Anh Tuan Luu, Siu Hui, and Jie Fu. 2021. Beyond Fully-Connected Layers with Quaternions: Parameterization of Hypercomplex Multiplications with $1/n$ Parameters. In *Proc. ICLR*.

[277] Bowen Zhang, Xiaojie Jin, Weibo Gong, Kai Xu, Zhao Zhang, Peng Wang, Xiaohui Shen, and Jiashi Feng. 2023. Multimodal Video Adapter for Parameter Efficient Video Text Retrieval. *Preprint at https://arxiv.org/abs/2301.07868* (2023).

[278] Jian-Wei Zhang, Yifan Sun, Yi Yang, and Wei Chen. 2022. Feature-Proxy Transformer for Few-Shot Segmentation. *Preprint at https://arxiv.org/abs/2210.06908* (2022).

[279] Linfeng Zhang and Kaisheng Ma. 2021. Improve object detection with feature-based knowledge distillation: Towards accurate and efficient detectors. In *Proc. ICLR*.

[280] Renrui Zhang, Hanqiu Deng, Bohao Li, Wei Zhang, Hao Dong, Hongsheng Li, Peng Gao, and Yu Qiao. 2022. Collaboration of Pre-trained Models Makes Better Few-shot Learner. *Preprint at https://arxiv.org/abs/2209.12255* (2022).

[281] Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. 2021. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *Preprint at https://arxiv.org/abs/2111.03930* (2021).

[282] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. 2022. Pointclip: Point cloud understanding by clip. In *Proc. CVPR*. 8552–8562.

[283] Xinbang Zhang, Jianlong Chang, Yiwen Guo, Gaofeng Meng, Shiming Xiang, Zhouchen Lin, and Chunhong Pan. 2021. DATA: Differentiable ArchiTecture Approximation With Distribution Guided Sampling. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 9 (2021), 2905–2920.

[284] Yiman Zhang, Hanting Chen, Xinghao Chen, Yiping Deng, Chunjing Xu, and Yunhe Wang. 2021. Data-free knowledge distillation for image super-resolution. In *Proc. CVPR*. 7852–7861.

[285] Yue Zhang, Hongliang Fei, Dingcheng Li, Tan Yu, and Ping Li. 2022. Prompting through Prototype: A Prototype-based Prompt Learning on Pretrained Vision-Language Models. *Preprint at https://arxiv.org/abs/2210.10841* (2022).

[286] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. 2019. Bridging theory and algorithm for domain adaptation. In *Proc. ICML*. PMLR, 7404–7413.

[287] Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning. *IEEE Trans. Know. and Data Engin.* (2021).

[288] Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. 2022. Neural Prompt Search. *Preprint at https://arxiv.org/abs/2206.04673* (2022).

[289] Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. 2023. What Makes Good Examples for Visual In-Context Learning? *Preprint at https://arxiv.org/abs/2301.13670* (2023).

[290] Zhengkun Zhang, Wenya Guo, Xiaojun Meng, Yasheng Wang, Yadao Wang, Xin Jiang, Qun Liu, and Zhenglu Yang. 2022. Hyperpelt: Unified parameter-efficient language model tuning for both language and vision-and-language tasks. *Preprint at https://arxiv.org/abs/2203.03878* (2022).

[291] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. 2022. Decoupled knowledge distillation. In *Proc. CVPR*. 11953–11962.

[292] Cairong Zhao, Yubin Wang, Xinyang Jiang, Yifei Shen, Kaitao Song, Dongsheng Li, and Duoqian Miao. 2022. Learning Domain Invariant Prompt for Vision-Language Models. *Preprint at https://arxiv.org/abs/2212.04196* (2022).

[293] Zhaohui Zheng, Rongguang Ye, Ping Wang, Dongwei Ren, Wangmeng Zuo, Qibin Hou, and Ming-Ming Cheng. 2022. Localization distillation for dense object detection. In *Proc. CVPR*. 9407–9416.

[294] Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. 2023. A Comprehensive Survey on Pretrained Foundation Models: A History from BERT to ChatGPT. *Preprint at https://arxiv.org/abs/2302.09419* (2023).

[295] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. Conditional prompt learning for vision-language models. In *Proc. CVPR*. 16816–16825.

[296] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. Learning to prompt for vision-language models. *IJCV* 130, 9 (2022), 2337–2348.

[297] Sheng Zhou, Yucheng Wang, Defang Chen, Jiawei Chen, Xin Wang, Can Wang, and Jiajun Bu. 2021. Distilling holistic knowledge with graph neural networks. In *Proc. CVPR*. 10387–10396.

[298] Ziqin Zhou, Bowen Zhang, Yinjie Lei, Lingqiao Liu, and Yifan Liu. 2022. ZegCLIP: Towards Adapting CLIP for Zero-shot Semantic Segmentation. *Preprint at https://arxiv.org/abs/2212.03588* (2022).

[299] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. 2022. Prompt-aligned gradient for prompt tuning. *Preprint at https://arxiv.org/abs/2205.14865* (2022).

Visual Tuning 37

[300] Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyao Zeng, Shanghang Zhang, and Peng Gao. 2022. PointCLIP V2: Adapting CLIP for Powerful 3D Open-world Learning. *Preprint at https://arxiv.org/abs/2211.11682* (2022).

[301] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proc. IEEE* 109, 1 (2020), 43–76.