Zoo-Tuning: Adaptive Transfer from a Zoo of Models

Yang Shu [*1] **Zhi Kou** [*1] **Zhangjie Cao** [1] **Jianmin Wang** [1] **Mingsheng Long** [1]

Abstract

With the development of deep networks on various large-scale datasets, a large zoo of pretrained models are available. When transferring from a model zoo, applying classic single-model-based transfer learning methods to each source model suffers from high computational cost and cannot fully utilize the rich knowledge in the zoo. We propose *Zoo-Tuning* to address these challenges, which learns to adaptively transfer the parameters of pretrained models to the target task. With the learnable channel alignment layer and adaptive aggregation layer, Zoo-Tuning *adaptively aggregates channel aligned pretrained parameters to derive the target model*, which simultaneously promotes knowledge transfer and adapts source models to downstream tasks. The adaptive aggregation substantially reduces the computation cost at both training and inference. We further propose lite Zoo-Tuning with the temporal ensemble of batch average gating values to reduce the storage cost at the inference time. We evaluate our approach on a variety of tasks, including reinforcement learning, image classification, and facial landmark detection. Experiment results demonstrate that the proposed adaptive transfer learning approach can more effectively and efficiently transfer knowledge from a zoo of models.

1. Introduction

Transfer learning leverages knowledge from existing datasets to improve the learning on the target task (Pan & Yang, 2009). With deep learning achieving state-of-the-art performance on various machine learning tasks (LeCun et al., 2015; Goodfellow et al., 2016), deep transfer learning has attracted more attention in recent years (Tan et al., 2018). A standard deep transfer learning paradigm is to leverage models pretrained on large-scale datasets (Russakovsky et al., 2015) and fine-tune the model on the target task (Yosinski et al., 2014), which is demonstrated to be a simple and effective solution in real-world applications and empirical studies (Kornblith et al., 2019).

However, most of the existing fine-tuning approaches only transfer from a single pretrained deep network to the target task (Xuhong et al., 2018; Li et al., 2019; Chen et al., 2019). With the increasing number of large-scale datasets in various fields, we usually have access to a zoo of deep models pretrained on various datasets with different methods such as supervised learning (He et al., 2016), self-supervised learning (Dosovitskiy et al., 2014) and unsupervised learning (Chen et al., 2020). Therefore, the large diverse model zoo calls for a new problem setting, *Transfer Learning from a Zoo of Models*, which aims to transfer knowledge from a model zoo to promote the learning of the target task.

The new problem setting introduces new challenges: (1) How to decide the extent of knowledge transfer from different pretrained models? The different pretrained models contain a diverse range of knowledge. Some pretrained models are more related to the target tasks, while some models are unrelated to or even negatively influence the target learning. For example, although it is common to initialize models with parameters trained on ImageNet, empirical evidence has shown that this practice offers little benefit to applications in medical imaging (Raghu et al., 2019). Therefore, it is important to decide the correct models to transfer from. (2) How to aggregate the knowledge from different pretrained models? The diverse pretrained models can be complementary to each other, which serve as a more complete knowledge base for the target task. Thus, a model aggregating mechanism is required to integrate knowledge from multiple pretrained models, which further improves the target task. Directly transferring each model to the target and assembling the networks may aggregate knowledge but suffers from the large training and inference cost linearly increasing with the size of the model zoo.

In this paper, we propose *Zoo-Tuning*, an effective and efficient solution that enables adaptive transfer from a zoo of models to downstream tasks. We decide to transfer model parameters to avoid the high computation burden of forwarding data through all the pretrained models. We first

[*]Equal contribution [1]School of Software, BNRist, Tsinghua University, Beijing, China. E-mail: Yang Shu (shuy18@mails.tsinghua.edu.cn). Correspondence to: Mingsheng Long <mingsheng@tsinghua.edu.cn>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

align channels of different pretrained model parameters by a channel alignment layer since pretrained models are trained on diverse datasets. We then aggregate the pretrained model parameters with the weights controlled by a gating network. The channel alignment layer and the aggregation module are learned, and the source pretrained models are tuned both by the target loss signals to adaptively fit the target task. We further develop a lite version of Zoo-Tuning with a unified gating value for all data in the target task computed by the temporal ensemble of average gating values of each data batch. Lite Zoo-Tuning further reduces the computational and storage cost at inference time, which serves as an option to trade-off between efficiency and performance.

The contribution of the paper can be summarized as:

- We propose Zoo-Tuning, an adaptive transfer method to enable transfer learning from a zoo of models. Zoo-Tuning aligns channels of source pretrained models and learns data-dependent gating networks to aggregate source model parameters, which transfers knowledge from a zoo of source pretrained models with a low computation burden. All the modules are learned, and the pretrained models are tuned both by the target loss signals to fit into the target task adaptively.
- We develop a lite version of Zoo-Tuning by the temporal ensemble of average gating values of each data batch, which further saves the inference and storage cost and serves as an optional method to trade-off between efficiency and performance.
- We conduct experiments on a variety of tasks, including reinforcement learning, image classification, and facial landmark detection. Experimental results demonstrate that the proposed method outperforms the extension of single model transfer methods and model zoo transfer methods while remains efficient.

2. Related Work

Transfer Learning. Transfer learning is a machine learning paradigm to transfer knowledge from source domains to improve the learning of the target task (Pan & Yang, 2009). A promising way to leverage knowledge in pretrained models is to use features extracted by the networks (Oquab et al., 2014; Donahue et al., 2014) or fine-tune from pretrained networks (Agrawal et al., 2014; Girshick et al., 2014). To study the transferability of a pretrained model, Yosinski et al. (2014) experimentally quantify the generality versus specificity of neurons in each layer of a deep convolutional neural network. Recently, many empirical studies consider the effect of transferring from a pretrained model on various downstream tasks and scenarios, such as classification (Kornblith et al., 2019; Zhai et al., 2019), few-shot learning (Ramalho et al., 2019), medical imaging (Raghu et al., 2019), object detection and instance segmentation (He et al., 2019). Some works also seek to fully utilize knowledge of the pretrained model in fine-tuning, such as regularizing the parameters (Xuhong et al., 2018), distilling intermediate features (Li et al., 2019) and penalizing small eigenvalues (Chen et al., 2019). However, it remains unclear how to extend single model transfer learning techniques to transfer learning from a zoo of models, where simply assembling single transfer models is fairly inefficient.

Transfer from Multiple Models. In the face of a zoo of multiple models, recent works seek to predict the transferability of pretrained models to select the best model in the zoo (Tran et al., 2019; Bao et al., 2019; Nguyen et al., 2020). These methods suffer from two shortcomings: The selection of the best model may be inaccurate, which hurts the performance. Besides, only one single best model is utilized, which wastes other rich knowledge in the whole model zoo. Some works leverage prior knowledge by inserting features from previously learned models (Rusu et al., 2016; Liu et al., 2019). These methods need to pass the input data through all models during training or even inference time, which may cause high computational and memory costs. Furthermore, these methods use pretrained models to guide the learning of a student model without tuning and adapting the whole zoo to the target. Guidance without adaptation may fail when the downstream tasks are more complex or less similar to the pretrained tasks.

Conditional Computation. Our method is also related to conditional computation (Davis & Arel, 2013; Cho & Bengio, 2014), where parts of the network are active on a per-example basis. Eigen et al. (2013) introduce the idea of using multiple mixture-of-experts (Masoudnia & Ebrahimpour, 2014) with their own gating networks as parts of a deep model. Shazeer et al. (2017) introduce sparsely-gated mixture-of-experts layers to form outrageously large neural networks. Different from these methods which increase model capacity to absorb sufficiently large data by mixing outputs of sub-networks, we aim to transfer knowledge in source models by adaptively aggregating model parameters.

3. Approach

In this section, we first introduce the problem setting of transfer learning from a zoo of models. Then we introduce our Zoo-Tuning approach, which consists of channel alignment and adaptive aggregation. We further propose a more efficient Lite Zoo-Tuning approach. Finally, we provide theoretical analysis on the computation cost of zoo tuning.

3.1. Transfer Learning from a Zoo of Models

The most common transfer learning scenario considers only one single pretrained model M at hand to serve the tar-

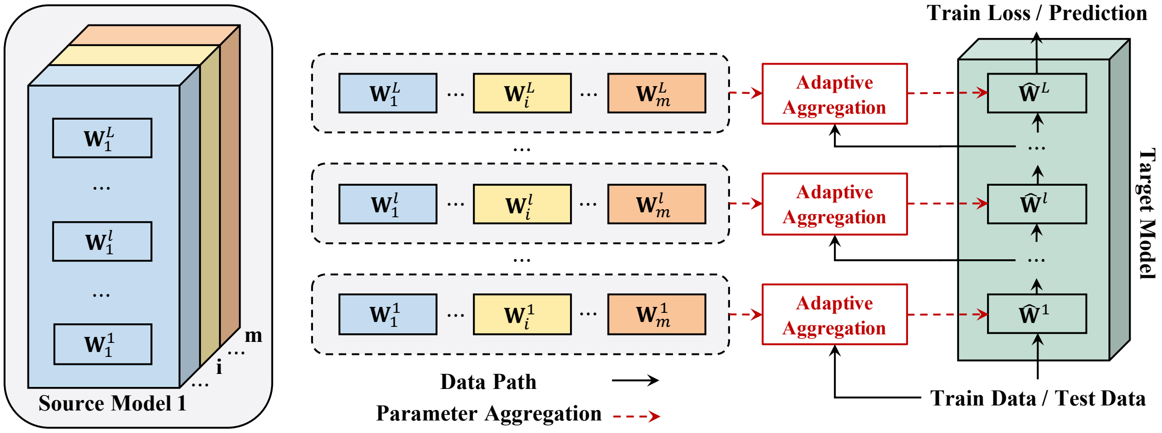


Figure 1. The framework of our proposed method. We derive the target model $\widehat{\mathbf{W}}$ by aggregating the parameters of the source models $\mathbf{W}_i$ in each layer, controlled by the learnable adaptive aggregation modules based on the input data. During training, the adaptive modules are trained, and the source parameters are tuned to transfer to the target task. After training, the tuned source models are aggregated depending on each query data for inference.

get task $\mathcal{D} = \{(x, y)\}$. In *transfer learning from a zoo of models*, we consider a more complicated situation where we have a zoo of pretrained models $\mathcal{M} = \{M_1, M_2, \cdots, M_m\}$. This problem is challenging in two ways: (1) The diverse pretrained models hold different relationships to the target tasks, which needs transferring knowledge from different pretrained models to different extents; (2) Different models are pretrained on various data and thus store different knowledge, which may be complementary to each other to solve the downstream tasks. How to aggregate knowledge from various pretrained models is an essential but difficult problem for model zoo transfer learning.

In this paper, we consider the situation that different models in the zoo have the same architecture but are trained with different *data*, *tasks*, or *pretraining algorithms*. This assumption of the same architecture is reasonable and has its value in practice since architectures such as ResNet (He et al., 2016) can be widely used in various datasets and tasks, and diverse pretrained models of these architectures with rich source knowledge are provided for use. It is easier and more reliable to apply these models with the same simple and familiar architectures, especially on new problems. Besides, The same architecture enables more effective layer-wise knowledge transfer, which is hard to realize on different architectures. A more relaxed situation where models have arbitrary architectures would be interesting and challenging to explore for future work.

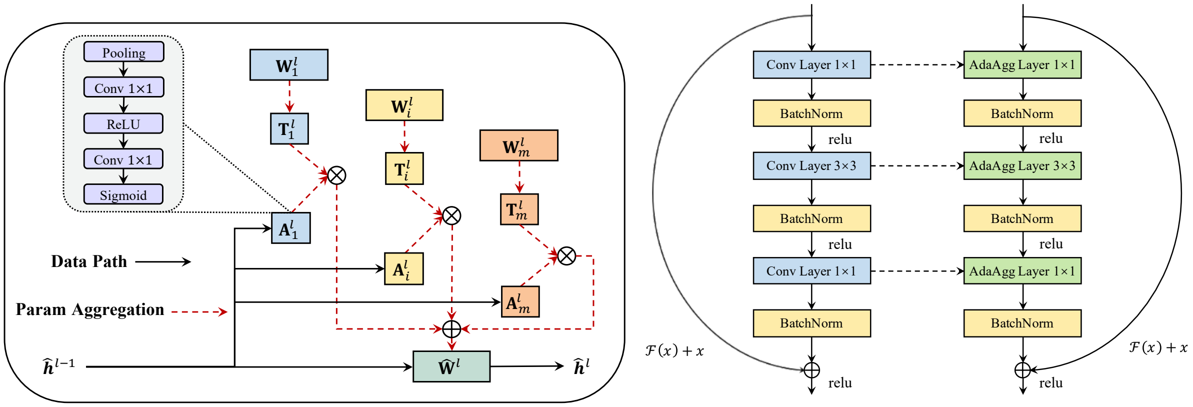
3.2. Zoo-Tuning

We address the problem of transfer learning from a zoo of models by Zoo-Tuning. The framework is shown in Figure 1. Zoo-Tuning enables knowledge transfer from multiple models by adaptively aggregating source model parameters in each layer, based on the input data, to form the target model. The adaptive aggregation consists of channel alignment and gating modules to control the extent of each model in transfer learning. As the adaptive aggregation mechanism is lightweight and the target data pass through the derived target model instead of all source models, the proposed approach only introduces similar inference time to a single model, which is computationally efficient. We further propose a lite version of Zoo-Tuning to reduce the storage cost.

Channel Alignment. Different models are separately trained on diverse datasets or tasks, so even parameters at the same channel of the same layer in different pretrained models may indicate different semantic meanings. The misaligned channels cause difficulty in aggregating parameters of different pretrained models. To address the problem, we adopt a channel alignment module that transforms and aligns channels of different pretrained models. We consider parameters $\mathbf{W}_i^l$ of a convolutional layer in any source model M_i with the size $C_{\text{out}} \times C_{\text{in}} \times K \times K$, where C_{out} is the number of output channels, C_{in} indicates input channels, and K is the kernel size of the convolutional layer. We adopt a lightweight convolutional layer $\mathbf{T}_i^l$ with 1×1 kernel of size $C_{\text{out}} \times C_{\text{out}} \times 1 \times 1$ as the channel alignment module. Specifically, the channels in the source convolutional parameters $\mathbf{W}_i^l$ are reorganized by the channel alignment layer to result in the transformed parameters $\widetilde{\mathbf{W}}_i^l$ as follows:

$$\widetilde{\mathbf{W}}_i^l = \mathbf{T}_i^l * \mathbf{W}_i^l, \tag{1}$$

where we also use $\mathbf{T}_i^l$ to denote the parameters of the alignment module. We show an implementation of the channel alignment module for source parameters of convolutional



(a) Adaptive Aggregation (AdaAgg) Layer

(b) Integrating AdaAgg Layers in a Residual Block

Figure 2. (a) Illustration of the Adaptively Aggregation Layer. The target input $\hat{h}^{l-1}$ goes through the gating networks $\mathbf{A}_i^l$ to compute gating values. The source parameters $\mathbf{W}_i^l$ are first aligned by $\mathbf{T}_i^l$ and then aggregated with these gating values to form the target parameters $\widehat{\mathbf{W}}^l$. The input $\hat{h}^{l-1}$ is finally forwarded through the layer parameterized by $\widehat{\mathbf{W}}^l$. (b) We can change the layers of the network backbone into AdaAgg layers to aggregate models in the zoo. Here is an example where the backbone is composed of residual blocks.

layers here. The idea is easy to extend to other kinds of layers, such as fully connected layers, by employing a linear alignment layer. We initialize the channel alignment layer as an identical mapping, which gives the target model a smooth warm-up from the pretrained weights.

Adaptive Aggregation. With channel-aligned source parameters, we develop an adaptive aggregation (**AdaAgg**) layer to dynamically aggregate source model parameters. We have two key insights in the design of the AdaAgg layer: (1) Each data point of each downstream task should have a different aggregation since each data point holds specific relationships with source tasks; (2) The aggregation should be computationally efficient for a large number of source models. We integrate these two key insights into the design of the AdaAgg layer. As shown in Figure 2(a), considering the l-th layer of the network, the AdaAgg layer is equipped with a gating network $\mathbf{A}_i^l$ for each source model M_i, which controls the mixing of its corresponding parameters $\mathbf{W}_i^l$. The gating network $\mathbf{A}_i^l$ takes the feature of the previous layer in the target model $\hat{h}^{l-1}$ as the input and outputs the gating value a_i^l. The aligned source parameters $\widetilde{\mathbf{W}}_i^l$ are aggregated with the gating values to derive the parameters of the target model in this layer $\widehat{\mathbf{W}}^l$ as follows:

$$\widehat{\mathbf{W}}^l = \sum_{i=1}^{m} a_i^l \widetilde{\mathbf{W}}_i^l = \sum_{i=1}^{m} \mathbf{A}_i^l(\hat{h}^{l-1}) \left(\mathbf{T}_i^l * \mathbf{W}_i^l\right), \quad (2)$$

We consider lightweight gating networks to reduce the computation and storage cost of the gating network. For example, for a convolutional layer, the gating network consists of a global average pooling layer, 2 convolutional layers with 1×1 kernel, and a sigmoid activation function. Such design brings little additional computational cost of the gating network compared to processing data with the original convolution operation, even with a large-scale model zoo.

We can easily change the backbone layers of source models into AdaAgg layers to aggregate source models' parameters in each layer. In Figure 2(b), we give an example of the residual block. With the target model parameters, the target data are passed through the target model for training and inference. We can solve the optimization problem of adapting the model zoo to the target task as follows:

$$\min_{\mathbf{\Theta}} \mathbb{E}_{(x,y)\sim\mathcal{D}} \mathcal{L}\left(f^L(\cdot; \widehat{\mathbf{W}}^L) \circ \cdots \circ f^1(x; \widehat{\mathbf{W}}^1), y\right), \quad (3)$$

where $\mathcal{D} = \{(x, y)\}$ is the target dataset, L is the total number of layers, f^l is the operation of the l-th layer parameterized by $\widehat{\mathbf{W}}^l$ and $\mathcal{L}$ is the loss for the target task. $\mathbf{\Theta} = (\mathbf{W}_i, \mathbf{A}_i, \mathbf{T}_i)$ includes source models parameters $\mathbf{W}_i^l$, channel alignment parameters $\mathbf{T}_i^l$, and gating network parameters $\mathbf{A}_i^l$ in all AdaAgg layers. All of these parameters are adaptively trained or tuned to fit for the target task.

3.3. Lite Zoo-Tuning

The adaptive aggregation introduced above is computationally efficient both during training and inference but still requires all the source parameters at the inference stage, as the gating values can be computed only when the target data is presented at inference time. As shown in Figure 3, to further save the storage for applying Zoo-Tuning to devices with limited resource, we relax the dependency of the gating values on each individual target sample to the dependency on the entire dataset, resulting in a *unified gating value* for all data during inference. During training, for a layer l of a source model i, we firstly compute gating values for each

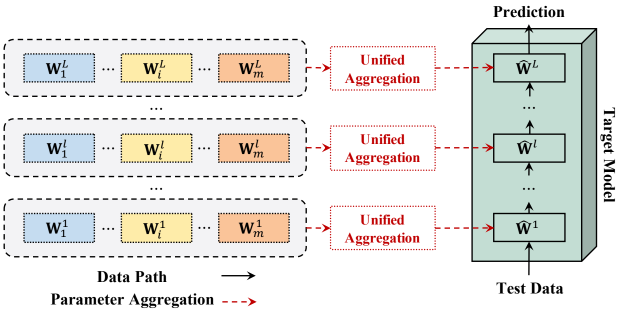


Figure 3. Lite Zoo-Tuning learns a unified adaptive aggregation for all data in the target task based on the temporal ensemble of average gating values of each data batch. Thus, we just need to store the aggregated target model $\widehat{\mathbf{W}}$ for all data during inference.

sample in the batch, denoted as $a_{i,j}^l$, where j is the index of the sample. Then we compute the batch average gating values and their temporal ensemble over the training batches as follows:

$$\bar{a}_i^l = \alpha \cdot \bar{a}_i^l + (1 - \alpha) \left(\frac{1}{b} \cdot \sum_{j=1}^{b} a_{i,j}^l \right). \tag{4}$$

We use the batch average values for all training data in the batch and update the temporal ensemble values with $\alpha = 0.9$, which is commonly used in temporal ensemble techniques. The temporal ensemble values reflect how the target task relies on each source pretrained model and thus serve as the unified gating values for all target data in inference. Now all target data share the same gating values, so we can pre-compute the aggregation of source parameters to form the target model before inference as follows:

$$\widehat{\mathbf{W}}^l = \sum_{i=1}^{m} \bar{a}_i^l \widetilde{\mathbf{W}}_i^l. \tag{5}$$

The key difference between Eqn. (2) and Eqn. (5) is that $\bar{a}_i^l$ in Eqn. (5) is shared by all test data and does not change with each sample. So in inference, the target data can be directly forwarded through the pre-aggregated target model. Thus, the cost of lite Zoo-Tuning model in storage and computation is close to one single model.

3.4. Complexity Analysis

As we propose layer-wise adaptive transfer of source parameters, for simplicity, we only analyze the complexity of one layer, which can be extended to the whole model. We consider aggregating convolutional layers of m pretrained models. Suppose the dimension of the layer is $C_{\text{out}} \times C_{\text{in}} \times K \times K$ where C_{out} and C_{in} are the number of output and input channels, and K is the kernel size. The input feature map has the dimension $C_{\text{in}} \times H \times W$, where H and W are spatial dimensions. W means the width of the feature map only in this Section 3.4 to avoid notation abuse. The original convolution operation has the complexity $O\left(HWK^2C_{\text{out}}C_{\text{in}}\right)$. Zoo-Tuning introduces additional computations for channel alignment, gating values, and adaptive aggregation with the computational complexity of $O\left(mK^2C_{\text{out}}^2C_{\text{in}}\right)$, $O\left(HWC_{\text{in}} + mC_{\text{in}}^2\right)$ and $O\left(mK^2C_{\text{out}}C_{\text{in}}\right)$ respectively, which is $O\left(mK^2C_{\text{out}}^2C_{\text{in}} + HWC_{\text{in}} + mC_{\text{in}}^2\right)$ in total. During inference, as channel alignment is data-independent and can be pre-computed, the cost becomes $O\left(mK^2C_{\text{out}}C_{\text{in}} + HWC_{\text{in}} + mC_{\text{in}}^2\right)$. Compared with the original cost of convolution operation, the additional cost for Zoo-Tuning is small. We will also empirically compare the computational complexity of other transfer learning methods and different variants of Zoo-Tuning.

4. Experiments

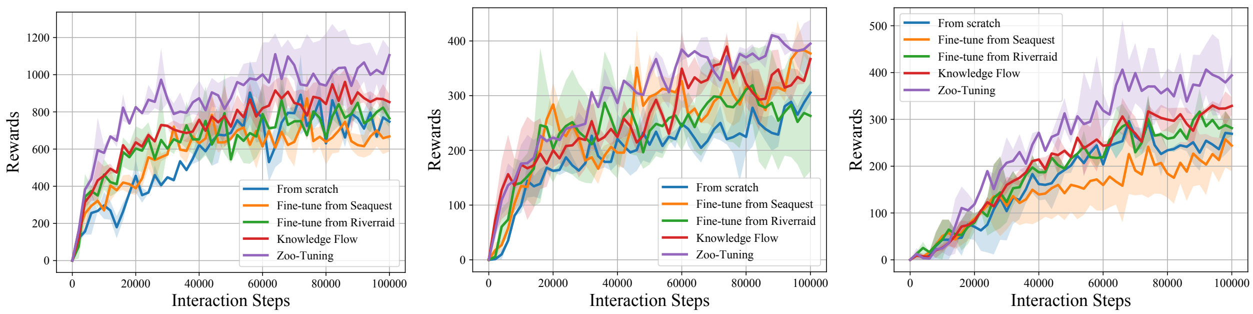
We conduct experiments within three experimental settings. In the first setting, we use a zoo of reinforcement learning models pretrained on various Atari games and transfer to a different set of tasks. In the other two settings, we use a zoo of diverse computer vision models pretrained on various large-scale datasets and transfer to multiple downstream tasks on classification and facial landmark detection. All experiments are implemented in the PyTorch framework.

4.1. Transfer Learning in Reinforcement Learning

To demonstrate the generalizability of the proposed Zoo-Tuning method to various domains, we first conduct experiments on reinforcement learning models.

Benchmarks. Although many reinforcement learning algorithms can achieve much better performance than humans on Atari games, they are still far less efficient than a human learner. So we measure our method at 100k interaction steps (400k environment steps with action repeat of 4) on Atari, which corresponds to the time for a human learner. We use the Seaquest and Riverraid tasks as source tasks and learn an optimal policy by reinforcement learning for each task. We transfer the pretrained reinforcement learning models to 3 downstream tasks: Alien, Gopher, and JamesBond.

Implementation Details. For the learning algorithm, we follow the implementation of Data-Efficient Rainbow (van Hasselt et al., 2019), which modifies hyper-parameters of Rainbow (Hessel et al., 2018) for data efficiency. The model of Data-Efficient Rainbow consists of 2 convolutional layers: 32 filters of size 5×5 with the stride of 5 and 64 filters of size 5×5 with the stride of 5, followed by a flatten layer and 2 noisy linear layers (Fortunato et al., 2018) with the hidden size of 256. We use Adam optimizer(Kingma & Ba, 2015) with a learning rate of 1×10^{-4}. Other hyper-parameters



(a) Alien (b) Gopher (c) JamesBond

Figure 4. Results of transferring pretrained models to downstream tasks in the reinforcement learning of Atari games.

are kept the same as those in van Hasselt et al. (2019). We repeat each experiment 5 times with different seeds and report the mean and variance of the results.

Results. As shown in Figure 4, in all the downstream tasks, Zoo-Tuning outperforms transferring from a single pretrained model and Knowledge Flow (Liu et al., 2019), which also transfers from multiple pretrained models. The results indicate that Zoo-Tuning enables utilizing knowledge from pretrained policies to promote target tasks by adaptive transfer. Simply fine-tuning from each pretrained model performs similarly with training from scratch, and even a little worse, demonstrating that when the source and target tasks have a large gap, brutely transferring knowledge may cause negative transfer.

4.2. Transfer Learning in Image Classification

For the zoo of models in the classification setting, we use 5 ResNet-50 models pretrained on representative computer vision datasets: (1) Supervised pretrained model and (2) Unsupervised pretrained model with MOCO (He et al., 2020) on ImageNet (Russakovsky et al., 2015), (3) Mask R-CNN (He et al., 2017) model for detection and instance segmentation, (4) DeepLabV3 (Chen et al., 2018) model for semantic segmentation, and (5) Keypoint R-CNN model for keypoint detection, pretrained on COCO-2017 challenge datasets of each task. *In total, the zoo of models are trained on millions of images across a wide range of computer vision tasks, which contains abundant knowledge in the computer vision domain.* All pretrained models are found in torchvision (Paszke et al., 2017) or original implementation.

Benchmarks. We divide the 7 downstream tasks into three types of benchmarks: General benchmarks, Fine-grained benchmarks, and Specialized benchmarks, to verify the efficacy of the proposed Zoo-Tuning on different kinds of images: **(1)** *General* benchmarks with **CIFAR-100** (Krizhevsky et al., 2009) and **COCO-70**: **CIFAR-100** contains 100 classes with 600 images per class, which are split into 500 training images and 100 testing images. **COCO-70** is constructed by cropping objects for each image in COCO dataset (Lin et al., 2014) and removing minimal items (with height and width). It contains 70 classes with more than $1,000$ images per category. **(2)** *Fine-grained* benchmarks with **FGVC Aircraft** (Maji et al., 2013), **Stanford Cars** (Krause et al., 2013) and **MIT-Indoors** (Quattoni & Torralba, 2009): **FGVC Aircraft** is a benchmark for the fine-grained aircraft categorization. It has 100 categories containing 100 images each. **Stanford Cars** contains $16,185$ images for 196 classes of cars. The data are split into $8,144$ training images and $8,041$ testing images. **MIT-Indoors** contains 67 Indoor categories, and a total of $15,620$ images. We use a subset of the dataset that has 80 images for training and 20 images for testing per class. **(3)** *Specialized* benchmarks with **DMLab** (Beattie et al., 2016) and **EuroSAT** (Helber et al., 2019): **DMLab** contains frames observed by the agent acting in the DeepMind Lab environment, which are annotated by the distance between the agent and various objects present in the environment. The data are split into $65,550$ training images, $22,628$ validation images and $22,735$ test images. **EuroSAT** dataset is based on Sentinel-2 satellite images covering 13 spectral bands and consisting of 10 classes with $27,000$ labeled and geo-referenced samples.

Implementation Details. We follow the common fine-tuning principle described in (Yosinski et al., 2014) and replace the last task-specific classification layer with a randomly initialized fully connected layer. We adopt SGD with a learning rate of 0.01 and momentum of 0.9 with the same training strategy (total $15k$ iterations for fine-tuning with learning rate decay per $6k$ iterations) for all pretrained models, compared methods and the proposed Zoo-Tuning. This ensures a fair comparison between different methods and avoids over-tuning on specific tasks. We adopt a batch size of 48, and all images are randomly resized and cropped to 224×224 as the input of the network. More details can be found in supplementary materials.

Table 1. Comparison of top-1 accuracy(%) and complexity on the classification benchmarks including General benchmark, Fine-grained benchmark, and Specialized benchmark.

Model	General		Fine-grained			Specialized			Train		Inference	
	Cifar-100	COCO-70	Aircraft	Cars	Indoors	DMLab	EuroSAT	Avg. Acc.	GFLOPs	Params	GFLOPs	Params
ImageNet sup.	81.18	81.97	84.63	89.38	73.69	74.57	98.43	83.41	4.12	23.71M	4.12	23.71M
MOCO pt.	75.31	75.66	83.44	85.38	70.98	75.06	98.82	80.66	4.12	23.71M	4.12	23.71M
MaskRCNN pt.	79.12	81.64	84.76	87.12	73.01	74.73	98.65	82.72	4.12	23.71M	4.12	23.71M
DeepLab pt.	78.76	80.70	84.97	88.03	73.09	74.34	98.54	82.63	4.12	23.71M	4.12	23.71M
Keypoint pt.	76.38	76.53	84.43	86.52	71.35	74.58	98.34	81.16	4.12	23.71M	4.12	23.71M
ensemble	82.26	82.81	**87.02**	**91.06**	73.46	**76.01**	98.88	84.50	20.60	118.55M	20.60	118.55M
distill	82.32	82.44	85.00	89.47	73.97	74.57	98.95	83.82	24.72	142.28M	4.12	23.71M
Knowledge Flow	81.56	81.91	85.27	89.22	73.37	75.55	97.99	83.55	28.83	169.11M	4.12	23.71M
Lite Zoo-Tuning	83.39	83.50	85.51	89.73	75.12	75.22	**99.12**	84.51	4.53	130.43M	4.12	23.71M
Zoo-Tuning	**83.77**	**84.91**	86.54	90.76	**75.39**	75.64	**99.12**	**85.16**	4.53	130.43M	4.18	122.54M

Results. We report the top-1 accuracy on the test data of each task and the complexity of each method. For our method, we report Zoo-Tuning and lite Zoo-Tuning. For the single-model transfer method, we compare with fine-tuning from every single pretrained model. For methods using all pretrained models, we compare with three methods: using the ensemble of fine-tuned source models for prediction, distilling from the ensemble, and Knowledge Flow (Liu et al., 2019), which is designed to transfer from multiple models. From Table 1. We have the following observations:

On all the three benchmarks, Zoo-Tuning consistently outperforms fine-tuning from each single pretrained model, which indicates that Zoo-Tuning successfully aggregates and utilizes the rich knowledge in the whole zoo of models.

Compared with the methods using all pretrained models, Zoo-Tuning shows higher or comparable performance on most of the tasks. Compared with the parameters in the model zoo, the additional parameters in Zoo-Tuning is about 10%, which shows that the adaptive modules are lightweight. With the adaptive parameter aggregation mechanism, Zoo-Tuning is more computationally efficient. Note that the ensemble predictions require fine-tuning all the candidate pretrained models on the target task firstly. Even at inference time, each query sample should go through all the fine-tuned models to get the final prediction, causing high inference cost. Distilling and knowledge flow show similar inference costs as Zoo-Tuning, but Zoo-Tuning achieves higher performance on almost all the tasks. The results demonstrate that Zoo-Tuning is a both effective and efficient solution to transfer learning from a zoo of models.

Lite Zoo-Tuning also outperforms compared methods on average accuracy. We specially compare it with distilling from the ensemble (Distill) since they are both efficient in inference. Although the performance gain is not large, lite Zoo-Tuning still outperforms Distill consistently on all tasks and achieves greater advantages in the training cost. This is because Distill still needs to forward data through all source models, while lite Zoo-Tuning only needs to pass the data through the aggregated model. Furthermore, Distill needs to fine-tune all the pretrained models on the target data first and then distill a target model from the ensemble outputs of fine-tuned models, which requires a high training cost linearly increasing with the number of source models. The results match the goal of lite Zoo-Tuning to substantially reduce the storage cost in inference while keeping relatively high performance, which is more scalable when training with a large number of source models.

Zoo-Tuning achieves higher accuracy than lite Zoo-Tuning, which demonstrates that capturing fine-grained data-dependent gating values would help to adapt the pretrained models to the target task but with more cost of storage and computation in inference. Lite Zoo-Tuning costs the same GFLOPs and parameters as the single model in inference, with slight performance drop than Zoo-Tuning, which serves as a trade-off between performance and efficiency.

4.3. Transfer Learning in Facial Landmark Detection

Benchmarks. To explore the usage of Zoo-Tuning on more diverse and complex downstream vision tasks, we use the same model zoo as the image classification tasks in Section 4.2 and consider transferring to three facial landmark detection tasks, **300W** (Sagonas et al., 2013), **WFLW** (Wu et al., 2018), and **COFW** (Burgos-Artizzu et al., 2013). The **300W** is a combination of HELEN (Le et al., 2012), LFPW (Belhumeur et al., 2013), AFW (Zhu & Ramanan, 2012), XM2VTS and IBUG datasets, where each face has 68 landmarks. We follow (Ren et al., 2016) and use the

3148 training images. We evaluate the performance using the full set containing 689 images. The **WFLW** dataset is a dataset built on the WIDER Face (Yang et al., 2016). There are 7500 training and 2500 testing images with 98 manual annotated landmarks. The **COFW** dataset consists of 1345 training and 507 testing faces with 29 facial landmarks.

Table 2. Comparison of NME results on facial landmark detection tasks: 300W, WFLW, and COFW.

MODEL	300W	WFLW	COFW
SCRATCH	3.66	5.33	4.20
IMAGENET SUP.	3.52	4.90	3.66
MOCO PT.	3.45	4.75	3.63
MASKRCNN PT.	3.53	4.87	3.67
DEEPLAB PT.	3.53	4.89	3.73
KEYPOINT PT.	3.50	4.90	3.66
ENSEMBLE	**3.33**	4.64	**3.46**
DISTILL	3.45	4.74	3.53
KNOWLEDGE FLOW	3.71	5.28	4.58
ZOO-TUNING	3.41	**4.58**	3.51

Implementation Details. We generally follow the protocol in Sun et al. (2019). We follow the standard training scheme in (Wu et al., 2018). All the faces are cropped by the provided boxes according to the center location and resized to 256×256. We augment the data by ± 30 degrees in-plane rotation, $0.75 - 1.25$ scaling, and randomly flipping. The models are trained for 60 epochs with a batch size of 16. We use Adam optimizer (Kingma & Ba, 2015). The base learning rate is 1×10^{-4} and is decayed by a rate of 0.1 at the 30-th and 50-th epochs. In testing, each keypoint location is predicted by transforming the highest heat value location to the original image space and adjusting it with a quarter offset in the direction from the highest response to the second highest response (Chen et al., 2017).

Results. We use the inter-ocular distance as normalization and report the normalized mean error (NME) for evaluation in Table 2. Comparing fine-tuning from each single pretrained model, we can observe that the MOCO (He et al., 2020) pretrained model generally outperforms other pretrained models when transferring to the facial landmark detection tasks. The results confirm that even commonly-used models such as the ImageNet pretrained model cannot dominate all downstream tasks, and it is important to select the more suitable pretrained models for the target task. Zoo-Tuning addresses the challenge by gating networks trained on the target task and consistently outperforms transferring from each single model. Knowledge Flow achieves little improvement than training from scratch and even performs worse on 300W and COFW. This method uses pretrained models as teachers to guide the learning of the student network. But this kind of guidance cannot fully utilize and adapt the knowledge in the zoo, especially when the target tasks are not close to pretrained tasks. Zoo-Tuning adapts the whole model zoo to the target, which is a more effective way of knowledge transfer. Zoo-Tuning achieves comparable performance with the ensemble, but with *much less computational cost* during training and inference.

4.4. Analysis

Variants of Zoo-Tuning. We compare Zoo-Tuning with its variants to demonstrate the efficacy of different modules in Zoo-Tuning. We use all the five pretrained computer vision models described above and transfer them to the COCO-70 dataset and the WFLW dataset. From Table 3, we have the following observations: (1) Zoo-Tuning outperforms Zoo-Tuning w/o $\mathbf{T}$, which demonstrates that channel alignment of source parameters improves the performance for adaptive aggregation. (2) Zoo-Tuning w/o $\mathbf{T}$ and Zoo-Tuning outperform average aggregation with a large margin. Average aggregation aggregates all the source parameters with the same weight, which treats all source parameters equally. The results demonstrate that it is essential to learn the gating values on the target task and adaptively fit the aggregation to better serve the target task.

Table 3. Ablation study on variants of Zoo-Tuning.

METHOD	COCO	WFLW
AVERAGE AGGREGATION	80.53	4.75
ZOO-TUNING W/O $\mathbf{T}$	83.92	4.64
ZOO-TUNING	**84.91**	**4.58**

Visualization of Gating Values. We visualize the gating values in each layer of each source model learned by Zoo-Tuning on the CIFAR-100 dataset. As shown in Figure 5, different source models have diverse gating values, which indicates that different source pretrained models have different relationships to the target task, and Zoo-Tuning learns to aggregate the source models for the target task adaptively. Different layers show different gating values, which matches the study on deep networks that different layers contain different knowledge and exhibit different transferability. The gating values show some insights on the transferability of the pretrained models. Overall, ImageNet supervised learning model generally has the highest values because ImageNet is more closely related to the CIFAR-100. This could also be verified by the results of fine-tuning from each single pretrained model in Table 1. Besides, the advantage of the ImageNet supervised model is mainly on top layers, and other networks also have high values in the bottom and intermediate layers. This matches the previous observation that knowledge in deep networks goes from general to task-specific as the layer goes deeper.

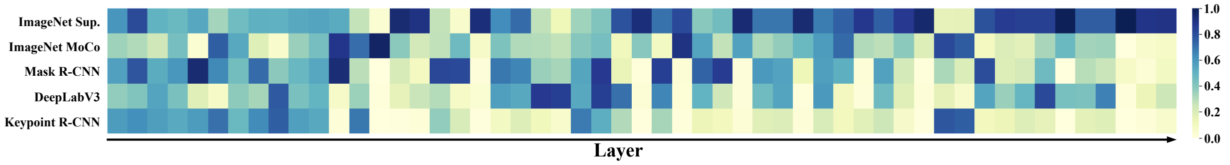


Figure 5. Visualization of gating values in each layer of source models using the five vision pretrained models as source models and using CIFAR-100 as the target task. From the left to right is from the bottom layer to the top layer. The darker color means a higher gating value.

Number of Parameters. Zoo-Tuning from a zoo of models outperforms the best single model in the zoo with large margins. However, there is a concern that this improvement may be due to the increase in model parameters. To evaluate how the number of parameters influences the final result, we conduct an ablation study where we change the 5 diverse pretrained models to 5 ImageNet supervised pretrained models and perform Zoo-Tuning from the 5 same models (denoted as 5× ImageNet). We report the results on the image classification datasets COCO-70 and Aircraft.

Table 4. Ablation study on parameters of the network.

Method	Aircraft	COCO	Param
ImageNet	84.63	81.97	1×
5× ImageNet	84.92	82.13	5×
Lite Zoo-Tuning	85.51	83.50	1×
Zoo-Tuning	**86.54**	**84.91**	5×

As shown in Table 4, although the model of transferring from 5× ImageNet has the same parameters as Zoo-Tuning from 5 diverse pretrained models, it achieves minor improvements compared with fine-tuning from one single model. Note that the proposed lite Zoo-Tuning also outperforms fine-tuning from a single ImageNet model, while lite Zoo-Tuning has the same number of parameters with a single model. The results indicate that the key to the superior performance of Zoo-Tuning is not simply increasing model parameters but adaptively aggregating rich knowledge from the source models. The results not only show the effectiveness of the proposed Zoo-Tuning, but also show the importance of knowledge aggregation in the problem of transfer learning from a zoo of diverse models.

Number of Pretrained Models. We study how the number of pretrained models in the zoo affects the proposed method. We experiment on the CIFAR-100 dataset by sequentially adding the MoCo, Keypoint R-CNN, DeepLabV3, Mask R-CNN, and ImageNet supervised models into the model zoo. We report the results of Zoo-Tuning with different numbers of pretrained models as well as the results of the single best models in the corresponding zoo in Table 5.

Table 5. Ablation study on the number of models.

# of models	2	3	4	5
Single Best	76.38	78.76	79.12	81.18
Zoo-Tuning	78.62	79.78	81.50	**83.77**

From the results, with different model zoo sizes of 2, 3, 4, and 5 pretrained models, Zoo-Tuning consistently outperforms fine-tuning from the best single model in the zoo, respectively. Note that it is even more difficult to quickly select out the best pretrained model in practice. This indicates that Zoo-Tuning can effectively and efficiently utilize the knowledge in the whole model zoo to promote transfer learning performance, which makes it a better choice in real-world applications. Also, we can find that the performance of Zoo-Tuning increases with the increasing number of pretrained models, which demonstrates that Zoo-Tuning can hold a varying number of pretrained models and is extendable to more source models.

5. Conclusion

In this paper, we propose Zoo-Tuning to enable transfer learning from a zoo of models. We align the channels of source parameters with a channel alignment layer and adopt a gating network depending on the input data for each source model to aggregate their parameters, which derives the target model. The channel alignment layer and the gating network are trained, and the source pretrained parameters are tuned by the target task to adaptively transfer knowledge from the zoo of source models to the target task. We further propose lite Zoo-Tuning with the temporal ensemble of batch average gating values, which further reduces the storage cost at inference time. Experiment results in reinforcement learning, image classification, and facial landmark detection demonstrate that Zoo-Tuning achieves state-of-the-art performance with small computational and storage costs.

Acknowledgments

This work was kindly supported by National Key R&D Program of China (2020AAA0109201), NSFC grants (62022050, 62021002, 61772299), Beijing Nova Program (Z201100006820041), and MOE Innovation Plan of China.

References

Agrawal, P., Girshick, R., and Malik, J. Analyzing the performance of multilayer neural networks for object recognition. In *European conference on computer vision*, pp. 329–344. Springer, 2014.

Bao, Y., Li, Y., Huang, S.-L., Zhang, L., Zheng, L., Zamir, A., and Guibas, L. An information-theoretic approach to transferability in task transfer learning. In *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 2309–2313. IEEE, 2019.

Beattie, C., Leibo, J. Z., Teplyashin, D., Ward, T., Wainwright, M., Küttler, H., Lefrancq, A., Green, S., Valdés, V., Sadik, A., et al. Deepmind lab. *arXiv preprint arXiv:1612.03801*, 2016.

Belhumeur, P. N., Jacobs, D. W., Kriegman, D. J., and Kumar, N. Localizing parts of faces using a consensus of exemplars. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2930–2940, 2013.

Burgos-Artizzu, X. P., Perona, P., and Dollár, P. Robust face landmark estimation under occlusion. In *Proceedings of the IEEE international conference on computer vision*, pp. 1513–1520, 2013.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Chen, X., Wang, S., Fu, B., Long, M., and Wang, J. Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning. In *Advances in neural information processing systems*, 2019.

Chen, Y., Shen, C., Wei, X.-S., Liu, L., and Yang, J. Adversarial posenet: A structure-aware convolutional network for human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1212–1221, 2017.

Cho, K. and Bengio, Y. Exponentially increasing the capacity-to-computation ratio for conditional computation in deep learning. *arXiv preprint arXiv:1406.7362*, 2014.

Davis, A. and Arel, I. Low-rank approximations for conditional feedforward computation in deep neural networks. *arXiv preprint arXiv:1312.4461*, 2013.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pp. 647–655. PMLR, 2014.

Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., and Brox, T. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in neural information processing systems*, 2014.

Eigen, D., Ranzato, M., and Sutskever, I. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*, 2013.

Fortunato, M., Azar, M. G., Piot, B., Menick, J., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., et al. Noisy networks for exploration. *International Conference on Learning Representations*, 2018.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. *Deep learning*, volume 1. MIT press Cambridge, 2016.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

He, K., Girshick, R., and Dollár, P. Rethinking imagenet pretraining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4918–4927, 2019.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Helber, P., Bischke, B., Dengel, A., and Borth, D. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., and Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Kornblith, S., Shlens, J., and Le, Q. V. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2661–2671, 2019.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d Object Representations for Fine-Grained Categorization. In *ICCV Workshop*, 2013.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Le, V., Brandt, J., Lin, Z., Bourdev, L., and Huang, T. S. Interactive facial feature localization. In *European conference on computer vision*, pp. 679–692. Springer, 2012.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436–444, 2015.

Li, X., Xiong, H., Wang, H., Rao, Y., Liu, L., Chen, Z., and Huan, J. Delta: Deep learning transfer using feature map with attention for convolutional networks. In *International Conference on Learning Representations*, 2019.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *ECCV*, 2014.

Liu, I.-J., Peng, J., and Schwing, A. G. Knowledge flow: Improve upon your teachers. In *International Conference on Learning Representations*, 2019.

Maji, S., Rahtu, E., Kannala, J., Blaschko, M., and Vedaldi, A. Fine-grained visual classification of aircraft. *Technical report*, 2013.

Masoudnia, S. and Ebrahimpour, R. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42(2): 275–293, 2014.

Nguyen, C., Hassner, T., Seeger, M., and Archambeau, C. Leep: A new measure to evaluate transferability of learned representations. In *International Conference on Machine Learning*, pp. 7294–7305. PMLR, 2020.

Oquab, M., Bottou, L., Laptev, I., and Sivic, J. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1717–1724, 2014.

Pan, S. J. and Yang, Q. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10): 1345–1359, 2009.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. 2017.

Quattoni, A. and Torralba, A. Recognizing indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 413–420. IEEE, 2009.

Raghu, M., Zhang, C., Kleinberg, J., and Bengio, S. Transfusion: Understanding transfer learning for medical imaging. In *Advances in neural information processing systems*, 2019.

Ramalho, T., Sousbie, T., and Peluchetti, S. An empirical study of pretrained representations for few-shot classification. *arXiv preprint arXiv:1910.01319*, 2019.

Ren, S., Cao, X., Wei, Y., and Sun, J. Face alignment via regressing local binary features. *IEEE Transactions on Image Processing*, 25(3):1233–1245, 2016.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., and Pantic, M. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 397–403, 2013.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., Mu, Y., Wang, X., Liu, W., and Wang, J. High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*, 2019.

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., and Liu, C. A survey on deep transfer learning. In *International conference on artificial neural networks*, pp. 270–279. Springer, 2018.

Tran, A. T., Nguyen, C. V., and Hassner, T. Transferability and hardness of supervised classification tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1395–1405, 2019.

van Hasselt, H., Hessel, M., and Aslanides, J. When to use parametric models in reinforcement learning? *In Advances in Neural Information Processing Systems*, 2019.

Wu, W., Qian, C., Yang, S., Wang, Q., Cai, Y., and Zhou, Q. Look at boundary: A boundary-aware face alignment algorithm. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2129–2138, 2018.

Xuhong, L., Grandvalet, Y., and Davoine, F. Explicit inductive bias for transfer learning with convolutional networks. In *International Conference on Machine Learning*, pp. 2825–2834. PMLR, 2018.

Yang, S., Luo, P., Loy, C.-C., and Tang, X. Wider face: A face detection benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5525–5533, 2016.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, 2014.

Zhai, X., Puigcerver, J., Kolesnikov, A., Ruyssen, P., Riquelme, C., Lucic, M., Djolonga, J., Pinto, A. S., Neumann, M., Dosovitskiy, A., et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019.

Zhu, X. and Ramanan, D. Face detection, pose estimation, and landmark localization in the wild. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 2879–2886. IEEE, 2012.