Customizing Student Networks From Heterogeneous Teachers
via Adaptive Knowledge Amalgamation

Chengchao Shen[1,*], Mengqi Xue[1,*], Xinchao Wang[2], Jie Song[1,3], Li Sun[1], Mingli Song[1,3]

[1]Zhejiang University, [2]Stevens Institute of Technology,

[3]Alibaba-Zhejiang University Joint Institute of Frontier Technologies

{chengchaoshen,mqxue,sjie,lsun,brooksong}@zju.edu.cn,xinchao.w@gmail.com

Abstract

A massive number of well-trained deep networks have been released by developers online. These networks may focus on different tasks and in many cases are optimized for different datasets. In this paper, we study how to exploit such heterogeneous pre-trained networks, known as teachers, so as to train a customized student network that tackles a set of selective tasks defined by the user. We assume no human annotations are available, and each teacher may be either single- or multi-task. To this end, we introduce a dual-step strategy that first extracts the task-specific knowledge from the heterogeneous teachers sharing the same sub-task, and then amalgamates the extracted knowledge to build the student network. To facilitate the training, we employ a selective learning scheme where, for each unlabelled sample, the student learns adaptively from only the teacher with the least prediction ambiguity. We evaluate the proposed approach on several datasets and experimental results demonstrate that the student, learned by such adaptive knowledge amalgamation, achieves performances even better than those of the teachers.

1. Introduction

Deep networks have been applied to almost all computer vision tasks and have achieved state-of-the-art performances, such as image classification [17, 29, 6, 11], semantic segmentation [21, 2, 1] and object detection [25, 19, 37]. This tremendous success is in part attributed to the large amount of human annotations utilized to train the parameters of the deep networks. In many cases, however, such training annotations are unavailable to the public due to for example privacy reasons. To reduce the re-training effort and enable the plug-and-play reproduction, many researchers have therefore shared online their pre-trained networks, which focus on different tasks or datasets.
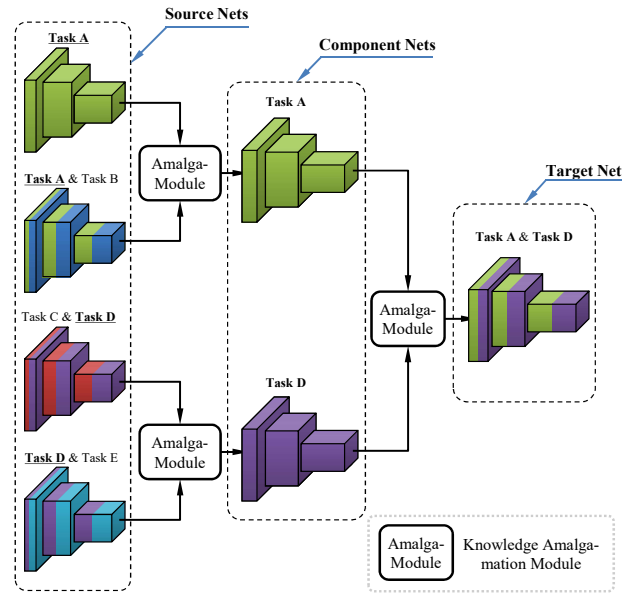
---
*Equal contribution

Figure 1. The dual-stage knowledge amalgamation strategy for customizing student networks. Given four source nets working on heterogeneous tasks, each of which may be either single- or multi-task, we cluster them into two groups, one for Task A and the other for Task D. We then conduct the first-round knowledge amalgamation for each group to derive the two components nets, based on which the second-round amalgamation is further carried out to produce the final student model specified by the user.

In this paper, we investigate how to utilize such pre-trained networks that focus on different tasks, which we term as *heterogeneous teachers*, to learn a customized and multitalented network, termed as the *student*. We assume that we are given a pool of well-trained teachers, yet have no access to any human annotation; each teacher can be either single- or multi-task, and may or may not overlap in tasks. Our goal is to train a compact and versatile student network that tackles a set of selective tasks defined by the user, via learning from the heterogeneous teachers. In other words, the student is expected to *amalgamate* the multidisciplinary

knowledge scattered among the heterogeneous teachers into its compact-sized model, so that it is able to perform the user-specified tasks.

The merit of this customized-knowledge amalgamation problem, therefore, lies in that it allows for reusing pre-trained deep networks to build a tailored student model on user's demand, again without having to access human annotations. To this end, we introduce a dual-stage strategy that conducts knowledge amalgamation twice. In this first stage, from the pool we pick heterogeneous teachers covering one or multiple desired tasks, which we term as *source nets*; we then cluster the source nets sharing the same task into groups, from each of which we learn a single-task network, termed as *component net*. In the second stage, we construct our final student network, termed as *target net*, by amalgamating the heterogeneous knowledge from the learned component nets.

We show an example in Fig. 1 to illustrate our problem setup and the overall workflow. Here we are given a pool of four source nets, of which one is single-task and the others are multi-task. We aim to train a compact target net, without human-labelled annotations, which in this case handles simultaneously Tasks A and D demanded by the user. In the first stage, we cluster the four source nets into two groups, one on Task A and the other on Task D, and learn a component net for each task; in the second stage, we amalgamate the two component nets to build the user-specified multi-task target net.

This dual-stage approach for knowledge amalgamation, as will be demonstrated in our experiments, turns out to be more effective than the one-shot approach that learns a multi-task target net directly from the heterogeneous source nets. Furthermore, it delivers the component nets as byproducts, which serve as the modular units that can be further integrated to produce any combined-task target nets, significantly enhancing the flexibility and modularity of the knowledge amalgamation.

As we assume no human-labelled ground truths are provided, it is crucial to decide which teacher among the multiple to use, so as to train the student effectively in both stages. In this regard, we exploit a selective learning scheme, where we feed unlabelled samples to the multiple teacher candidates and allow the student to, for each sample, learn adaptively only from the teacher with the least prediction ambiguity. Specifically, we adopt the chosen teacher's feature maps and score vectors as supervisions to train the student, where the feature learning is achieved via a dedicated *transfer bridge* that aligns the features from the teacher and the student. Please note that, for each sample, we conduct the teacher selection and update which teacher to learn from.

In short, our contribution is a novel knowledge amalgamation strategy that customizes a multitalented student network from a pool of single- or multi-task teachers handling different tasks, without accessing human annotations. This is achieved via a dual-step approach, where the modular and single-task component networks are derived in the first step followed by their being amalgamated in the second. Specifically, for each unlabelled sample, we utilize a selective learning strategy to decide which teacher to imitate adaptively, and introduce a dedicated transfer bridge for feature learning. Experimental results on several datasets demonstrate that the learned student models, despite their compact sizes, consistently outperform the teachers in their specializations.

2. Related Work

Knowledge distillation [8] adopts a teacher-guiding-student strategy where a small student network learns to imitate the output of a large teacher network. In this way, the large teacher network can transfer knowledge to the student network with smaller model size, which is widely applied to model compression. Following [8], some works are proposed to exploit the intermediate representation to optimize the learning of student network, such as Fit-Net [26], DK^{2}PNet [32], AT [36] and NST [13]. In summary, these works pay more attention on knowledge transfer among the same classification task. Transfer learning is proposed to transfer knowledge from source domain to target domain to save data on target domain [24]. It contains two main research directions: cross-domain transfer learning [22, 12, 10, 4] and cross-task one [9, 3, 5, 35]. In the case of cross-domain transfer learning, the dataset adopted by source domain and the counterpart of target domain are different in domain but the same in category. Also, cross-task transfer learning adopts the datasets that have the same domain but different categories. Transfer learning mainly focuses on compensating for the deficit of data on target domain with enough data on source domain. By contrast, our approach amalgamates multiple pre-trained models to obtain a multitalented model using unlabelled data.

To exploit knowledge of massive trained deep-learning-based models, researchers have made some promising attempts. MTZ [7] merges multiple correlated trained models by sharing neurons among these models for cross-model compression. Knowledge flow [14] transfers knowledge from multiple teacher models to student one with strategy that student learns to predict with the help of teachers, but gradually reduce the dependency on teachers, finally predict independently. Despite very promising solutions, the above approaches still depend on labelled dataset, which is not suitable for our application scenario where no human labels are available.

The approach of [28] proposes to transfer knowledge from multiple trained models into a single one in a layer-wise manner with unlabelled dataset. It adopts an auto-
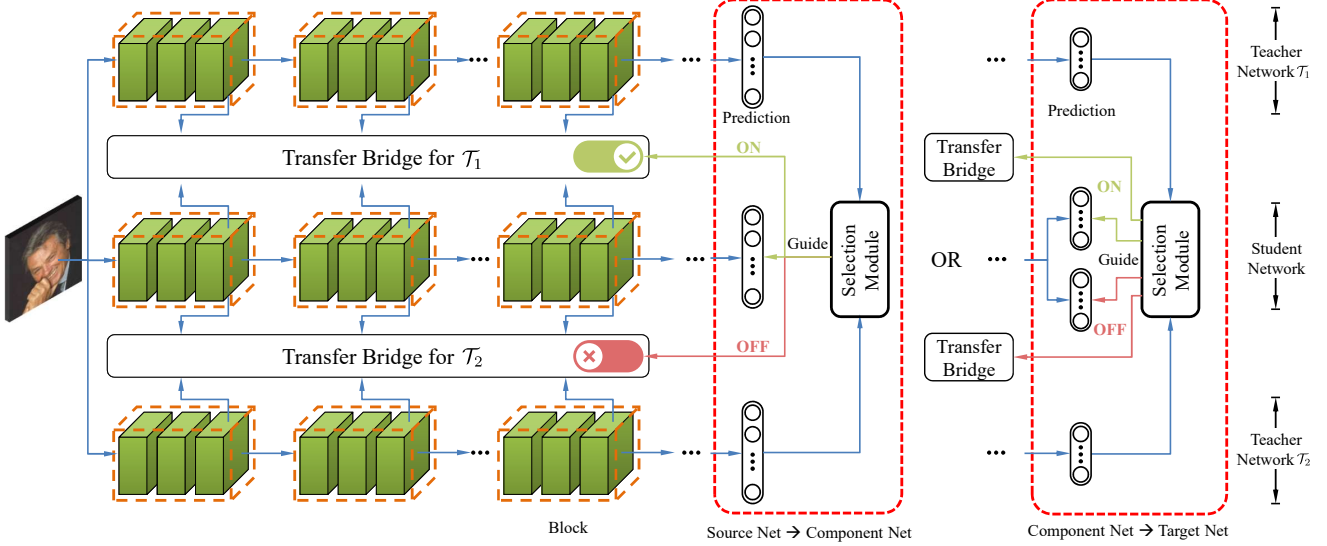
Figure 2. Amalgamating knowledge from multiple teachers. The student learns both the predictions and the features from a teacher model, chosen among multiple via a selective learning module. The features of this selected teacher network are then transferred to the student network via the transfer bridge in a block-wise manner. The two amalgamation steps, i.e., source-to-component and component-to-target, undergo the same process.

encoder architecture to amalgamate features from multiple single-task teachers. Several knowledge amalgamation methods are also proposed to handle the above task [33, 23, 34]. The proposed approach here, on the other hand, handles teachers working on both single or multiple tasks, and follows a dual-stage strategy tailored for customizing the student network that also gives rise to component nets as byproducts.

3. The Proposed Approach

In this section, we give more details on the proposed approach for customizing multi-task students. We first give an overview of the overall process, then introduce the transfer bridge for learning the features of the student, afterwards we describe the selective learning scheme for choosing teachers adaptively, and finally show the loss function.

3.1. Overview

Our problem setup is as follows. We assume that we are given a pool of pre-trained source nets, each of which may be single- or multi-task, where the former can be treated as a degenerated case of the latter. These source nets may be trained for distinct tasks and optimized for different datasets. Let K_i denote the set of tasks handled by source net i, and let $\mathcal{K} = \bigcup_i K_i$ denote the set of tasks covered by all the teachers. Our goal is to customize a student model that tackles a set of user-specified tasks, denoted by $K_s \subseteq \mathcal{K}$. Also, we use M_s to denote the number of tasks to be trained for the student, i.e., $|K_s| = M_s$. As the initial attempt along this line, for now we focus on image classification and assume the source nets all take the form of the widely adopted resnet [6]. The proposed approach, however, is not restricted to resnet and is applicable to other architectures as well.

To this end, we adopt a dual-stage strategy to conduct the selective knowledge amalgamation. In the first stage, we pick all the source nets that cover one or multiple tasks specified by the users, i.e., $i : K_i \cap K_s \neq \varnothing$, and then cluster them into M_s groups, each of which focus on one task only. For each such group we carry out the first-round knowledge amalgamation and derive a component net tailored for each task, all of which together are further amalgamated again in the second round to form the final multi-task target network.

The two rounds of knowledge amalgamation are achieved in a similar manner, as depicted in Fig. 2. In the first round, we refer to the source and the component respectively as teachers and students, and in the second, we refer to the component and the target respectively as teachers and student. Specifically, we conduct a block-wise learning scheme, as also done in [30, 6, 11], where a transfer bridge is established between each teacher and the student so as to allow the student to imitate the features of the teacher. In both amalgamation rounds, for each unlabelled sample, student adaptively learns from only one selected teacher, which is taken to be the one that yields the least prediction ambiguity. For In what follows, we introduce the proposed transfer bridge and the selective learning strategy in details.

3.2. Transfer Bridge

A transfer bridge, as shown in Fig. 3, is set up between the student and each teacher, in aim to align the features of the student and the teachers so that the former can learn
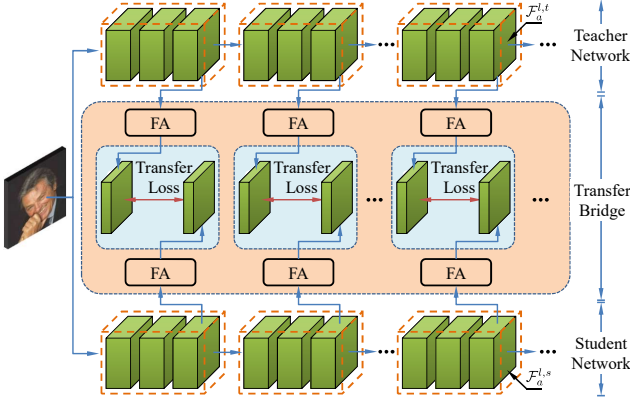
Figure 3. Transfer bridge between a teacher network and a student. The block-wise features from teacher and student are respectively transformed into $\mathcal{F}_a^t$ and $\mathcal{F}_a^s$ by the FA module, which are then utilized for computing the transfer loss.

from the latter. As the teachers may be multi-task and therefore comprise knowledge not at the interest of the student, we would have to "filter" and transform the related features from the teacher in a way that is learnable by the student. This is achieved via a dedicated feature alignment (FA) module and a regularized loss function, discussed as follows.

Feature Alignment (FA). An FA module, which learns to filter and align the target-task-related features, is introduced between each block of the teacher and the student. In our implementation, FA takes the form of an 1×1 convolutional operation [28, 30, 18]. As depicted in Fig. 4, the feature maps of both the student and the teacher are weighted and summed to obtain a new feature map across channels by the 1×1 convolutional operation. We write,

$$\mathcal{F}_{a,c} = \sum_{c'=1}^{C_{in}} w_{c,c'} \mathcal{F}_{c'}, \tag{1}$$

where $\mathcal{F}_{a,c}$ denotes the c-th channel of aligned feature maps $\mathcal{F}_a$, $\mathcal{F}_{c'}$ denotes the c'-th channel of input feature maps from the teacher or the student, and $w_{c,c'}$ denotes the weight of 1×1 convolutional operation, which transforms $\mathcal{F}_{c'}$ to $\mathcal{F}_{a,c}$.

Transfer Loss and Weight Regularization. To supervise the feature learning, we define a transfer loss based on the aligned features of the teacher and the student. Let $\mathcal{F}_a^{l,t}$ denote the feature maps from block l of the teacher network and let $\mathcal{F}_a^{l,s}$ denote those of the student. We first introduce the vanilla transfer loss, as follows,

$$\mathcal{L}_a^{l,t} = \frac{1}{C_{out}^l H^l W^l} \| \mathcal{F}_a^{l,s} - \mathcal{F}_a^{l,t} \|^2, \tag{2}$$

where C_{out}^l, H^l and W^l denotes the channel, height and width size of $\mathcal{F}_a^{l,t}$ or $\mathcal{F}_a^{l,s}$, respectively.
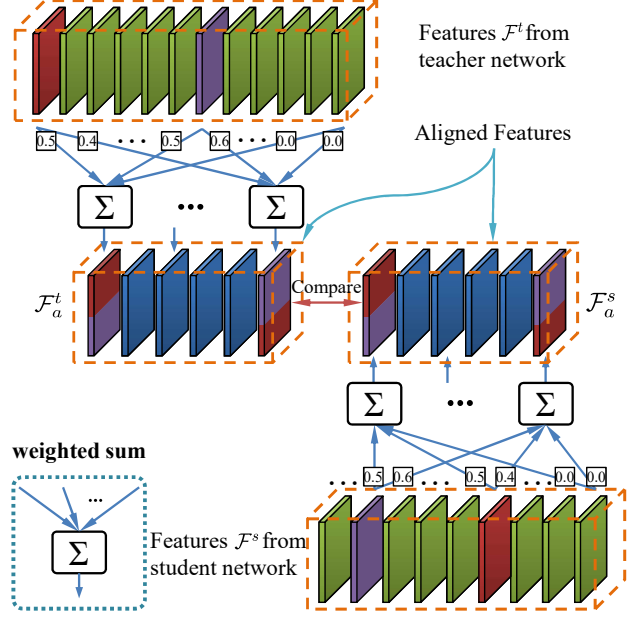

Figure 4. Feature alignment. The features from teacher network and student network are transformed and aligned using 1×1 convolutional operation.

This vanilla transfer loss alone, however, may lead to trivial solutions: by taking the two features maps to be zero, the loss collapses to zero. To avoid such degenerated case, we impose a regularization on the transfer loss. As the aligned features are controlled by the learnable parameters w_{ij}, we introduce a constraint of w_{ij}, as follows,

$$\sum_{i=1}^{C_{in}^l} w_{ij}^{(l,t)2} = 1, \tag{3}$$

which on the one hand limits the magnitude of w_{ij} to a reasonable range, and on the other hand eliminate the trivial solutions. For the sake of optimization, we then relax the above hard constraint into a soft one:

$$\mathcal{L}_{reg}^{l,t} = \frac{1}{C_{out}^l} \sum_{j=1}^{C_{out}^l} (\sum_{i=1}^{C_{in}^l} w_{ij}^{(l,t)2} - 1)^2, \tag{4}$$

which are further added to the final loss described in Sec. 3.4.

3.3. Selective Learning

As we assume no ground-truth annotations are given for training the student and meanwhile multiple teachers handling the same task might be available, we would have to ensure that, for each unlabelled sample, we allow the student to learn from only the "best" possible teacher among many. Since there are, again, no ground truths for evaluating the teacher with the best sample-wise performance, we

resort to learning from the teacher with the most "confident" prediction. In other words, the student imitates the predictions and features of the teacher with the least prediction ambiguity.

Here, we use entropy impurity to measure the prediction ambiguity: the smaller the value is, the higher the confidence of prediction is. The teacher with minimal entropy impurity is therefore selected to guide the learning of student network:

$$\mathrm{I}(p^t(x)) = -\sum_i p_i^t(x) \log(p_i^t(x)), \qquad (5)$$

$$t_{se} = \underset{t}{\arg\min}\, \mathrm{I}(p^t(x)), \qquad (6)$$

where t_{se} indexes the selected teacher.

3.4. Loss Function

To imitate the predictions of teachers, we introduce a soft target loss between the predictions of teacher networks and that of the student. Since the student is required to learn multiple teachers and the outputs of teachers are typically different from each other, a learnable scale parameter λ_t is introduced to compensate such scale difference. We write,

$$\mathcal{L}_{soft}^t = \frac{1}{C_{cls}} \| \mathcal{F}_{score}^s - \lambda_t \mathcal{F}_{score}^t \|^2, \qquad (7)$$

where $\mathcal{F}_{score}^s$ and $\mathcal{F}_{score}^t$ denote the logits before softmax layer from student and teacher, respectively, and C_{cls} denotes the length of logits.

The total loss of knowledge amalgamation between source nets and component net and the one between component nets and target net are defined as follows,

$$\mathcal{L}_{total} = \sum_{l=1}^{L-1} \{ \mathcal{L}_a^{(l,t_{se})} + \mathcal{L}_{reg}^{(l,t_{se})} \} + \mathcal{L}_{soft}^{t_{se}}, \qquad (8)$$

where L denotes the number of blocks in source, component or target net.

4. Experiments

In this section, we show the experimental results of the proposed adaptive knowledge amalgamation. We start by introducing the datasets we used and the implementation details, and then provide the quantitative analysis including results on attribute and category classification. More results and additional details can be found in the supplementary material.

4.1. Experiment Settings

4.1.1 Datasets

CelebFaces Attributes Dataset (CelebA) [20] is a large-scale face attributes dataset, which consists of more than

Dataset	Partition	
	2 parts	4 parts
Stanford Dogs	$\mathcal{D}_1, \mathcal{D}_2$	$\mathcal{D}_1', \mathcal{D}_2', \mathcal{D}_3', \mathcal{D}_4'$
CUB-200-2011	$\mathcal{B}_1, \mathcal{B}_2$	$\mathcal{B}_1', \mathcal{B}_2', \mathcal{B}_3', \mathcal{B}_4'$
FGVC-Aircaft	$\mathcal{A}_1, \mathcal{A}_2$	$\mathcal{A}_1', \mathcal{A}_2', \mathcal{A}_3', \mathcal{A}_4'$
Cars	$\mathcal{C}_1, \mathcal{C}_2$	$\mathcal{C}_1', \mathcal{C}_2', \mathcal{C}_3', \mathcal{C}_4'$

Table 1. The partition of four fine-grained datasets for the training of source nets. Each set contains the same number of categories.

200K celebrity images, each with 40 attribute annotations. It contains $162,770$ images for training, $19,868$ images for validation and $19,962$ ones for testing. Due to it's large size and massive attribute annotations, it can be used to build a well-trained source network pool to verify the proposed approach. We randomly split the training set into six parts with the same size, in which five parts are used to train five different multi-task teachers and the remaining one is used as unlabelled training data for the student. The experiments of network customization are conducted on two attribute groups: mouth-related attributes and hair-related attributes. More experiments on other attribute groups can be found in the supplementary material.

Besides experiments on attribute recognition, four fine-grained datasets are used to evaluate the effectiveness on network customization of category recognition. Stanford Dogs [15] contains $12,000$ images about 120 different kinds of dogs. FGVC-Aircraft [27] consists of $10,000$ images of 100 aircraft variants. CUB-200-2011 [31] is a bird dataset, which includes $11,788$ images from 200 bird species. Cars [16] comprises $16,185$ images of 196 classes of cars. The four datasets can be categorized into two groups: animal-related and vehicle-related dataset. As shown in Tab. 1, all datasets are randomly split into several sets, each of which contains the same number of categories. For example, both $\mathcal{D}_1$ and $\mathcal{D}_2$ contain 60 breeds of dogs, $\mathcal{D}_1'$ to $\mathcal{D}_4'$ contain 30 breeds of dogs, respectively. The details of each set can be found in the supplementary material.

4.1.2 Implementation

The proposed method is implemented by PyTorch on a Quadro M6000 GPU. The source nets adopt the same network architecture: resnet-18 [6], which are trained by fine-tuning the ImageNet pretrained model. Both component net and target net adopt resnet-18-like network architectures. The adopted net has the same net structure as the original resnet-18, except the channel number of feature maps. For example, the target net amalgamates knowledge from multiple component nets, so the target net should be more "knowledgeable" than a single component net, which should have more channels than component net. More implementation details can be found in the supplementary ma-

Source Net	Attributes	Source Net	Attributes
$\mathcal{S}_1^{\text{mouth}}$	big lips, narrow eyes, pale skin	$\mathcal{S}_6^{\text{mouth}}$	mouth slightly open
$\mathcal{S}_2^{\text{mouth}}$	big lips, chubby, young	$\mathcal{S}_7^{\text{mouth}}$	mouth slightly open, chubby blurry, blond hair
$\mathcal{S}_3^{\text{mouth}}$	smiling, arched eyebrows, attractive, black hair	$\mathcal{S}_8^{\text{mouth}}$	wearing lipstick, arched eyebrows, attractive
$\mathcal{S}_4^{\text{mouth}}$	smiling, bags under eyes, blurry, blond hair	$\mathcal{S}_9^{\text{mouth}}$	wearing lipstick, bags under eyes, blurry
$\mathcal{S}_5^{\text{mouth}}$	smiling, bushy eyebrows, oval face, brown hair	$\mathcal{S}_{10}^{\text{mouth}}$	wearing lipstick, bushy eyebrows, oval face

Table 2. Source nets that work on multiple attribute recognition tasks on the CelebA dataset.

Model	Mouth-Related Attributes			
	Big Lips	Smiling	Mouth Slightly Open	Wearing Lipstick
Source Net	$\mathcal{S}_1^{\text{mouth}}(68.7)$, $\mathcal{S}_2^{\text{mouth}}(68.5)$	$\mathcal{S}_3^{\text{mouth}}(88.6)$, $\mathcal{S}_4^{\text{mouth}}(88.6)$, $\mathcal{S}_5^{\text{mouth}}(87.5)$	$\mathcal{S}_6^{\text{mouth}}(89.6)$, $\mathcal{S}_7^{\text{mouth}}(89.5)$	$\mathcal{S}_8^{\text{mouth}}(90.4)$, $\mathcal{S}_9^{\text{mouth}}(90.4)$, $\mathcal{S}_{10}^{\text{mouth}}(90.3)$
Component Net	$69.2^{\uparrow 0.5,0.7}$	$90.5^{\uparrow 1.9,1.9,3.0}$	$91.4^{\uparrow 1.8,1.9}$	$91.7^{\uparrow 1.3,1.3,1.4}$
Target Net	$69.2^{\uparrow 0.5,0.7}$	$90.8^{\uparrow 2.2,2.2,3.3}$	$91.4^{\uparrow 1.8,1.9}$	$91.8^{\uparrow 1.4,1.4,1.5}$

Model	Hair-Related Attributes			
	Black Hair	Blond Hair	Brown Hair	Bangs
Source Net	$\mathcal{S}_1^{\text{hair}}(85.2)$, $\mathcal{S}_2^{\text{hair}}(86.9)$	$\mathcal{S}_3^{\text{hair}}(94.0)$, $\mathcal{S}_4^{\text{hair}}(94.2)$	$\mathcal{S}_5^{\text{hair}}(86.4)$, $\mathcal{S}_6^{\text{hair}}(86.3)$, $\mathcal{S}_7^{\text{hair}}(86.7)$	$\mathcal{S}_8^{\text{hair}}(94.5)$, $\mathcal{S}_9^{\text{hair}}(94.4)$
Component Net	$87.8^{\uparrow 2.6,0.9}$	$95.0^{\uparrow 1.0,0.8}$	$88.0^{\uparrow 1.6,1.7,1.3}$	$95.2^{\uparrow 0.7,0.8}$
Target Net	$87.9^{\uparrow 2.7,1.0}$	$95.0^{\uparrow 1.0,0.8}$	$88.1^{\uparrow 1.7,1.8,1.4}$	$95.2^{\uparrow 0.7,0.8}$

$\uparrow$ denotes performance improvement compared with the corresponding source net.

Table 3. The performance (%) of knowledge amalgamation from source nets to component net and from component nets to target net on the CelebA dataset. Number in parentheses denotes the accuracy of the corresponding source net. Unlike the component net handles only one task, the target net handles four tasks simultaneously.

terial.

4.2. Experimental Results

In what follows, we show network customization results for attribute- and category-classification, learning from various numbers of teachers, ablation studies by turning off some of the modules, as well as the results of one-shot amalgamation.

4.2.1 Network Customization for Attribute

In the first amalgamation step, multiple related source nets are amalgamated into a single component net to obtain a component task. Tab. 2 collects 10 source nets, each of which contains a mouth-related attribute recognition task. For example, $\mathcal{S}_1^{\text{mouth}}$ is a source net for multiple tasks: "big lips", "narrow eyes" and "pale skin", including a mouth-related attribute task: "big lips". Combined with $\mathcal{S}_2^{\text{mouth}}$ that also works on "big lips" task, they are amalgamated into a component net for "big lips" task, as shown in Tab. 3. In the second amalgamation step, multiple component nets specified by user are amalgamated into the target net. In Tab. 3, the component nets about mouth-related attributes:

"big lips", "smiling", "mouth slightly open", and "wearing lipstick" are used to customize the corresponding target net.

From Tab. 3, we observe consistent experimental results on two attribute groups[*] as follows. On the one hand, the performance of component net is superior to those of the corresponding source nets. Also, the obtained component nets are more compact than the ensemble of all source nets, as shown in Tab. 4. In particular, for "smiling" attribute, the component net outperforms the source net $\mathcal{S}_5^{\text{mouth}}$ by 3.0%. It supports that our approach is indeed able to transfer knowledge from multiple source nets into the component net, and the transferred knowledge can significantly supplement the knowledge deficiency of a single source net. On the other hand, the target net achieves comparable or better performance on the corresponding tasks, yet is more resource-efficient. The net parameters and computation load (FLOPs: Float Operations) of target net, as shown in Tab. 4, are much lower than the summation of all component nets,

To validate the flexibility of network customization, we also customize target net with different numbers of compo-

---

[*]The lookup table for the hair-related source nets as Tab. 2 is provided in the supplementary material.

Model	Parameters	FLOPs
Source Nets	111.8M	36.3G
Component Nets	44.8M	14.5G
Target Net	22.1M	7.0G

Table 4. The comparison of resource required in 10 source nets, 4 component nets and target net in Tab. 3, including the number of parameters and FLOPs.

Model	**Mouth-Related Attributes**		
	smiling lipstick	smiling mouth open lipstick	big lips smiling mouth open lipstick
Target Net	91.1 91.9	91.2 91.7 91.7	69.2 90.8 91.4 91.8
Model	**Hair-Related Attributes**		
	black hair brown hair	black hair brown hair bangs	black hair blond hair brown hair bangs
Target Net	87.8 88.2	87.7 88.1 95.2	87.9 95.0 88.1 95.2

Table 5. The performance (%) of the customization of target net with different numbers of component nets on the CelebA dataset.

nent nets, for which the results are shown in Tab. 5. These results demonstrate that our proposed approach can be competent to the customization for different numbers of component nets.

4.2.2 Network Customization for Category

We also conduct network customization experiments on category recognition. As shown in Tab. 6, source nets on four datasets are provided. For example, source net for part of Stanford Dogs $\mathcal{D}_1$: $\mathcal{S}_1^{\text{dog}}$ is trained on the category sets $\mathcal{D}_1$ and $\mathcal{B}_1'$. The source nets for Stanford Dogs include $\mathcal{S}_1^{\text{dog}}$ and $\mathcal{S}_2^{\text{dog}}$ for $\mathcal{D}_1$, $\mathcal{S}_3^{\text{dog}}$ and $\mathcal{S}_4^{\text{dog}}$ for $\mathcal{D}_2$. To customize a target net for category set $\mathcal{D}_1 \cup \mathcal{D}_2$, the dual-step amalgamation is implemented as follows. In the first step, source nets $\mathcal{S}_1^{\text{dog}}$ and $\mathcal{S}_2^{\text{dog}}$ are amalgamated into a component net for $\mathcal{D}_1$. In the same way, source nets $\mathcal{S}_3^{\text{dog}}$ and $\mathcal{S}_4^{\text{dog}}$ are amalgamated into a component net for $\mathcal{D}_2$. In the second step, component nets for $\mathcal{D}_1$ and $\mathcal{D}_2$ are amalgamated into the final target net. Experiments on the remaining datasets are implemented in the same way.

The experimental results shown in Tab. 7 demonstrate

Dataset	**Source Nets**							
	$\mathcal{S}_1$		$\mathcal{S}_2$		$\mathcal{S}_3$		$\mathcal{S}_4$	
Dogs	$\mathcal{D}_1$	$\mathcal{B}_1'$	$\mathcal{D}_1$	$\mathcal{B}_2'$	$\mathcal{D}_2$	$\mathcal{B}_3'$	$\mathcal{D}_2$	$\mathcal{B}_4'$
CUB	$\mathcal{B}_1$	$\mathcal{D}_1'$	$\mathcal{B}_1$	$\mathcal{D}_2'$	$\mathcal{B}_2$	$\mathcal{D}_3'$	$\mathcal{B}_2$	$\mathcal{D}_4'$
Aircraft	$\mathcal{A}_1$	$\mathcal{C}_1'$	$\mathcal{A}_1$	$\mathcal{C}_2'$	$\mathcal{A}_2$	$\mathcal{C}_3'$	$\mathcal{A}_2$	$\mathcal{C}_4'$
Cars	$\mathcal{C}_1$	$\mathcal{A}_1'$	$\mathcal{C}_1$	$\mathcal{A}_2'$	$\mathcal{C}_2$	$\mathcal{A}_3'$	$\mathcal{C}_2$	$\mathcal{A}_4'$

Table 6. The source nets for network customization of category recognition on four fine-grained datasets, whose name is abbreviated as "Dogs", "CUB", "Aircraft" and "Cars", respectively.
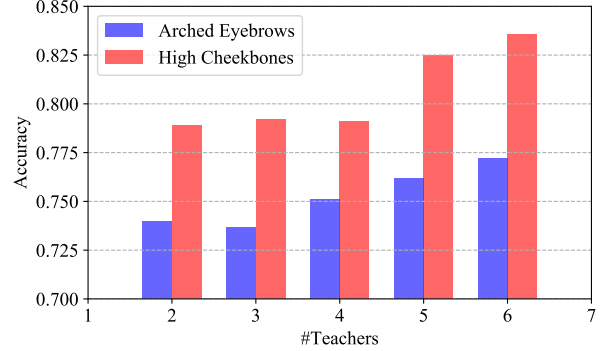

Figure 5. The performance of knowledge amalgamation for different number of source nets on the CelebA dataset.

that the component nets consistently outperform the corresponding source nets, and the target net achieves comparable or better accuracy than the corresponding component net. It supports that our proposed method also works on category recognition task.

4.2.3 Learning from Varying Numbers of Teachers

To investigate the effect of knowledge amalgamation for more teachers, we also conduct experiments in which varying numbers of source nets are amalgamated into a single component net. The experiments are implemented on two face attribute recognition tasks, "arched eyebrows" and "high cheekbones", as shown in Fig. 5. With more teachers, the performance tends to be better for both face attribute recognition tasks. By integrating more teachers, the student network may potentially "absorb" more complementary knowledge from multiple teachers and significantly reduce erroneous guidances from teachers.

4.2.4 Ablation Study

Ablation study is conducted on several attributes to investigate the effectiveness of the modules adopted in our proposed approach. Specifically, we verify the effectiveness of each module by comparing the whole model to the model without the corresponding module. Additional compared method is knowledge distillation, which does not contain

Model	Category Sets			
	Stanford Dogs $\mathcal{D}_1$	Stanford Dogs $\mathcal{D}_2$	FGVC-Aircraft $\mathcal{A}_1$	FGVC-Aircraft $\mathcal{A}_2$
Source Net	$\mathcal{S}_1^{dog}(87.4), \mathcal{S}_2^{dog}(87.3)$	$\mathcal{S}_3^{dog}(87.9), \mathcal{S}_4^{dog}(87.7)$	$\mathcal{S}_1^{air}(70.1), \mathcal{S}_2^{air}(71.3)$	$\mathcal{S}_3^{air}(65.4), \mathcal{S}_4^{air}(65.2)$
Component Net	$88.2^{\uparrow 0.8, 0.9}$	$88.5^{\uparrow 0.6, 0.8}$	$71.5^{\uparrow 1.4, 0.2}$	$66.4^{\uparrow 1.0, 1.2}$
Target Net	$88.4^{\uparrow 1.0, 1.1}$	$88.6^{\uparrow 0.7, 0.9}$	$71.8^{\uparrow 1.7, 0.5}$	$66.8^{\uparrow 1.4, 1.6}$

Model	Category Sets			
	CUB-200-2011 $\mathcal{B}_1$	CUB-200-2011 $\mathcal{B}_2$	Cars $\mathcal{C}_1$	Cars $\mathcal{C}_2$
Source Net	$\mathcal{S}_1^{bird}(74.5), \mathcal{S}_2^{bird}(74.8)$	$\mathcal{S}_3^{bird}(73.9), \mathcal{S}_4^{bird}(74.0)$	$\mathcal{S}_1^{car}(69.5), \mathcal{S}_2^{car}(71.1)$	$\mathcal{S}_3^{car}(71.2), \mathcal{S}_4^{car}(71.3)$
Component Net	$75.4^{\uparrow 0.9, 0.6}$	$74.8^{\uparrow 0.9, 0.8}$	$72.1^{\uparrow 2.6, 1.0}$	$72.8^{\uparrow 1.6, 1.5}$
Target Net	$75.8^{\uparrow 1.3, 1.0}$	$75.4^{\uparrow 1.5, 1.4}$	$72.5^{\uparrow 3.0, 1.4}$	$73.1^{\uparrow 1.9, 1.8}$

$\uparrow$ denotes performance improvement compared with the corresponding source network.

Table 7. The performance (%) of knowledge amalgamation from source nets to component net and from component net to target net on four fine-grained datasets.

Module	Attributes		
	black hair	mouth slightly open	brown hair
KD [8]	87.1	90.2	87.4
wo/TB	87.4	91.3	87.7
wo/TS	87.4	91.0	87.8
whole model	**87.8**	**91.4**	**88.0**

Table 8. The performance (%) for ablation study on the CelebA dataset. *KD* denotes knowledge distillation (baseline). *TB* denotes transfer bridge. *TS* denotes teacher-selective learning.

Method	Attributes	
	Black Hair	Blond Hair
one-shot amalgamation	85.6	86.1
two-stage amalgamation	**87.6**	**95.1**

Table 9. The performance (%) comparison between one-shot amalgamation and two-stage amalgamation on the CelebA dataset.

the optimization of student net and leads to poorer performance.

transfer bridge module and teacher selective learning strategy.

The results shown in Tab. 8 demonstrate that both transfer bridge and selective learning strategy significantly improve the performance of the model. The transfer bridges deliver the partial task-demanded intermediate features of teacher networks to the student network, which provide more supervision to the student network compared to knowledge distillation. And the selective learning strategy takes the most confident teacher as the learning target, which can significantly reduce the misleading information provided by teachers.

4.2.5 One-shot Amalgamation

To further explore network customization methods, we compare an intuitive variant of our proposed dual-stage method: one-shot amalgamation. In this scenario, multiple sources nets are directly amalgamated into target net without the component net as the intermediate byproduct. The experiments are conducted on two face attribute recognition tasks, as shown in Tab. 9. The results demonstrate that two-stage amalgamation method outperforms the one-shot one on both of face attributes. Because one-shot amalgamation is required to simultaneously learn knowledge from more source networks, instead of learning from few component nets adopted in two-stage method, it potentially complicates

5. Conclusion and Future Work

In this paper, we propose an adaptive knowledge amalgamation method to learn a user-customized student network, without accessing human annotations, from a pool of single- or multi-task teachers working on distinct tasks. This is achieved specifically via a dedicated dual-stage approach. In the first stage, source nets covering the same task are clustered into groups, from each of which a component net is learned; in the second, the components are further amalgamated into the user-specified target net. Both stages undergo a similar knowledge amalgamation process, where for each unlabelled sample, the student learns the features and predictions of only one teacher, taken to be the one with the least prediction ambiguity. The feature learning is achieved via a dedicated transfer bridge, in which the features of the student are aligned with those of the selected teacher for learning. We conduct experiments on several datasets and show that, the learned student that comes in a compact size, yields consistent superior results to those of the teachers in their own specializations. For future work, we plan to customize networks using teachers of different network architectures.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(12):2481–2495, 2017.

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(4):834–848, 2018.

[3] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4109–4118, 2018.

[4] Zhengming Ding, Sheng Li, Ming Shao, and Yun Fu. Graph adaptive knowledge transfer for unsupervised domain adaptation. In *European Conference on Computer Vision (ECCV)*, pages 37–52, 2018.

[5] Behnam Gholami, Ognjen Rudovic, and Vladimir Pavlovic. Punda: Probabilistic unsupervised domain adaptation for knowledge transfer across visual categories. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 3601–3610, 2017.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[7] Xiaoxi He, Zimu Zhou, and Lothar Thiele. Multi-task zipping via layer-wise neuron sharing. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6019–6029, 2018.

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[9] Seunghoon Hong, Junhyuk Oh, Honglak Lee, and Bohyung Han. Learning transferrable knowledge for semantic segmentation with deep convolutional neural network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3204–3212, 2016.

[10] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. Deep transfer metric learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 325–333, 2015.

[11] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[12] Haoshuo Huang, Qixing Huang, and Philipp Krähenbühl. Domain transfer through deep activation matching. In *European Conference on Computer Vision (ECCV)*, pages 611–626, 2018.

[13] Zehao Huang and Naiyan Wang. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219*, 2017.

[14] Alexander Schwing Iou-Jen Liu, Jian Peng. Knowledge flow: Improve upon your teachers. In *International Conference on Learning Representations (ICLR)*, 2019.

[15] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *Workshop on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[16] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *International IEEE Workshop on 3D Representation and Recognition (3dRR)*, 2013.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[18] Tianhong Li, Jianguo Li, Zhuang Liu, and Changshui Zhang. Few sample knowledge distillation for efficient network compression. *arXiv preprint arXiv:1812.01839*, 2018.

[19] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, pages 21–37. Springer, 2016.

[20] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *The IEEE International Conference on Computer Vision (ICCV)*, 2015.

[21] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[22] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and S Yu Philip. Transfer feature learning with joint distribution adaptation. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 2200–2207, 2013.

[23] Sihui Luo, Xinchao Wang, Gongfan Fang, Yao Hu, Dapeng Tao, and Mingli Song. Knowledge amalgamation from heterogeneous networks by common feature learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

[24] Sinno Jialin Pan, Qiang Yang, et al. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 22(10):1345–1359, 2010.

[25] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 91–99, 2015.

[26] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations (ICLR)*, 2014.

[27] E. Rahtu M. Blaschko A. Vedaldi S. Maji, J. Kannala. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[28] Chengchao Shen, Xinchao Wang, Jie Song, Li Sun, and Mingli Song. Amalgamating knowledge towards comprehensive classification. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 3068–3075, 2019.

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[31] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[32] Zhenyang Wang, Zhidong Deng, and Shiyao Wang. Accelerating convolutional neural networks with dominant convolutional kernel and knowledge pre-regression. In *European Conference on Computer Vision (ECCV)*, pages 533–548, 2016.

[33] Jingwen Ye, Yixin Ji, Xinchao Wang, Kairi Ou, Dapeng Tao, and Mingli Song. Student becoming the master: Knowledge amalgamation for joint scene parsing, depth estimation, and more. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[34] Jingwen Ye, Xinchao Wang, Yixin Ji, Kairi Ou, and Mingli Song. Amalgamating filtered knowledge: Learning task-customized student from multi-task teachers. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

[35] Shan You, Chang Xu, Chao Xu, and Dacheng Tao. Learning from multiple teacher networks. In *The ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1285–1294, 2017.

[36] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations (ICLR)*, 2017.

[37] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. Single-shot refinement neural network for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4203–4212, 2018.