Progressive Adversarial Networks for Fine-Grained Domain Adaptation

Sinan Wang, Xinyang Chen, Yunbo Wang, Mingsheng Long (✉), and Jianmin Wang
School of Software, BNRist, Tsinghua University, China
Research Center for Big Data, Tsinghua University, China
Beijing Key Laboratory for Industrial Big Data System and Application
{wang-sn17, chenxiny17, wangyb15}@mails.tsinghua.edu.cn, {mingsheng, jimwang}@tsinghua.edu.cn

Abstract

Fine-grained visual categorization has long been considered as an important problem, however, its real application is still restricted, since precisely annotating a large fine-grained image dataset is a laborious task and requires expert-level human knowledge. A solution to this problem is applying domain adaptation approaches to fine-grained scenarios, where the key idea is to discover the commonality between existing fine-grained image datasets and massive unlabeled data in the wild. The main technical bottleneck lies in that the large inter-domain variation will deteriorate the subtle boundaries of small inter-class variation during domain alignment. This paper presents the Progressive Adversarial Networks (PAN) to align fine-grained categories across domains with a curriculum-based adversarial learning framework. In particular, throughout the learning process, domain adaptation is carried out through all multi-grained features, progressively exploiting the label hierarchy from coarse to fine. The progressive learning is applied upon both category classification and domain alignment, boosting both the discriminability and the transferability of the fine-grained features. Our method is evaluated on three benchmarks, two of which are proposed by us, and it outperforms the state-of-the-art domain adaptation methods.

1. Introduction

Fine-grained recognition aims to categorize an object among a large number of subordinate categories within the same root category. It is a valuable problem in the sense that it could potentially endow machine learning models with strong cognitive abilities approaching human experts on some tasks. For example, we might be interested in distinguishing subordinate species of birds such as pacific gull or black-tailed gull. In recent years, there has been great advance in some fundamental problems of fine-grained recognition. On one hand, the ability of deep networks for identifying subtle differences between highly similar objects has been greatly improved [23, 43, 45, 16, 46, 25, 13]. On the other hand, an increasing number of fine-grained image datasets have been collected, including a variety of root categories such as birds [55, 54, 50], dogs [18, 27], flowers [36, 1], aircrafts [52, 32], cars [44, 22, 26, 58], and food [4].

Still, it is unrealistic to cover all subordinate categories, and the limited size of the existing datasets still hampers the scalability of the fine-grained recognition algorithms. Annotating large-scale image datasets with fine-grained labels is time-consuming and requires strong expertise, especially for some particular application domains. To solve this problem, a promising idea is to apply the domain adaptation approaches [38] to fine-grained recognition tasks. For example, learning to categorize birds in the field guide may help recognize bird species in the wild, as illustrated in Figure 1. Thus, we may transfer the common knowledge from existing labeled datasets to massive unlabeled data, and save the efforts of dense fine-grained annotations.

However, there are new challenges in the context of fine-grained domain adaptation: the concurrence of large
inter-domain variations, small inter-class variations, and large intra-class variations. The classic domain adaptation algorithms overcome the inter-domain variations by making images from different domains have similar distributions in the feature space [40, 28, 47, 12]. When it comes to the fine-grained domain adaptation, the situation becomes more complicated in that we have to confront tough issues brought by the fine-grained categorization. A combination of large intra-class variations and small inter-class variations may deteriorate the inter-class boundaries, and thus make the classic domain adaptation algorithms fail in respectively mapping objects of neighboring categories from the source domain to the target domain. As in Figure 1, yellow-headed blackbirds and bobolinks are perceptually similar and may be mismatched across domains.

This paper aims to address these challenges by designing a new fine-grained domain adaptation method, and presents Progressive Adversarial Networks (PAN). In fine-grained scenarios, natural objects have taxonomic ranks in biology, and man-made objects also have reasonable hierarchical labels. The general idea is integrating curriculum learning [3] and adversarial learning [15] to enable domain adaptation progressively from coarse-grained categories (easy) to fine-grained categories (difficult). This disentangles the difficulties by large inter-domain variations, small inter-class variations, and large intra-class variations. The training process of our model only depends on hierarchical category labels on the source domain. We evaluate our method on three benchmarks, two of which are proposed by us, based on several existing datasets for fine-grained visual categorization and one brand-new dataset we collect from the web and filter manually. We demonstrate that the proposed approach outperforms the state-of-the-art domain adaption methods.

2. Related Work

2.1. Fine-Grained Visual Categorization

In recent years, fine-grained visual categorization has become a prevalent problem in computer vision. As it requires expertise to recognize the subtle differences between the subordinate categories within the same root category, some methods introduced additional labels such as part-annotations and visual attributes to enhance fine-grained recognition [5, 59, 60, 53, 14].

Instead of using the cost-prohibitive part-annotations or additional attributes, some work attempted to improve the fine-grained recognition performance in other ways. Krause et al. [20] tried to solve the fine-grained recognition problem by generating parts using co-segmentation and alignment. Lin et al. [25] proposed a two-stream CNN model based on the bilinear pooling, which is also trained with category labels. Gao et al. [13] presented the compact bilinear pooling method as an extension of [25] to lower the computation complexity while retaining comparable accuracy. Other variants of the original bilinear pooling method were soon proposed and applied on the neural network models for fine-grained recognition [24, 19]. Dubey et al. introduced confusion in the activations [10] and revisited Maximum-Entropy [11]. To further alleviate the difficulty of collecting expert-level annotations manually, some methods were proposed to make the fine-grained recognition models benefit from the large-scale but noisy web data [21, 14, 57].

The above methods achieved fairly good performance even without part-annotations, but yet, their scalability is limited by the lack of fine-grained annotations for the vast subordinate categories in the real world.

2.2. Domain Adaptation

Domain adaptation is to transfer knowledge from the source domain to the target domain, which saves the cost of manual annotations [38]. The discrepancy between the source domain and the target domain causes the main difficulties for knowledge transfer. To learn domain-invariant, transferable features, some work proposed different adaptation layers based on deep networks [49, 28, 30]. Some more recent work studied the domain-adversarial methods, which incorporated the adversarial learning [15] into the domain adaptation framework [47, 12]. These models aligned the feature distributions of different domains by trying to fool the domain discriminator. Through further conditioning the adversarial adaptation models on the discriminative information in class predictions, CDAN [29] sheds light into the direction in addressing the problem of fine-grained cross-domain recognition. PFAN [7] adopts an “Easy-to-Hard Transfer Strategy” to select easy samples from the target domain and align these pseudo-labeled samples with their corresponding source categories. A basic distinction from our work is that PFAN learns from progressive samples without exploring the granularity information, while our method learns from progressive granularity diametrically.
These methods above are insightful. Unfortunately, all of them were not specifically designed for fine-grained cross-domain visual categorization and did not explore the label hierarchy specific in fine-grained recognition scenarios.

2.3. Fine-Grained Domain Adaptation

Fine-grained domain adaptation was first studied by Gebru et al. [14]. They proposed a model that was trained with annotated web images and evaluated with real-world data, using the domain adaptation approach proposed in [47] and requiring additional annotations of attributes. Only when labeled images on the target domain are available, can its semi-supervised adaptive loss be performed, which is only a tailored design for fine-grained domain adaptation.

With a unique design of exploiting strong supervision, Xu et al. [57] utilized detailed annotations including object bounding boxes and part landmarks, in addition to standard image-level labels. As much knowledge as possible was transferred from existing strongly supervised datasets to weakly supervised web images.

Additionally, Cui et al. [9] achieved obvious improvements on several fine-grained visual benchmarks, by fine-tuning well-performing CNNs pre-trained on the large-scale iNaturalist2017 dataset [51]. They proposed a measure to estimate domain similarity and selected a subset from the source domain that is more similar to target domain.

All above methods obtained encouraging performance. However, the problem setups are completely different from ours, as in Table 1. Our approach does not require attributes, bounding boxes or part landmarks but relies on the hierarchical labels that are easier to obtain in fine-grained tasks. To our knowledge, our work is the first method designed for unsupervised fine-grained domain adaptation only depending on hierarchical image-level labels from source domain.

3. Method

In the fine-grained domain adaptation problem, we are given a source domain $S = \{(x, y_f, y_c^k)\} \forall_{n_s}$ examples with both fine-grained label $y_f$ and coarse-grained labels $\{y_c^k\}_{k=1}^{K}$ in a $K$-layer class hierarchy, and a target domain $T = \{(x, ?, ?)\} \forall_{n_t}$ unlabeled examples. There is a large discrepancy between the joint distributions $P(x, y)$ and $Q(x, y)$ of source domain and target domain respectively. Due to the distribution shift, a fine-grained recognition model trained on $S$ cannot perform accurately on $T$.

The domain adversarial networks [12] are performant domain adaptation models. They usually consist of three network modules: the feature extractor $F$, the domain discriminator $D$ and the label predictor $Y$. A combination of $F$ and $Y$ is trained with recognition objectives (only with labels from the source domain). Simultaneously, to extract domain transferable features, $F$ and $D$ work together and play an adversarial game. The domain discriminator $D$ is trained to distinguish the source domain from the target domain, while the feature extractor $F$ is trained to confuse $D$, keeping it away from making correct judgments.

The Progressive Adversarial Network (PAN) exploits hierarchical labels of fine-grained objects. As opposed to fine-grained labels in the bottom layer of the label hierarchy, we refer to higher-level labels in the label hierarchy as coarse-grained labels. Fine-grained domain adaptation is very difficult due to its large inter-domain variations, small inter-class variations, and large intra-class variations. In contrast, coarse-grained domain adaptation is easy. Inspired by curriculum learning [3], we begin with the easy granularity, and then progressively move to the difficult granularity. An accurate sup-class alignment across domains works as a solid foundation for sub-class alignment.

3.1. Progressive Granularity Learning

In progressive granularity learning (PGL), we progressively change the granularity of supervision for the recognition task on the source domain from coarse-grained to fine-grained during training. We replace fine-grained ground-truth labels with dynamic mixing of fine-grained ground-truth labels and coarse-grained predicted distributions given by the recognition model trained at the corresponding granularity, denoted as progressive labels, as shown in Figure 2. Predicted distributions convey information of relationships between classes, which are considered beneficial to domain adaptation [47]. The coarse-grained labels could be more than one levels, say $K$ ($K \geq 1$). The coarse-grained CNN, an auxiliary network effective at training and will be removed at inference, is introduced with a feature extractor $G$ and $K$ label predictors $C^k, k = 1, \ldots, K$. The data point $x$ with coarse-grained labels $y_c^k, k = 1, \ldots, K$ is fed into the coarse-grained CNN. The coarse-grained CNN is trained on the source domain by minimizing the recognition objective:

$$\sum_{k=1}^{K} L_y (\hat{y}_c^k, y_c^k),$$

where $\hat{y}_c^k = C^k (G (x))$ is the $k$-th coarse-grained predicted distribution and $L_y$ is the cross-entropy (CE) loss.

The fine-grained labels of all images are explored by the fine-grained CNN, which is trained by minimizing a novel coarse-fine hybrid loss we propose in this paper:

$$L_b (\hat{y}_c^k | \{y_c^k\}_{k=1}^{K}, \hat{y}_f, y_f) = D_{KL} \left( \varepsilon y_f + (1 - \varepsilon) \sum_{k=1}^{K} \frac{y_c^k}{K} \| \hat{y}_f \right),$$

where $D_{KL}$ is the Kullback-Leibler divergence, and $\hat{y}_f = Y (F (x_i))$ is the fine-grained predicted distribution, $y_f$ is the corresponding ground-truth label. Besides, $\hat{y}_c^k$ has been extended to the same dimension as $\hat{y}_f$, according to a class subordination strategy as illustrated in Figure 3.
While progressive granularity learning (PGL) enables the supervised task on the source domain to progressively move from coarse to fine, it does not necessarily guarantee that the hierarchical classes are aligned across domains progressively, first sup-classes and then sub-classes. Fortunately, the predicted distribution \( \hat{y} = Y(F(x)) \) conveys important discriminative information. And it is progressive as \( Y \) and \( F \) are trained to minimize coarse-fine hybrid loss. Inspired by incorporating the conditional information into the discriminator of GANs [35], we feed the progressive label \( \hat{y} \) with the feature \( f = F(x) \) into the domain discriminator \( D \) to enable progressive adversarial learning (PAL).

While it is natural to choose the concatenation of \( \hat{y} \) and \( f \) as the input to the domain discriminator \( D \), as adopted by conditional GANs [34, 42, 17, 37], such a fusion strategy is not expressive enough to model the complex relationship between \( \hat{y} \) and \( f \). Ingeniously, CDAN [29] employs outer product to replace the concatenation. However, the exploding feature dimension requires excessive memory.

To enable progressive adversarial learning, we employ a different bilinear transformation to combine the predicted distribution \( \hat{y} \) and the feature \( f \). However, though feature disruption the transferability of the features, as subtle relationships between classes are destroyed. These subtle relationships are expected to play an important role in domain adaptation when small inter-class variations greatly heighten the uncertainty of domain alignment.

3.2. Progressive Adversarial Learning

While progressive granularity learning (PGL) enables the supervised task on the source domain to progressively move from coarse to fine, it does not necessarily guarantee that the hierarchical classes are aligned across domains progressively, first sup-classes and then sub-classes. Fortunately, the predicted distribution \( \hat{y} = Y(F(x)) \) conveys important discriminative information. And it is progressive as \( Y \) and \( F \) are trained to minimize coarse-fine hybrid loss. Inspired by incorporating the conditional information into the discriminator of GANs [35], we feed the progressive label \( \hat{y} \) with the feature \( f = F(x) \) into the domain discriminator \( D \) to enable progressive adversarial learning (PAL).

While it is natural to choose the concatenation of \( \hat{y} \) and \( f \) as the input to the domain discriminator \( D \), as adopted by conditional GANs [34, 42, 17, 37], such a fusion strategy is not expressive enough to model the complex relationship between \( \hat{y} \) and \( f \). Ingeniously, CDAN [29] employs outer product to replace the concatenation. However, the exploding feature dimension requires excessive memory.

To enable progressive adversarial learning, we employ a different bilinear transformation to combine the predicted distribution \( \hat{y} \) and the feature \( f \). However, though feature
embeddings with predicted class information can enhance discriminability, it has side effect that may completely destroy the subtle differences between features in fine-grained scenarios, especially when the sample is misclassified. And missclassification is more likely to occur in fine-grained tasks. Thus it is necessary to additionally introduce a residual connection by concatenating the features. Finally, the fusion result is fed to the domain discriminator $D$:

$$ Bi(\hat{y}, f) = (\hat{y}^T Af + b) \oplus f,$$

where $A$ and $b$ are the learnable weight and bias of the bilinear transformation, and $\oplus$ is the residual concatenation.

3.3. Progressive Adversarial Network

The architecture of Progressive Adversarial Network (PAN) is shown in Figure 2. The coarse-grained predictors $C^k|_{k=1}^K$, the fine-grained label predictor $Y$, and the domain discriminator $D$ are jointly trained by unifying progressive granularity learning and progressive adversarial learning:

$$ O (G, C^k|_{k=1}^K, F, Y, D) $$

$$ = \frac{1}{n_y} \sum_{x \in S} \sum_{k=1}^K L_y(C^k(G(x)), y^k) $$

$$ + \frac{1}{n_y} \sum_{x \in S} L_h(C^k(G(x))|_{k=1}^K, Y(F(x)), y^f) $$

$$ - \lambda \sum_{x \in S \cup T} L_d(D(Bi(Y(F(x)), F(x)), d), $$

where $d$ is the domain label of $x$, $\lambda$ is a hyperparameter, and $n = n_y + n_t$. $L_y$ is the cross-entropy loss for coarse-grained recognition, which is minimized by $G$ and $C^k|_{k=1}^K$. $L_h$ is the proposed coarse-fine hybrid loss for fine-grained recognition, minimized by $Y$ and $F$. $L_d$ is the cross-entropy loss for domain discrimination, minimized by $D$ and maximized by $F$. Eventually $G, C^k|_{k=1}^K, F, Y, D$ converge to:

$$ (\hat{G}, \hat{C}^k|_{k=1}^K) = \arg \min_{G,C^k|_{k=1}^K} O (G, C^k|_{k=1}^K, F, Y, D), $$

$$ (\hat{F}, \hat{Y}) = \arg \min_{F,Y} O (G, C^k|_{k=1}^K, F, Y, D), $$

$$ (\hat{D}) = \arg \max_D O (G, C^k|_{k=1}^K, F, Y, D). $$

3.4. Theoretical Explanation

Coarse-grained models have higher generalization performance and domain adaptation on coarse-grained categories is easier. Dubey et al. [11] defines the diversity of features as the sum of the eigenvalues of the equivalent covariance matrix. And fine-grained problems are characterized as feature distributions with the following property:

$$ \sqrt{\nu(\Phi^F, P_x^F)} \ll \sqrt{\nu(\Phi^G, P_x^G)}, $$

where $\nu$ denotes the diversity of the features, $P_x^F$ is the fine-grained data distribution yielded by feature extractor $\Phi^F$, and $P_x^G$ is the generic data distribution yielded by feature extractor $\Phi^G$. The diversity in fine-grained visual categorization tasks is considered to be smaller than coarse-grained tasks. The norm of weights $\Theta$ of the final classifier layer is lower bounded by [11] ($H$ is the entropy):

$$ \|\Theta\|^2 \geq \frac{\log(C) - E_{x \sim P_x} (H(F(\cdot|x); \Theta))}{2\sqrt{\nu(\Phi, \Phi^F)}}. $$

Unlike coarse-grained categorization tasks, the fine-grained tasks generally have smaller diversity of features, which will enlarge the norm of the classifier weights $\Theta$ and make learning much more difficult. Hence, we introduce both the progressive granularity learning and progressive adversarial learning to enable domain adaptation from coarse-grained (easy) to fine-grained (hard). This progressive strategy can lower the difficulty of fine-grained domain adaptation.

4. Experiments

We evaluate the proposed PAN with state-of-the-art image classification and domain adaptation models based on deep learning architectures. Experiments are conducted on an existing benchmark CompCars [58] and two brand-new benchmarks we construct, CUB-Paintings and Birds-31.

### Table 2. Number of images in the three benchmarks.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Domains</th>
<th>#Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>CompCars  [58]</td>
<td>Web, Surveillance</td>
<td>33,780, 44,481</td>
</tr>
<tr>
<td>CUB-Paintings</td>
<td>CUB-200-2011, CUB-200-Painting</td>
<td>11,788, 3,047</td>
</tr>
<tr>
<td>Birds-31</td>
<td>CUB-200-2011, iNaturalist2017</td>
<td>1,848, 2,857</td>
</tr>
</tbody>
</table>

4.1. Datasets

4.1.1 Benchmark 1: CompCars

We evaluate PAN on CompCars [58] which can be split into two domains: Web (W) and Surveillance (S), as shown in Table 2. Only two levels of classes are available: 281
models (finer) and 68 Makes (coarser). Figure 4 are example images of the first 31 categories (models).

Notice that the car surveillance images are all in front views, with various environment conditions such as foggy, and at night, which are quite different from web images, indicating that the transfer task \( S \rightarrow W \) is extremely challenging. This may not be conducive to the evaluation of methods. So we construct another two novel benchmarks.

### 4.1.2 Benchmark II: CUB-Paintings

CUB-Paintings contains two domains: CUB-200-2011 (C) and CUB-200-Paintings (P), as shown in Table 2. Figure 5 are example images of the first 31 categories, with obvious visual domain gap. Images are organized in a four-levels hierarchy. From finer to coarser, there are 200 Species, 122 Genera, 38 Families, and 14 Orders.

CUB-200-2011 [54] is a fine-grained visual categorization benchmark with 11,788 bird images in 200 species.

CUB-200-Paintings is a dataset of bird paintings we collect from the web and filter manually. The class lists of CUB-200-Paintings and CUB-200-2011 are identical. We search Internet to collect candidate images for a total of 200 classes. Both the English common name and binomial name are adopted as retrieval keywords. Watercolors, oil paintings, pencil drawings, stamps, and cartoons are all within the scope of being selected. Then candidate images are further filtered manually. Only paintings with obvious species characteristics or with reliable labels are retained. However, this dataset needs further polishing. 3,047 images are insufficient for training very deep models, considering there are 200 categories. Potential label noise needs to be eliminated.

### 4.1.3 Benchmark III: Birds-31

There are three domains in Birds-31: CUB-200-2011 (C), iNaturalist2017 (I). Not all of the images from the original datasets are incorporated into Birds-31. The numbers of images selected are 1,848, 2,988 and 2,857 respectively. Figure 6 shows example images of all 31 categories from Birds-31. In contrast to CUB-Paintings, inter-domain variations of Birds-31 are relatively smaller. Labels are in four levels. Specifically, there are 31 Species, 25 Genera, 16 Families, and 4 Orders.

iNaturalist2017 [51] is a benchmark for iNaturalist 2017 competition. There are 5,089 categories in it, with 579,184 training images and 95,986 validation images.

We employ binomial nomenclature to categorize objects from these three datasets, and then get the intersection, 123 categories. As Benchmark II contains up to 200 categories and numbers of samples vary greatly across domains, 31 categories with a balanced sample size are selected finally.

### 4.2. Implementation

We implement all deep methods in PyTorch and we use NVIDIA Titan RTX for training. We fine-tune ResNet-50 [16] model pre-trained on ImageNet. The classifier layers are trained from scratch, and their learning rate is set 10 times that of the other layers. We adopt mini-batch SGD with momentum of 0.9. Batch size is fixed to 36. The learning rate strategy is the same as [12]. Consistent with [12], hyperparameter \( \lambda \) changes from 0 to 1 following a schedule of \( \lambda = \frac{1-\exp(-10\rho)}{1-\exp(-10\rho)} \) in all experiments. For fair comparison, all parameters are not changed across all transfer tasks.

### 4.3. Results

We evaluate Progressive Adversarial Networks (PAN) and report the average classification accuracy based on three random experiments. In addition to the widely-used baseline Domain Adversarial Neural Network (DANN) [12], we also compare PAN with generic visual categorization, fine-grained visual categorization and domain adaptation methods: ResNet-50 [16], Inception-v3 [46], Bilinear CNN [25], Deep Adaptation Network (DAN) [28], Joint Adaptation Network (JAN) [31], Adversarial Discriminative Domain Adaptation (ADDA) [48], Multi-Adversarial Domain Adaptation (MADA) [39], Maximum Classifier Discrepancy (MCD) [41], Conditional Adversarial Domain Adaptation (CDAN) [29], Batch Spectral Penalization (BSP) [5], and Stepwise Adaptive Feature Norm (SAFN) [56].
On CompCars as in Table 4, our method performs best across both transfer tasks. It outperforms CDAN+BSP, the second best method, by 2.1 percent. On CUB-Paintings as in Table 5, our method achieves the best performance across all two transfer tasks. We raise average accuracy from the baseline DANN of 50.28% to 59.16%, a boost of more than 8 percent. On Birds-31 as in Table 3, our method achieves the highest average accuracy and the best performance across all six tasks. The accuracy is improved by about 5 percent compared to DANN.

Note that PAN yields larger boosts on CompCars and CUB-Paintings than on Birds-31. There are two reasons. First, the inter-domain variations of the former are much larger than the latter, as shown in Figures 4, 5, and 6. Small inter-domain variations imply less gain by bridging the domain gap. Second, the classification accuracy of Birds-31 is generally higher, leaving us with a relatively smaller room for improvement. For example, in task N → C, the accuracy of most methods is about 90%. And, as some neighboring categories are visually indistinguishable, the performance of expert annotators is only 93% [6].

4.4. Analyses

Ablation Study. Removing PGL and preserving PAL, we denote the remainder of PAN by PAN-w.o.-Pro. or PAL Only. Note that without PGL, PAL Only is not progressive any more. The accuracy decreases sharply with PAL Only (Table 6). We also testify the concatenation operator in PAL. PAL outperforms PAL (w/o concat), proving that the concatenation operator can prevent the model from destroying subtle differences between features. PGL Only still outperforms the baseline DANN by 4 percent.

Hierarchy Selection. PAN exploits labels at all levels. On the dataset CUB-Paintings, coarse-grained labels are at three levels: genus, family, and order. We analyzed the variants of PAN with coarse-grained labels at only one level, with improved results by PAN shown in Table 7.

Curriculum Schedule. The curriculum schedule that
the $\varepsilon$ in Equation (2) follows the same as that of $\lambda$ in Equation (6). This simple and commonly-used strategy [12] outperforms the others, as shown in Table 8.

Table 6. Ablation Study: Accuracy (%) on CUB-Paintings.

<table>
<thead>
<tr>
<th>Method</th>
<th>C $\rightarrow$ P</th>
<th>P $\rightarrow$ C</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAL (w/o concat)</td>
<td>62.46±0.30</td>
<td>45.32±0.37</td>
<td>53.89</td>
</tr>
<tr>
<td>PAL Only</td>
<td>63.05±0.19</td>
<td>45.83±0.33</td>
<td>54.44</td>
</tr>
<tr>
<td>PGL Only</td>
<td>61.04±0.29</td>
<td>46.69±0.12</td>
<td>53.87</td>
</tr>
<tr>
<td>PAN (PGL+PAL)</td>
<td>67.40±0.02</td>
<td>50.92±0.26</td>
<td>59.16</td>
</tr>
</tbody>
</table>

Table 7. Accuracy (%) of PAN with different coarse-grained label levels on CUB-Paintings.

<table>
<thead>
<tr>
<th>Level</th>
<th>Num</th>
<th>C $\rightarrow$ P</th>
<th>P $\rightarrow$ C</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genus</td>
<td>122</td>
<td>65.37±0.46</td>
<td>48.33±0.35</td>
<td>56.85</td>
</tr>
<tr>
<td>Family</td>
<td>38</td>
<td>65.51±0.37</td>
<td>48.02±0.16</td>
<td>56.76</td>
</tr>
<tr>
<td>Order</td>
<td>14</td>
<td>66.32±0.34</td>
<td>49.43±0.23</td>
<td>57.88</td>
</tr>
<tr>
<td>Class</td>
<td>1</td>
<td>64.68±0.23</td>
<td>46.92±0.30</td>
<td>55.80</td>
</tr>
<tr>
<td>G+F+O</td>
<td>–</td>
<td>67.40±0.02</td>
<td>50.92±0.26</td>
<td>59.16</td>
</tr>
</tbody>
</table>

Table 8. Accuracy (%) of PAN with different curriculum strategies on CUB-Paintings.

<table>
<thead>
<tr>
<th>Schedule</th>
<th>Ours</th>
<th>Linear</th>
<th>Step</th>
<th>Exponential</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg</td>
<td>59.16</td>
<td>54.67</td>
<td>54.88</td>
<td>55.19</td>
</tr>
</tbody>
</table>

Distribution Discrepancy. In domain adaptation theory [2, 33], A-distance is a measure of inter-domain variation:

$$d_A = 2(1 - 2err), \quad (10)$$

where $err$ is the error rate of a classifier that is trained to discriminate the source domain and the target domain. Figure 7(a) depicts $d_A$ on transfer tasks C $\rightarrow$ P and P $\rightarrow$ C, with features extracted by ResNet-50, DANN, PAN-w.o.-Pro. and PAN. It is notable that $d_A$ on features extracted by PAN are the smallest on both transfer tasks, which implies that these features are more transferable across domains.

Ideal Joint Hypothesis. The expected error $\mathcal{E}_T(h)$ of a hypothesis $h$ on the target domain can be bounded as [2]

$$\mathcal{E}_T(h) \leq \mathcal{E}_S(h) + \frac{1}{2} d_{H\Delta H}(S, T) + \varepsilon_{\text{ideal}}, \quad (11)$$

where $\mathcal{E}_S(h)$ is the source error, $d_{H\Delta H}(S, T)$ is the $H\Delta H$-distance measuring the domain shift, and $\varepsilon_{\text{ideal}}$ is the error of an ideal joint hypothesis $h^* = \min_h \mathcal{E}_S(h) + \mathcal{E}_T(h)$ on labeled source and target domains. $\varepsilon_{\text{ideal}}$ is defined as

$$\varepsilon_{\text{ideal}} = \mathcal{E}_S(h^*) + \mathcal{E}_T(h^*), \quad (12)$$

which measures the discriminability of features. For further analysis of our method, we investigate this indicator of discriminability. The average error rate of the new classifier trained on the labeled data of source and target domains is half of $\varepsilon_{\text{ideal}}$. The results are shown in Figure 7(b). As expected, PAN enhances the discriminability of features.

Feature Diversity. Figure 7(c) is a plot of the top 2 principal components (PCs) of features of ResNet-50 trained from fine-grained (red) and coarse-grained (blue) labels on CUB-200-2011, following the experiment in [11]. Fine-grained features are concentrated with less diversity, in accordance with the theoretical analysis in Section 3.4.

Weight Sharing. The feature extractors should not share weights. Differences between fine-grained features are subtle and sharing weights destroys the subtle differences crucial for discriminability. Using weight sharing, the average accuracy on CUB-Paintings drops from 59.16% to 51.48%.

5. Conclusion

In this paper, we proposed the Progressive Adversarial Networks (PAN) to solve the fine-grained domain adaptation problem with only hierarchical image-level labels. The key idea of our model is to align the corresponding classes across domains from coarse-grained to fine-grained, first sup-classes and then sub-classes. We also theoretically explained the proposed approach from the perspective of feature diversity. We compared PAN with prior works on three benchmarks for fine-grained domain adaptation. And experimental results testified the effectiveness of our method.

6. Acknowledgments

This work was supported in part by the Natural Science Foundation of China (61772299, 71690231), and the MOE Strategic Research Project on Artificial Intelligence Algorithms for Big Data Analysis.
References


