Bayesian Uncertainty Matching for Unsupervised Domain Adaptation

Jun Wen 1,2, Nenggan Zheng1,2*, Junsong Yuan3, Zhefeng Gong4 and Changyou Chen3
1Qiushi Academy for Advanced Studies, Zhejiang University, Hangzhou, China
2College of Computer Science and Technology, Zhejiang University, Hangzhou, China
3Computer Science and Engineering Department, State University of New York at Buffalo
4Department of Neurobiology, Zhejiang University School of Medicine, Hangzhou, China
{junwen,zfgong}@zju.edu.cn, zng@cs.zju.edu.cn, {jsyuan,changyou}@buffalo.edu

Abstract

Domain adaptation is an important technique to alleviate performance degradation caused by domain shift, e.g., when training and test data come from different domains. Most existing deep adaptation methods focus on reducing domain shift by matching marginal feature distributions through deep transformations on the input features, due to the unavailability of target domain labels. We show that domain shift may still exist via label distribution shift at the classifier, thus deteriorating model performances. To alleviate this issue, we propose an approximate joint distribution matching scheme by exploiting prediction uncertainty. Specifically, we use a Bayesian neural network to quantify prediction uncertainty of a classifier. By imposing distribution matching on both features and labels (via uncertainty), label distribution mismatching in source and target data is effectively alleviated, encouraging the classifier to produce consistent predictions across domains. We also propose a few techniques to improve our method by adaptively reweighting domain adaptation loss to achieve nontrivial distribution matching and stable training. Comparisons with state of the art unsupervised domain adaptation methods on three popular benchmark datasets demonstrate the superiority of our approach, especially on the effectiveness of alleviating negative transfer.

1 Introduction

Many machine-learning algorithms assume that training and test data, typically in terms of feature-label pairs, denoted as \( \{x_i, y_i\}_i \), are drawn from the same feature-label space with the same distribution, where \( x_i \) is the feature while \( y_i \) is the label of \( x_i \). However, this assumption rarely holds in practice as the data distribution is likely to change over time and space. Though state-of-the-art deep convolutional features have shown invariant to low-level variations to some degree, they are still susceptible to domain-shift, as it is expensive to manually label sufficient training data that cover diverse application domains. A typical solution is to further finetune a learned deep model on task-specific datasets. However, it is still prohibitively difficult and expensive to obtain enough labeled data for finetuning on a big deep network. Instead of re-collecting labeled data for every possible new task, unsupervised domain-adaptation methods are adopted to alleviate performance degradations by transferring knowledge from related labeled source domains to an unlabeled target domain [Ganin et al., 2016; Li et al., 2017; Zhou et al., 2019].

When adopting domain adaptation, certain assumptions must be imposed on how distributions change across domains. For instance, most existing domain adaptation methods consider a covariate shift situation where the distributions on source and target domains only differ in the marginal feature-distribution \( P(X) \), with an identical conditional distribution \( P(Y|X) \) assumption. Here we use \( X \) and \( Y \) to denote random variables whose realizations are features \( x_i \) and labels \( y_i \), either from the source data \((X_s, Y_s)\) or target data \((X_t, Y_t)\). In this setting, an early attempt is to match the feature distribution \( P(X) \) on source and target domains by importance reweighting [Huang et al., 2007]. State-of-the-art approaches reduce domain-shift by learning domain-invariant representations through deep neural transformations \( G_\phi(X) \), parameterized by \( \phi \), such that \( P(G_\phi(X_s)) \approx P(G_\phi(X_t)) \). This is often

*Corresponding author.
achieved by optimizing a deep network to minimize some distribution-discrepancy measures [Sun and Saenko, 2016; Tzeng et al., 2017]. Because there is no label in the target domain for unsupervised domain adaptation, most existing methods simply assume $P(Y_t | G_\phi(x_t)) \approx P(Y_t | G_\phi(x_s))$ by sharing a classifier learned with source labeled data only. However, this is typically not true in practice as the source-learned classifier tends to be biased toward the source. As shown in Figure 1 (a), though the feature-distributions are well matched, the classifiers may still perform poorly in the target domain due to label-distribution mismatch.

In this paper, we alleviate the above problem by proposing an approximate joint-distribution matching scheme. Specifically, due to the lack of label information in a target domain, we propose to match the model prediction uncertainty, a second-order statistic equivalent, induced by the conditional distribution $P(Y_t | G_\phi(X))$. We obtain the prediction uncertainty by imposing a Bayesian neural network (BNN) which induces posterior distributions over weights of a neural network. Without uncertainty matching, the BNN classifier is expected to produce high uncertainty for the unseen target-domain data, due to the bias induced by training on the source domain. By contrast, with prediction uncertainty matching, one is able to achieve an approximate joint-distribution matching, alleviating domain-shift on the classifier. The contributions of our work are summarized as follows:

- Different from most existing domain-adaptation methods, which only focus on reducing marginal feature-distribution discrepancy, we propose to match joint feature-label distributions by exploiting model prediction uncertainty, effectively alleviating conditional-distribution shift imposed by the classifier.

- We employ BNNs to quantify prediction uncertainty. Through additional source and target uncertainty discrepancy minimization, both fine-grained marginal feature-distribution and conditional label-distribution matching are achieved.

- Extensive experimental results on standard domain-adaptation benchmarks demonstrate the effectiveness of the proposed method, outperforming current state-of-the-art approaches.

2 Related Works

Domain Adaptation. Domain adaptation methods seek to learn discriminative features from neighbouring source domains to target domains. This is usually achieved by learning domain-invariant features [Ben-David et al., 2010]. Previous methods usually seek to align source and target feature through subspace learning [Gong et al., 2012]. Recently, deep adversarial-domain-adaptation approaches have taken over and achieved state-of-the-art performances [Hoffman et al., 2018; Wen et al., 2019]. These methods attempt to reduce domain discrepancy by optimizing deep networks with an adversarial objective produced by a discriminator network that is trained to distinguish features of target from source domains. Though significant marginal distribution-shift can be reduced, these methods fail to fully address the conditional label-distribution shift problem. There are some recent models trying to address this issue by utilizing pseudo-labels [Long et al., 2018; Chen et al., 2018]. However, most of them are deterministic models, which can not essentially reduce the conditional domain-shift, due to the unavailability of target-domain labels.

Bayesian Uncertainty. Uncertainty can be achieved by adopting Bayesian neural networks. A typical BNN assigns a prior distribution, e.g., a Gaussian prior distribution, over the weights, instead of deterministic weights as in standard neural networks. Given observed data, approximate inference is performed to calculate posterior distribution of the weights, such as the methods in [Graves, 2011; Blundell et al., 2015]. A more effective way to calculate Bayesian uncertainty is to employ the dropout variational inference [Gal and Ghahramani, 2016], which is adopted in this paper.

3 The Proposed Method

3.1 The Overall Idea

Given a labeled source-domain dataset $D_s = ({X_s}, {Y_s})$ and an unlabeled target-domain dataset $D_t = ({X_t})$, the goal of unsupervised domain-adaptation is to learn an adapted model from the labeled source-domain data to the unlabeled target-domain data. The source and target domains are assumed to be sampled from two joint distributions $P_s(X_s, Y_s)$ and $P_t(X_t, Y_t)$, respectively, with $P_s \neq P_t$. The joint distribution of feature-label pairs can be decomposed as:

$$P(X, Y) = P(Y | X)P(X).$$ (1)

Limitations of Traditional Methods. Most existing domain-adaptation methods reduce domain-shift by learning a deep feature-transformation $G_\phi$ such that $P(G_\phi(X_s)) \approx P(G_\phi(X_t))$, and a shared classifier network $P_\theta(Y_s | G_\phi(X_s))$, parameterized by $\theta$, using labeled source data $D_s$. To adapt to a target domain, the learned $P_\theta(Y_s | G_\phi(X_s))$ is adopted to form the target-domain joint distribution $P(G_\phi(X_t))P_\theta(Y_t | G_\phi(X_t))$. It is easy to see that directly adopting $P_\theta(Y_s | G_\phi(X_s))$ in the target-domain is unable to match the true joint distributions $P_s(X_s, Y_s)$ and $P_t(X_t, Y_t)$, as $P_\theta(Y_s | G_\phi(X_s))$ only reflects feature-label information in the source domain.

Our Method

In this paper, we propose to jointly reduce the marginal-distribution shift ($P(X_s) \neq P(X_t)$) and conditional-distribution shift ($P(Y_s | X_s) \neq P(Y_t | X_t)$) by exploiting prediction uncertainty. Specifically, our model consists of a probabilistic BNN feature extractor $G_\phi$ with inputs $X_s$ or $X_t$, and a BNN classifier $C_\theta$ with inputs $G_\phi(X_s)$ or $G_\phi(X_t)$. The classifier $C_\theta$, which corresponds to the conditional distribution $P(Y | G_\phi(X))$ and is parameterized by $\theta$, learns to classify samples from both domains.

As discussed in the Introduction, directly learning to match $P(Y_s | G_\phi(X_s))$ and $P(Y_t | G_\phi(X_t))$ is unfeasible due to the unavailability of target labels. To overcome the difficulty, we instead learn to match the prediction uncertainty, a second-order statistics equivalent. The intuition is that if the second-
We employ Bayesian neural network (BNN) to quantify model uncertainty. Denote the weights with high uncertainty by a source-trained classifier. If one dropout probability, \( \hat{\vartheta} \), is performed to generate approximate samples from the posterior distribution. This approach is equivalent to using a Bernoulli variational distribution \( q_{\vartheta}(W) \) [Gal and Ghahramani, 2016], parameterized by \( \vartheta \), to approximate the true model weights \( (W) \) posterior. As proven in [Gal and Ghahramani, 2016], the dropout inference essentially minimizes the KL divergence between the approximate distribution and the posterior of a deep Gaussian process. For classification, the objective can be formulated as:

\[
    \mathcal{L}(\vartheta, \rho) = -\frac{1}{N} \sum_{i=1}^{N} \log p(y_i | f^{W_i}(x_i)) + \frac{1 - \rho}{2N} ||\vartheta||^2, \tag{2}
\]

where \( N \) is the number of training samples, \( p \) denotes the dropout probability, \( W_i \) is sampled according to the dropout variational distribution \( q_{\vartheta}(W) \) [Gal and Ghahramani, 2016], and \( \vartheta \) is the set of the variational distribution’s parameters.

The final prediction can be obtained by marginalizing over the approximate posterior distribution on weights, which is approximated using Monte Carlo integration as follows:

\[
p(y_i = c | x_i, X, Y) = \frac{1}{T} \sum_{t=1}^{T} \text{Softmax}(f^{W_t}(x_i)), \tag{3}
\]

with \( T \) sampled masked weights, namely forwarding each sample \( x_t \) through the feature extractor \( G_{\phi} \) and classifier \( C_{\theta} \) for \( T \) times with weights sampled according to the dropout inference. The uncertainty of the prediction can be summarized using different metrics. In this paper, we use two metrics: 1) entropy of the averaged probabilistic prediction, and 2) variance of all prediction vectors. The entropy and variance based prediction uncertainty are denoted as \( \mathcal{U}_{\text{entro}} \) and \( \mathcal{U}_{\text{var}} \), respectively, formulated as:

\[
\mathcal{U}_{\text{entro}}(x_i) = H(\frac{1}{T} \sum_{t=1}^{T} \text{Softmax}(C_{\theta}(G_{\phi}(x_i))/\tau)), \tag{4}
\]

\[
\mathcal{U}_{\text{var}}(x_i) = \frac{1}{T} \sum_{t=1}^{T} (C_{\theta}(G_{\phi}(x_i)) - \frac{1}{T} \sum_{t=1}^{T} C_{\theta}(G_{\phi}(x_i)))^2, \tag{5}
\]

where \( H(\cdot) \) denotes the information entropy function and \( \tau \) the temperature of the Softmax, which controls the uncertainty level.

### 3.3 Distribution Adaptation

In this section, we describe how to simultaneously alleviate the marginal and conditional domain-shift by matching the approximate joint distributions of the source and target feature-label pairs.

**Joint-Distribution Adaptation**

We employ adversarial learning to match source and target statistics to reduce distribution discrepancy, as adversarial domain-adaptation methods have achieved state-of-the-art performances [Goodfellow et al., 2014; Tzeng et al., 2017]. Basically, the procedure is described by a two-player game. The first player, a domain discriminator \( D \), is trained to distinguish source from target data; while the second player, the feature extractor \( G_{\phi} \), is trained to learn features that confuse the domain discriminator. By learning a best possible discriminator, the feature extractor is expected to learn features that are best domain-invariant. This learning procedure can be described by the following minimax game:

\[
\min_{G_{\phi}} \max_{D} \mathcal{L}_{\text{adv}} = -\frac{1}{n_s} \sum_{i=1}^{n_s} \log(D(G_{\phi}(x_i^s))) - \frac{1}{n_t} \sum_{i=1}^{n_t} \log(1 - D(G_{\phi}(x_i^t))), \tag{6}
\]

where \( n_s \) and \( n_t \) are the number of training samples from source and target, respectively.

However, this typical adversarial minimax game for domain adaptation may be problematic in two aspects: 1) trivial feature alignment; and 2) unstable training. The domain discriminator fails to consider the relationship between learned features and the decision boundary of the classifier during feature alignment, which may lead to boundary target samples or trivial alignment with a huge-capacity \( G_{\phi} \) [Shu et al., 2018].
We aim to achieve nontrivial feature alignment by enforcing additional classifier prediction consistency during matching. Furthermore, noisy or hard-to-match samples may lead to unstable adversarial training. These confusing samples, which typically endow high prediction uncertainty, may produce unreliable gradients and deteriorate the training. They may also direct the $G_\phi$ to learn features that is non-discriminative for classifying target samples, especially with a huge-capacity $G_\phi$. Thus, we aim to attenuate the influence of noisy samples and reinforce the influence of easy-to-match target samples by adaptively re-weighting the adversarial loss. Specifically, we propose the following modified objective:

$$
\min_{\alpha} \max_{\phi} \mathcal{L}_{adv} = -\frac{1}{n_s} \sum_{i=1}^{n_s} \left( \alpha_u x_i^s \log(D(G_\phi(x_i^s), \mathcal{U}(x_i^s))) - \frac{1}{n_t} \sum_{i=1}^{n_t} (\alpha_x t_i \log(1 - D(G_\phi(x_i^t), \mathcal{U}(x_i^t)))) \right),
$$

(7)

where $\mathcal{U}(\cdot)$ is the prediction uncertainty formulated in Equation (4) or Equation (5). Both $\alpha_x$ and $\alpha_u$ are the adaptation loss weights, defined as:

$$
\alpha_x = \begin{cases} 
0 & \mathcal{U}(x_i) > t_u \\
\frac{N \exp(-\tau(x_i))}{\sum_{i=1}^{N} \exp(-\tau(x_i))} & \mathcal{U}(x_i) \leq t_u,
\end{cases}
$$

(8)

where $N$ is the number of training samples and $t_u$ denotes the uncertainty threshold constraining the influence of samples with uncertainty larger than $t_u$. For samples with uncertainty less than $t_u$, the weights are normalized within each training batch with more attention paid on the certain samples. It is worth noting that we found directly using the uncertainty without normalization for the re-weighting as done in [Kendall and Gal, 2017; Long et al., 2018] tends to discourage a model from predicting low uncertainty for all samples. With such an adaptive joint-distribution adaptation objective, we aim to achieve non-trivial feature alignment and enable safer transfer.

### Conditional-Distribution Adaptation

Note the joint-distribution-matching scheme described in the last section does not necessarily guarantee a good conditional-distribution adaptation. In this section, we aim to reduce the conditional distribution shift and learn a domain-invariant classifier. Due to the infeasibility of directly minimizing the conditional distribution discrepancy $\|P_\theta(Y|G_\phi(X_s)) - P_\theta(Y|G_\phi(X_t))\|_q$, we propose to approximate it by matching prediction uncertainty, a second-order statistic equivalent, with a BNN as the classifier. We exploit prediction uncertainty to detect and quantify domain-shift of a classifier. By minimizing the uncertainty discrepancy between source and target, we aim to approximately reduce the domain-shift of the classifier; and the objective $\mathcal{L}_u$ can be formulated as:

$$
\mathcal{L}_u = \|\mathcal{U}(X_s) - \mathcal{U}(X_t)\|_q,
$$

(9)

where we set $q = 2$ as we found it achieves better performances than $q = 1$. The prediction uncertainty discrepancy is estimated within each batch during training.

To enable discriminative feature transferring, the feature extractor $G_\phi$ and classifier $C_\theta$ are also trained to minimize the source supervised loss $\mathcal{L}_c$ using source labels, defined as:

$$
\mathcal{L}_c = -\frac{1}{n_s} \sum_{i=1}^{n_s} y_i^s \cdot \log \text{Softmax}(C_\theta(G_\phi(x_i^s))/\tau_c),
$$

(10)

where $y_i^s$ is the true label of the source sample $x_i^s$ and $\tau_c$ is the Softmax temperature for source classification.

Integrating all objectives together, the final learning procedure is formulated as:

$$
\min_{G_\phi, C_\theta} \max_{\phi} \mathcal{L}_{final} = \mathcal{L}_c + \lambda_{adv} \mathcal{L}_{adv} + \lambda_u \mathcal{L}_u,
$$

(11)

where $\lambda_{adv}$ and $\lambda_u$ are hyper-parameters that trade-off the objectives in the unified optimization problem.

According to the analysis of [Ben-David et al., 2010], the expected target error is upper-bounded by the following three terms: 1) source error, 2) domain divergence, and 3) conditional-distribution discrepancy across domains. We aim to improve the marginal distribution matching to reduce the second term by minimizing $\mathcal{L}_{adv}$ to achieve joint feature-uncertainty adaptation. While the third term is ignored by most of existing domain adaptation methods, we are able to reduce it via uncertainty matching and $\mathcal{L}_u$ minimization.

### 4 Experiments

We compare our method with state-of-the-art domain-adaptation approaches on several benchmark datasets: **USPS-MNIST-SVHN** dataset [Hoffman et al., 2018], **Office-31** dataset [Saenko et al., 2010], and the recently introduced **Office-home** dataset [Venkateswara et al., 2017].

**USPS-MNIST-SVHN.** This dataset is used for digits recognition with 3 domains: MNIST, USPS, and SVNH. MNIST is composed of gray images of size $28 \times 28$; USPS contains $16 \times 16$ grey digits; and SVHN consists of $32 \times 32$ color digits images, which are more challenging and might contain more than one digit in each image. We evaluate our method using the three typical adaptation tasks: USPS→MNIST (two tasks) and SVHN→MNIST (one task). Following the same evaluation protocol of [Hoffman et al., 2018], we use the standard training sets for domain-adaptation training and report adaptation results on the test sets.

**Office-31.** This dataset is widely used for visual domain adaptation [Saenko et al., 2010]. It consists of 4,652 images and 31 categories collected from three different domains: Amazon (A) from amazon.com, Webcam (W) and DSLR (D), taken by web camera and digital SLR camera in different environmental settings, respectively. We evaluate all methods on the following four challenging settings: A→W and A→D.

**Office-home.** This is one of the most challenging visual domain adaptation datasets [Venkateswara et al., 2017], which consists of 15,588 images with 65 categories of everyday objects in office and home settings. There are four significantly different domains: Art (Ar) consisting of 2427 painting, sketches or artistic depiction images, Clipart (Cl) containing 4365 images, Product (Pr) with 4439 images and Real-World (Rw) comprising of 4357 regularly captured images. We report performances of all the 12 adaptation tasks to enable thorough evaluations: Ar→Cl, Ar→Pr, Ar→Rw, Cl→Pr, Cl→Rw, and Pr→Rw.
We dropout all fully-connected layers with a dropout ratio 
\[ q = 0.5. \] Improvements are not observed with further dropout on convolution layers.

### 4.2 Results

The results on the digit recognition task are shown in Table 1. Our(Entro) achieves the best performances on most of the tasks. The CyCADA align features at both pixel-level and feature-level. RAAN alleviates conditional distribution shift by matching label distributions. CADN-M attempts to learn domain-invariant interactions between learned features and classifier through conditional adversarial learning. On these tasks, the plenty of source labels prevents the low-capacity LeNet-like model from overfitting the source labels, thus the advantages of our method over DAAN and CADN-M mainly come from uncertainty discrepancy minimization that alleviates the classifier bias.

Our(Entro) consistently outperforms Our(Var). The distinct performance gap can be explained as follows. The entropy captures the cross-category probability spread of the prediction while the variance measures the deviation of prediction probabilities around the mean. The entropy uncertainty is more sensitive to the multi-peak probability spread across different categories. During training, the output probabilities of unmatched or boundary target samples usually cluster around two or more peaks, namely uncertain among several neighboring categories. In this case, the variance measure would obfuscate this multi-peak information. In the following, we only report the performances of Our(Entro).

Performances on the Office-31 and Office-home datasets are reported in Table 2 and Table 3, respectively. Again, our model achieves the best performances on most of the tasks. Due to the smaller size of the labeled source dataset and the huge capacity of the AlexNet, the models easily overfit the source labels while being jointly trained to reduce the marginal distribution discrepancy. The overfitting harms the transferability of the aligned features, resulting in learning trivial features for the target domain. In this case, our model alleviates this problem by jointly enforcing feature alignment and classifier prediction consistency.

**Negative Transfer.** Negative transfer happens when features are falsely aligned and domain adaptation causes deteriorated performances. Existing marginal distribution matching methods easily induce negative transfer when the marginal distributions between source and target are inherently different, e.g., the source domain is smaller or larger than the target. We conduct experiments on the Office-31 dataset with the 31$\rightarrow$25 task by removing 6 classes from the target, and the 25$\rightarrow$25(6+) task by treating 6 extra target classes as noise images. We compare our method with DANN and MADA [Pei et al., 2018] which is showed effective on alleviating negative transfer. The results are reported in Table 4. It is seen that DANN suffers obvious negative transfer on the 31$\rightarrow$25 task. The effectiveness of our method on alleviating negative transfer is significant. Adaptive joint feature-uncertainty distribution matching encourages the model to mix source and target samples that best match with each other, thus alleviating the harmful effects of noisy samples.

---

**Table 1:** Accuracy (%) of unsupervised domain adaptation on digit recognition tasks.

<table>
<thead>
<tr>
<th>Method</th>
<th>SVHN$\rightarrow$MNIST</th>
<th>MNIST$\rightarrow$USPS</th>
<th>USPS$\rightarrow$MNIST</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDA</td>
<td>76.0 ± 1.8</td>
<td>89.4 ± 0.2</td>
<td>90.1 ± 0.8</td>
<td>85.2</td>
</tr>
<tr>
<td>RAAN</td>
<td>89.2</td>
<td>89.0</td>
<td>92.1</td>
<td>90.1</td>
</tr>
<tr>
<td>LFPDA</td>
<td>86.9 ± 0.5</td>
<td>92.2 ± 0.4</td>
<td>92.5 ± 0.3</td>
<td>90.5</td>
</tr>
<tr>
<td>CyCADA</td>
<td>90.4 ± 0.4</td>
<td>95.6 ± 0.2</td>
<td>96.5 ± 0.1</td>
<td>94.2</td>
</tr>
<tr>
<td>CDANN-M</td>
<td>89.2</td>
<td>96.5</td>
<td>97.1</td>
<td>94.3</td>
</tr>
<tr>
<td>Ours(Var)</td>
<td>80.3 ± 0.7</td>
<td>93.5 ± 0.4</td>
<td>94.7 ± 0.3</td>
<td>89.5</td>
</tr>
<tr>
<td>Ours(Entro)</td>
<td>91.5 ± 0.5</td>
<td>95.7 ± 0.4</td>
<td>98.1 ± 0.2</td>
<td>95.1</td>
</tr>
</tbody>
</table>

**Table 2:** Accuracy (%) on the Office31 dataset for unsupervised domain adaptation.

<table>
<thead>
<tr>
<th>Method</th>
<th>A$\rightarrow$W</th>
<th>A$\rightarrow$D</th>
<th>W$\rightarrow$A</th>
<th>D$\rightarrow$A</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet</td>
<td>61.6 ± 0.4</td>
<td>62.8 ± 0.5</td>
<td>49.8 ± 0.4</td>
<td>51.1 ± 0.6</td>
<td>56.6</td>
</tr>
<tr>
<td>DANN</td>
<td>73.0 ± 0.5</td>
<td>72.3 ± 0.3</td>
<td>51.2 ± 0.5</td>
<td>53.4 ± 0.4</td>
<td>62.5</td>
</tr>
<tr>
<td>ADDA</td>
<td>73.5 ± 0.6</td>
<td>71.6 ± 0.4</td>
<td>53.5 ± 0.6</td>
<td>54.6 ± 0.5</td>
<td>63.3</td>
</tr>
<tr>
<td>LFPDA</td>
<td>75.2 ± 0.3</td>
<td>72.1 ± 0.5</td>
<td>54.2 ± 0.5</td>
<td>56.9 ± 0.5</td>
<td>64.6</td>
</tr>
<tr>
<td>JAN</td>
<td>74.9 ± 0.3</td>
<td>71.8 ± 0.2</td>
<td>55.0 ± 0.4</td>
<td>58.3 ± 0.3</td>
<td>63.0</td>
</tr>
<tr>
<td>CDANN-M</td>
<td>78.3 ± 0.2</td>
<td>76.3 ± 0.1</td>
<td>57.3 ± 0.3</td>
<td>57.3 ± 0.2</td>
<td>62.3</td>
</tr>
<tr>
<td>Ours(Entro)</td>
<td>78.9 ± 0.4</td>
<td>77.8 ± 0.3</td>
<td>56.6 ± 0.5</td>
<td>57.4 ± 0.4</td>
<td>67.7</td>
</tr>
</tbody>
</table>

**Comparied Methods.** The state-of-the-art deep domain-adaptation methods we compared include: Domain Adversarial Neural Network (DANN) [Ganin et al., 2016], Adversarial Discriminative Domain Adaptation (ADDA) [Tzeng et al., 2017], Joint Adaptation Networks(JAN) [Long et al., 2017], Conditional Domain Adversarial Network (CADN) [Long et al., 2018], Cycle-Consistent Adversarial Domain Adaptation (CyCADA) [Hoffman et al., 2018], Re-weighted Adversarial Adaptation Network (RAAN) [Chen et al., 2018], Local Feature Patterns for Domain Adaptation (LFPDA) [Wen et al., 2019]. We follow standard evaluation protocols of unsupervised domain adaptation as in [Long et al., 2017]. For our model, we report performances with uncertainty estimated with entropy and variance formulations, denoted as Our(Entro) and Our(Var), respectively.

**4.1 Implementation Details**

**CNN Architectures.** For digit classification datasets, we use the same architecture as in ADDA [Tzeng et al., 2017]. All digit images are resized to $28 \times 28$ for fair comparisons.

On the Office-31 and the Office-home datasets, we finetune the AlexNet pre-trained from the ImageNet. Following the DANN [Ganin et al., 2016], a bottleneck layer $fc_7$ with 256 units is added after the $fc_7$ layer for adaptation. We adopt the same image random flipping and cropping strategy as in JAN [Long et al., 2017].

**Hyper-parameters.** To enable stable training, we progressively increase the importance of the adaptation loss and set $\lambda_{adv} = \frac{1}{\exp(\gamma m)} - 1$, where $\gamma = -10$ and $m$ denotes the training progress ranging from 0 to 1. We use a similar hyper-parameter selection strategy as in DANN, called reverse validation. We set $\lambda_u = 0.25 \lambda_{adv}$ to ensure uncertainty reduction. With $\gamma = 1.5$, we forward each sample $T = 12$ times to obtain prediction uncertainty. We set $t_u = 0.2$, for adaptation loss re-weighting, and $\tau_c = 1.8$ for source classification loss. We dropout all fully-connected layers with a dropout ratio $q = 0.5$. Improvements are not observed with further dropout on convolution layers.
DANN significantly reduces marginal discrepancy while with much less boundary target features comparing to DANN (best viewed in color).

Table 3: Accuracy (%) on the Office-home dataset for unsupervised domain adaptation.

<table>
<thead>
<tr>
<th>Method</th>
<th>Ar→Cl</th>
<th>Ar→Pr</th>
<th>Ar→Rw</th>
<th>Cl→Ar</th>
<th>Cl→Pr</th>
<th>Cl→Rw</th>
<th>Pr→Ar</th>
<th>Pr→Cl</th>
<th>Pr→Rw</th>
<th>Rw→Ar</th>
<th>Rw→Cl</th>
<th>Rw→Pr</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet</td>
<td>26.3</td>
<td>32.6</td>
<td>41.3</td>
<td>22.1</td>
<td>41.7</td>
<td>42.1</td>
<td>20.5</td>
<td>20.3</td>
<td>51.1</td>
<td>31.0</td>
<td>27.9</td>
<td>54.9</td>
<td>34.3</td>
</tr>
<tr>
<td>DANN</td>
<td>36.4</td>
<td>45.2</td>
<td>54.7</td>
<td>35.2</td>
<td>51.8</td>
<td>55.1</td>
<td>31.6</td>
<td>39.7</td>
<td>59.3</td>
<td>45.7</td>
<td>46.4</td>
<td>65.9</td>
<td>47.3</td>
</tr>
<tr>
<td>JAN</td>
<td>35.5</td>
<td>46.1</td>
<td>57.7</td>
<td>36.4</td>
<td>53.3</td>
<td>54.5</td>
<td>33.4</td>
<td>40.3</td>
<td>60.1</td>
<td>45.9</td>
<td>47.4</td>
<td>67.9</td>
<td>48.2</td>
</tr>
<tr>
<td>CDAN-M</td>
<td>38.1</td>
<td>50.3</td>
<td>60.3</td>
<td>39.7</td>
<td>56.4</td>
<td>57.8</td>
<td>35.5</td>
<td>43.1</td>
<td>63.2</td>
<td>48.4</td>
<td>48.5</td>
<td>71.1</td>
<td>51.0</td>
</tr>
<tr>
<td>Ours(Entro)</td>
<td><strong>40.3</strong></td>
<td><strong>51.6</strong></td>
<td><strong>61.5</strong></td>
<td><strong>37.9</strong></td>
<td><strong>58.0</strong></td>
<td><strong>58.6</strong></td>
<td><strong>33.6</strong></td>
<td><strong>45.9</strong></td>
<td><strong>61.8</strong></td>
<td><strong>50.1</strong></td>
<td><strong>50.9</strong></td>
<td><strong>71.7</strong></td>
<td><strong>51.8</strong></td>
</tr>
</tbody>
</table>

Table 4: Accuracy (%) on the Office31 dataset with 31→25 and 25→25(+6) adaptation tasks. As we can see, DANN adaptation effectively reduces target prediction uncertainty (source uncertainty is assured to be low) and improves target test accuracy. Our model further significantly reduces the discrepancy between source and target prediction uncertainty. The nearly synchronous increase of target accuracy and decrease of cross-domain prediction uncertainty discrepancy further indicates that uncertainty matching alleviates domain-shift and improves domain adaptation.

5 Conclusions

We have proposed a novel and effective approach for joint-distribution matching by exploiting prediction uncertainty. To achieve this, we adopt a Bayesian neural network to model prediction uncertainty. Unlike most of existing deep domain-adaptation methods that only reduce marginal feature-distribution shift, the proposed method additionally alleviates conditional distribution shift lingering in the last classifier. Experimental results verify the advantages of the proposed method over state-of-the-art unsupervised domain-adaptation approaches. More interestingly, we also have shown that the proposed method can effectively alleviate negative transfer in domain adaptation.

Acknowledgments

This work is supported by the Zhejiang Provincial Natural Science Foundation (LR19F020005), National Natural Science Foundation of China (61572433, 31471063, 31671074) and thanks for a gift grant from Baidu inc. Also partially supported by the Fundamental Research Funds for the Central Universities.
References


