Conditional Adversarial Networks for Multi-Domain Text Classification

Yuan Wu [1], Diana Inkpen [2] and Ahmed El-Roby [1]
[1]Carleton University, Ottawa, Ontario, Canada
[2]University of Ottawa, Ottawa, Ontario, Canada
{yuan.wu3, Ahmed.ElRoby}@carleton.ca   Diana.Inkpen@uottawa.ca

Abstract

In this paper, we propose conditional adversarial networks (CANs), a framework that explores the relationship between the shared features and the label predictions to impose stronger discriminability to the learned features, for multi-domain text classification (MDTC). The proposed CAN introduces a conditional domain discriminator to model the domain variance in both the shared feature representations and the class-aware information simultaneously, and adopts entropy conditioning to guarantee the transferability of the shared features. We provide theoretical analysis for the CAN framework, showing that CAN's objective is equivalent to minimizing the total divergence among multiple joint distributions of shared features and label predictions. Therefore, CAN is a theoretically sound adversarial network that discriminates over multiple distributions. Evaluation results on two MDTC benchmarks show that CAN outperforms prior methods. Further experiments demonstrate that CAN has a good ability to generalize learned knowledge to unseen domains.

1 Introduction

Text classification is a fundamental task in Natural Language Processing (NLP) and has received constant attention due to its wide applications, ranging from spam detection to social media analytics (Pang et al., 2002; Hu and Liu, 2004; Choi and Cardie, 2008; Socher et al., 2012; Vo and Zhang, 2015). Over the past couple of decades, supervised machine learning methods have shown dominant performance for text classification, such as Naive Bayes Classifiers (Troussas et al., 2013), Support Vector Machines (Li et al., 2018) and Neural Networks (Wu et al., 2020). In particular, with the advent of deep learning, neural network-based text classification models have gained impressive achievements. However, text classification is well known to be highly domain-dependent, the same word could convey different sentiment polarities in different domains (Glorot et al., 2011). For example, the word *infantile* expresses neutral sentiment in baby product review (e.g., *The infantile cart is easy to use*), while in book review, it often indicates a negative polarity (e.g., *This book is infantile and boring*). Thus a text classifier trained on one domain is likely to make spurious predictions on another domain whose distribution is different from the training data distribution. In addition, it is always difficult to collect sufficient labeled data for all interested domains. Therefore, it is of great significance to explore how to leverage available resources from related domains to improve the classification accuracy on the target domain.

The major line of approaches to tackle the above problem is multi-domain text classification (MDTC) (Li and Zong, 2008), which can handle the scenario where labeled data exist for multiple domains, but in insufficient amounts to training an effective classifier. Deep learning models have yielded impressive performance in MDTC (Wu and Guo, 2020; Wu et al., 2021). Most recent MDTC methods adopt the shared-private paradigm, which divides the latent space into two types: one is the shared feature space for all domains with the aim of capturing domain-invariant knowledge, the other one is the private feature space for each domain which extracts domain-specific knowledge. To explicitly ensure the optimum separations among the shared latent space and multiple domain-specific feature spaces, the adversarial training (Goodfellow et al., 2014) is introduced in MDTC. By employing the adversarial training, the domain-specific features can be prevented from creeping into the shared latent space, which will lead to feature redundancy (Liu et al., 2017). In adversarial training, a multinomial domain discriminator is trained
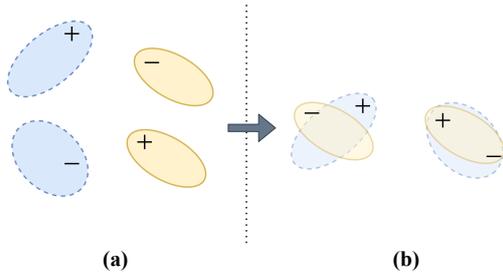
Figure 1: The mismatch risk when aligning the marginal distributions in MDTC, we present the case containing two domains D_1 and D_2. The blue regions denote distributions of D_1, and the yellow regions denote distributions of D_2. (a) The scenario before performing domain alignment. (2) When aligning the marginal distributions, a mismatch may occur with regard to the label.

against a shared feature extractor to minimize the divergences across different domains. When the domain discriminator and the shared feature extractor reach equilibrium, the learned shared features can be regarded as domain-invariant and used for the subsequent classification. The adversarial training-based MDTC approaches yield the state-of-the-art results (Liu et al., 2017; Chen and Cardie, 2018). However, these methods still have a significant limitation: when the data distributions present complex structures, adversarial training may fail to perform global alignment among domains. Such a risk comes from the challenge that in adversarial training, only aligning the marginal distributions can not sufficiently guarantee the discriminability of the learned features. The features with different labels may be aligned, as shown in Figure 1. The critical mismatch can lead to weak discriminability of the learned features.

In this paper, motivated by the conditional generative adversarial networks (CGANs), which aligns distributions of real and generated images via conditioning the generator and discriminator on extra information (Mirza and Osindero, 2014), we propose conditional adversarial networks (CANs) to address the aforementioned challenge. The CAN method introduces a conditional domain discriminator that models domain variance in both shared features and label predictions, exploring the relationship between shared feature representations and class-aware information conveyed by label predictions to encourage the shared feature extractor to capture more discriminative information. Moreover, we use entropy conditioning to avoid the risk of conditioning on the class-aware informa-

tion with low certainty. The entropy conditioning strategy can give higher priority to easy-to-transfer instances. We also provide a theoretical analysis demonstrating the validity of CANs. Our approach adopts the shared-private paradigm. We validate the effectiveness of CAN on two MDTC benchmarks. It can be noted that CAN outperforms the state-of-the-art methods for both datasets. Finally, we empirically illustrate that CAN has the ability to generalize in cases where no labeled data exist for a subset of domains. The contributions of our work are listed as follows:

- We propose conditional adversarial networks (CANs) for multi-domain text classification which incorporate conditional domain discriminator and entropy conditioning to perform alignment on the joint distributions of shared features and label predictions to improve the system performance.

- We present the theoretical analysis of the CAN framework, demonstrating that CANs are minimizers of divergences among multiple joint distributions of shared features and label predictions, and providing the condition where the conditional domain discriminator reaches its optimum.

- We evaluate the effectiveness of CAN on two MDTC benchmarks. The experimental results show that CAN yields state-of-the-art results. Moreover, further experiments on unsupervised multi-source domain adaptation demonstrate that CAN has a good capacity to generalize to unseen domains.

2 Related Work

Multi-domain text classification (MDTC) was first proposed by (Li and Zong, 2008), aiming to simultaneously leverage all existing resources across different domains to improve system performance. Currently, there are two main streams for MDTC: one strand exploits covariance matrix to model the relationship across domains (Dredze and Crammer, 2008; Saha et al., 2011; Zhang and Yeung, 2012); the other strand is based on neural networks, sharing the first several layers for each domain to extract low-level features and generating outputs with domain-specific parameters. The multi-task convolutional neural network (MT-CNN) utilizes a convolutional layer in which only the lookup table is

shared for better word embeddings (Collobert and Weston, 2008). The collaborative multi-domain sentiment classification (CMSC) combines a classifier that learns common knowledge among domains with a set of classifiers, one per domain, each of which captures domain-dependent features to make the final predictions (Wu and Huang, 2015). The multi-task deep neural network (MT-DNN) maps arbitrary text queries and documents into semantic vector representations in a low dimensional latent space and combines tasks as disparate as operations necessary for classification (Liu et al., 2015).

Pioneered by the generative adversarial network (GAN) (Goodfellow et al., 2014), adversarial learning has been firstly proposed for image generation. (Ganin et al., 2016) applies adversarial learning in domain adaptation to extract domain-invariant features across two different distributions (binary adversarial learning). (Zhao et al., 2017) extends it to multiple adversarial learning, enabling the model to learn domain-invariant representations across multiple domains. However, only considering domain-invariant features can not provide optimal solutions for MDTC, because domain-specific information also plays an important role in training an effective classifier. (Bousmalis et al., 2016) proposes the shared-private paradigm to combine domain-invariant features with domain-specific ones to perform classification, illustrating that this scheme can improve system performance. To date, many state-of-the-art MDTC models adopt the adversarial learning and the shared-private paradigm. The adversarial multi-task learning for text classification (ASP-MTL) utilizes long short-term memory (LSTM) without attention as feature extractors and introduces orthogonality constraints to encourage the shared and private feature extractors to encode different aspects of the inputs (Liu et al., 2017). The multinomial adversarial network (MAN) exploits two forms of loss functions to train the domain discriminator: the least square loss (MAN-L2) and negative log-likelihood loss (MAN-NLL) (Chen and Cardie, 2018). The multi-task learning with bidirectional language models for text classification (MT-BL) introduces language modeling as an auxiliary task to encourage the domain-specific feature extractors to capture more syntactic and semantic information, and a uniform label distribution-based loss constraint to the shared feature extractor to enhance the ability to learn domain-invariant features (Yang and Shang, 2019).

Adversarial learning has several advantages, such as Markov chains are not needed and no inference is required during learning (Mirza and Osindero, 2014). However, there still exists an issue in adversarial learning. When data distributions embody complex structures, adversarial learning can fail in performing the global alignment. The conditional generative adversarial network (CGAN) is proposed to address this problem (Mirza and Osindero, 2014). In CGAN, both the generator and discriminator are conditioned on some extra information, such as labels or data from other modalities, to yield better results. The conditional adversarial training mechanism has been explored in transfer learning. The conditional domain adversarial networks (CDANs) condition the domain discriminator on a multilinear map of feature representations and category predictions so as to enable discriminative alignment of multi-mode structures (Long et al., 2018). The conditional generative adversarial networks for structured domain adaptation learn a conditional generator to transform the feature maps of source domain images as if they were extracted from the target domain, and a discriminator to encourage realistic transformations for the semantic segmentation of urban scenes (Hong et al., 2018). Sharing some spirit of CGAN, this paper extends conditional adversarial learning in MDTC, enabling a domain discriminator on the shared features while conditioning it on the class-aware information conveyed by the label predictions. Moreover, in order to guarantee the generalizability of the learned features, we also utilize the entropy conditioning strategy.

3   Approach

In this paper, we consider MDTC tasks in the following setting. Assume there exist M domains $\{D_i\}_{i=1}^{M}$. For each domain, both labeled and unlabeled samples are taken into consideration. Specifically, D_i contains two parts: a limited amount of labeled samples $\mathbb{L}_i = \{(\mathbf{x}_j, y_j)\}_{j=1}^{l_i}$; and a large amount of unlabeled samples $\mathbb{U}_i = \{\mathbf{x}_j\}_{j=1}^{u_i}$. The challenge of MDTC lies in how to improve the system performance of mapping the input $\mathbf{x}$ to its corresponding label y by leveraging all available resources across different domains. The performance is measured as the average classification accuracy across the M domains.
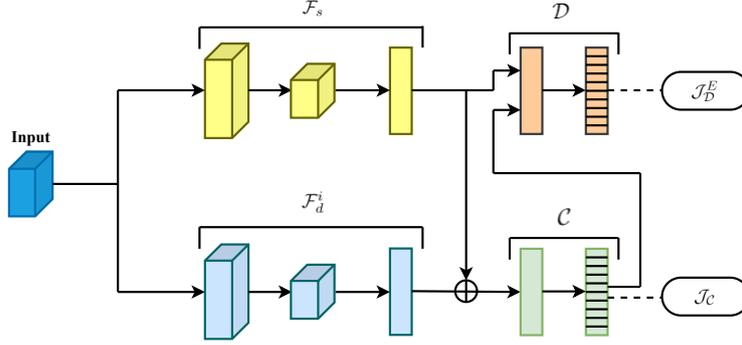
Figure 2: The architecture of the CAN model. A shared feature extractor $\mathcal{F}_s$ learns to capture domain-invariant features; each domain-specific feature extractor $\mathcal{F}_d^i$ learns to capture domain-dependent features; a conditional domain discriminator $\mathcal{D}$ models shared feature distributions by conditioning on discriminative information provided by label predictions; a classifier $\mathcal{C}$ is used to conduct text classification; $\mathcal{J}_\mathcal{C}$ is the classification loss function; $\mathcal{J}_\mathcal{D}^E$ is the entropy conditioning adversarial loss function which guides the domain-invariant feature extraction.

3.1 Model Architecture

We propose conditional adversarial networks (CANs), as shown in Figure 2, which adopt the shared-private scheme and consist of four components: a shared feature extractor $\mathcal{F}_s$, a set of domain-specific feature extractors $\{\mathcal{F}_d^i\}_{i=1}^M$, a conditional domain discriminator $\mathcal{D}$, and a text classifier $\mathcal{C}$. The shared feature extractor $\mathcal{F}_s$ learns to capture domain-invariant features that are beneficial to classification across all domains, while each domain-specific feature extractor $\mathcal{F}_d^i$ aims to learn knowledge that is unique to its own domain. The architecture of these feature extractors are flexible and can be decided based on the practical task. For instance, it can adopt the form of a convolutional neural network (CNN), a recurrent neural network (RNN), or a multi-layer perceptron (MLP). Here, a feature extractor generates vectors with a fixed length, which is considered as the hidden representation of certain input. The classifier $\mathcal{C}$ takes the concatenation of a shared feature and a domain-specific feature as its input and outputs label probabilities. The conditional domain discriminator $\mathcal{D}$ takes the concatenation of a shared feature and the prediction of the given instance provided by $\mathcal{C}$ as its input and predicts the likelihood of that instance coming from each domain.

3.2 Conditional Adversarial Training

Adversarial learning has been successfully investigated in minimizing divergences among domains (Chen and Cardie, 2018; Zhao et al., 2017). In standard adversarial learning for MDTC, a two-player mini-max game is conducted between a domain discriminator and a shared feature extractor:

the domain discriminator is trained to distinguish features across different domains, and the shared feature extractor aims to deceive the discriminator. By performing this mini-max optimization, the domain-invariant features can be learned. The error function of the domain discriminator corresponds well to the divergences among domains. Most MDTC methods align the marginal distributions. However, the transferability with representations transition from general to specific along deep networks is decreasing significantly (Yosinski et al., 2014), only adapting the marginal distributions is not sufficient to guarantee the global alignment. In addition, when the data distributions embody complex structures, which is a real scenario for NLP applications, there is a high risk of failure by matching features with different labels.

Recent advances in the conditional generative adversarial network (CGAN) disclose that better alignment on two different distributions can be obtained by conditioning the generator and discriminator on class-aware information (Mirza and Osindero, 2014). The discriminative information provided by the label prediction potentially reveals the structure information underlying the data distribution. Thus, conditional adversarial learning can better model the divergences among domains on shared feature representations and label predictions. Unlike the prior works that adapting the marginal distributions (Liu et al., 2017; Chen and Cardie, 2018), our proposed CAN framework is formalized on aligning joint distributions of shared features and label predictions. There exist two training flows in our model. Due to the nature of adversarial learning, the conditional domain dis-

criminator is updated with a separate optimizer, while the other components of CAN are trained with the main optimizer. These two flows are supposed to complement each other. Denote $\mathcal{L}_{\mathcal{C}}$ and $\mathcal{L}_{\mathcal{D}}$ as the loss functions of the classifier $\mathcal{C}$ and the conditional domain discriminator $\mathcal{D}$, respectively. We utilize the negative log-likelihood (NLL) loss to encode these two loss functions:

$$\mathcal{L}_{\mathcal{C}}(\widetilde{y}, y) = -\log P(\widetilde{y} = y) \tag{1}$$

$$\mathcal{L}_{\mathcal{D}}(\widetilde{d}, d) = -\log P(\widetilde{d} = d) \tag{2}$$

where y is the true label, $\widetilde{y}$ is the label prediction, d is the domain index and $\widetilde{d}$ is the domain label prediction. Therefore, we formulate CAN as a mini-max optimization problem with two competitive terms defined as follows:

$$\mathcal{J}_{\mathcal{C}} = \sum_{i=1}^{M} \mathbb{E}_{(\mathbf{x}, y) \sim \mathbb{L}_i}[\mathcal{L}_{\mathcal{C}}(\mathcal{C}_i, y)] \tag{3}$$

$$\mathcal{J}_{\mathcal{D}} = \sum_{i=1}^{M} \mathbb{E}_{\mathbf{x} \sim \mathbb{L}_i \cup \mathbb{U}_i}[\mathcal{L}_{\mathcal{D}}(\mathcal{D}([\mathcal{F}_s(\mathbf{x}), \mathcal{C}_i]), d)] \tag{4}$$

where $[\cdot, \cdot]$ is the concatenation of two vectors, $\mathcal{C}_i = \mathcal{C}([\mathcal{F}_s(\mathbf{x}), \mathcal{F}_d^i(\mathbf{x})])$ is the prediction probability of the given instance $\mathbf{x}$. $\mathcal{C}$ and $\mathcal{D}$ adopt MLPs with a softmax layer on top. For the domain-specific feature extractors $\{\mathcal{F}_d^i\}_{i=1}^{M}$, the training is straightforward, as their objective is simple: help $\mathcal{C}$ perform better classification. While the shared feature extractor $\mathcal{F}_s$ has two goals: (1) help $\mathcal{C}$ reduce prediction errors, and (2) confuse $\mathcal{D}$ to reach equilibrium.

3.3 Entropy Conditioning

We condition the domain discriminator $\mathcal{D}$ on the joint variable $(f, c) = (\mathcal{F}_s(\mathbf{x}), \mathcal{C}_i)$. For brevity, here we use f and c to denote $\mathcal{F}_s(\mathbf{x})$ and $\mathcal{C}_i$, respectively. If we enforce different instances to have equal importance, the hard-to-transfer instances with uncertain predictions may deteriorate the system performance (Saito et al., 2019). In order to alleviate the harmful effects introduced by the hard-to-transfer instances, we introduce the entropy criterion $E(c) = -\sum_{k=1}^{2}[c_k log c_k]$ to quantify the uncertainty of label predictions, where c_k is the

probability of predicting an instance to category k (negative: k = 1, positive: k = 2). By using the entropy conditioning, the easy-to-transfer instances with certain predictions are given higher priority. We reweigh these instances by an entropy-aware term: $w(c) = 1 + e^{-E(c)}$. Therefore, the improved $\mathcal{J}_{\mathcal{D}}$ is defined as:

$$\mathcal{J}_{\mathcal{D}}^{E} = \sum_{i=1}^{M} \mathbb{E}_{\mathbf{x} \sim \mathbb{L}_i \cup \mathbb{U}_i}[w(c)\mathcal{L}_{\mathcal{D}}(\mathcal{D}([\mathcal{F}_s(\mathbf{x}), \mathcal{C}_i]), d)] \tag{5}$$

Therefore, the mini-max game of CAN is formulated as:

$$\min_{\mathcal{F}_s, \{\mathcal{F}_d^i\}_{i=1}^{M}, C} \max_{\mathcal{D}} \quad \mathcal{J}_{\mathcal{C}} + \lambda \mathcal{J}_{\mathcal{D}}^{E} \tag{6}$$

where λ is a hyperparameter balancing the two objectives. The entropy conditioning empowers the entropy minimization principle (Grandvalet and Bengio, 2005) and controls the certainty of the predictions, enabling CAN have the ability to generalize on unseen domains with no labeled data. The CAN training is illustrated in Algorithm 1.

Algorithm 1 Stochastic gradient descent training algorithm

1: **Input:** labeled data $\mathbb{L}_i$ and unlabeled data $\mathbb{U}_i$ in M domains; a hyperparameter λ.
2: **for** number of training iterations **do**
3:     Sample labeled mini-batches from the multiple domains $B^{\ell} = \{B_1^{\ell}, \cdots, B_M^{\ell}\}$.
4:     Sample unlabeled mini-batches from the multiple domains $B^u = \{B_1^u, \cdots, B_M^u\}$.
5:     Calculate $loss = \mathcal{J}_{\mathcal{C}} + \lambda \mathcal{J}_{\mathcal{D}}^{E}$ on B^{ℓ} and B^u; Update $\mathcal{F}_s, \{\mathcal{F}_d^i\}_{i=1}^{M}, \mathcal{C}$ by descending along the gradients $\Delta loss$.
6:     Calculate $l_D = \mathcal{J}_{\mathcal{D}}^{E}$ on B^{ℓ} and B^u; Update $\mathcal{D}$ by ascending along the gradients Δl_D.
7: **end for**

3.4 Theoretical Analysis

In this section, we present an analysis showing the validity of the CAN approach for MDTC. All proofs are given in the Appendix. The objective of CAN is equivalent to minimizing the total divergence among the M joint distributions. First, we define different joint distributions as $P_i(f, c) \triangleq$

$P(f = \mathcal{F}_s(\mathbf{x}), c = \mathcal{C}_i | \mathbf{x} \in D_i)$. Combining $\mathcal{L}_{\mathcal{D}}$ with $\mathcal{J}_{\mathcal{D}}$, the objective of $\mathcal{D}$ can be written as:

$$\mathcal{J}_{\mathcal{D}} = -\sum_{i=1}^{M} \mathbb{E}_{(f,c) \sim P_i}[\log \mathcal{D}_i([f,c])] \quad (7)$$

where $\mathcal{D}_i([f,c])$ yields the probability of the vector $([f,c])$ coming from the i-th domain. We first derive that CAN could achieve its optimum if and only if all M joint distributions are identical.

Lemma 1. *For any given $\mathcal{F}_s$, $\{\mathcal{F}_d^i\}_{i=1}^{M}$ and $\mathcal{C}$, the optimum conditional domain discriminator $\mathcal{D}^*$ is:*

$$\mathcal{D}_i^*([f,c]) = \frac{P_i(f,c)}{\sum_{j=1}^{M} P_j(f,c)} \quad (8)$$

Then we provide the main theorem for the CAN framework:

Theorem 1. *Let $\widetilde{P}(f,c) = \frac{\sum_{i=1}^{M} P_i(f,c)}{M}$, when $\mathcal{D}$ is trained to its optimum $\mathcal{D}^*$, we have:*

$$\mathcal{J}_{\mathcal{D}^*} = M \log M - \sum_{i=1}^{M} KL(P_i(f,c) || \widetilde{P}(f,c)) \quad (9)$$

where $KL(\cdot)$ is the Kullback-Leibler (KL) divergence (Aslam and Pavlu, 2007) of each joint distribution $P_i(f,c)$ to the centroid $\widetilde{P}(f,c)$.

Finally, considering the non-negativity and convexity of the KL-divergence (Brillouin, 2013), we have:

Corollary 1. *When $\mathcal{D}$ is trained to its optimum $\mathcal{D}^*$, $\mathcal{J}_{\mathcal{D}^*}$ is $M \log M$. The optimum can be obtained if and only if $P_1(f,c) = P_2(f,c) = ... = P_M(f,c) = \widetilde{P}(f,c)$.*

Therefore, by using conditional adversarial training, we can train the conditional domain discriminator on the joint variable (f,c) to minimize the total divergence across different domains, yielding promising performance on MDTC tasks.

4 Experiments

We evaluate the effectiveness of the CAN model on both MDTC and unsupervised multi-source domain adaptation tasks. The former refers to the setting where the test data falls into one of the M domains, and the latter refers to the setting where the test data comes from an unseen domain without labels. Moreover, an ablation study is provided for further analysis of the CAN model.

Domain	Train	Dev.	Test	Unlabeled	Avg. L	Vocab.
Books	1400	200	400	2000	159	62K
Electronics	1398	200	400	2000	101	30K
DVD	1400	200	400	2000	173	69K
Kitchen	1400	200	400	2000	89	28K
Apparel	1400	200	400	2000	57	21K
Camera	1397	200	400	2000	130	26K
Health	1400	200	400	2000	81	26K
Music	1400	200	400	2000	136	60K
Toys	1400	200	400	2000	90	28K
Video	1400	200	400	2000	156	57K
Baby	1300	200	400	2000	104	26K
Magazine	1370	200	400	2000	117	30K
Software	1315	200	400	475	129	26K
Sports	1400	200	400	2000	94	30K
IMDB	1400	200	400	2000	269	44K
MR	1400	200	400	2000	21	12K

Table 1: Statistics of the FDU-MTL dataset

4.1 Experimental Settings

Dataset We conduct experiments on two MDTC benchmarks: the Amazon review dataset (Blitzer et al., 2007) and the FDU-MTL dataset (Liu et al., 2017). The Amazon review dataset consists of four domains: books, DVDs, electronics, and kitchen. For each domain, there exist 2,000 instances: 1,000 positive ones and 1,000 negative ones. All data was pre-processed into a bag of features (unigrams and bigrams), losing all word order information. In our experiments, the 5,000 most frequent features are used, representing each review as a 5,000-dimensional vector. The FDU-MTL dataset is a more complicated dataset, which contains 16 domains: books, electronics, DVDs, kitchen, apparel, camera, health, music, toys, video, baby, magazine, software, sport, IMDB, and MR. All data in the FDU-MTL dataset are raw text data, tokenized by the Stanford tokenizer. The detailed statistics of the FDU-MTL dataset are listed in Table 1.

Implementation Details All experiments are implemented by using PyTorch. The CAN has one hyperparameter: λ, which is fixed as 1 in all experiments, the parameter sensitivity analysis is presented in the Appendix. We use Adam optimizer (Kingma and Ba, 2014), with the learning rate 0.0001, for training. The batch size is 8. We adopt the same model architecture as in (Chen and Cardie, 2018). For the Amazon Review dataset, MLPs are used as feature extractors, with an input size of 5,000. Each feature extractor is composed of two hidden layers, with size 1,000 and 500, respectively. The output size of the shared feature extractor is 128 while 64 for the domain-specific ones. The dropout rate is 0.4 for each component. Classifier

Domain	CMSC-LS	CMSC-SVM	CMSC-Log	MAN-L2	MAN-NLL	CAN(Proposed)
Books	82.10	82.26	81.81	82.46	82.98	**83.76 ± 0.20**
DVD	82.40	83.48	83.73	83.98	84.03	**84.68 ± 0.16**
Electr.	86.12	86.76	86.67	87.22	87.06	**88.34 ± 0.14**
Kit.	87.56	88.20	88.23	88.53	88.57	**90.03 ± 0.19**
AVG	84.55	85.18	85.11	85.55	85.66	**86.70 ± 0.11**

Table 2: MDTC classification accuracies on the Amazon review dataset.

Domain	MT-CNN	MT-DNN	ASP-MTL	MAN-L2	MAN-NLL	MT-BL	CAN(Proposed)
books	84.5	82.2	84.0	87.6	86.8	**89.0**	87.8 ± 0.2
electronics	83.2	81.7	86.8	87.4	88.8	90.2	**91.6 ± 0.5**
dvd	84.0	84.2	85.5	88.1	88.6	88.0	**89.5 ± 0.4**
kitchen	83.2	80.7	86.2	89.8	89.9	90.5	**90.8 ± 0.3**
apparel	83.7	85.0	87.0	**87.6**	**87.6**	87.2	87.0 ± 0.7
camera	86.0	86.2	89.2	91.4	90.7	89.5	**93.5 ± 0.1**
health	87.2	85.7	88.2	89.8	89.4	**92.5**	90.4 ± 0.6
music	83.7	84.7	82.5	85.9	85.5	86.0	**86.9 ± 0.1**
toys	89.2	87.7	88.0	90.0	90.4	**92.0**	90.0 ± 0.3
video	81.5	85.0	84.5	89.5	**89.6**	88.0	88.8 ± 0.4
baby	87.7	88.0	88.2	90.0	90.2	88.7	**92.0 ± 0.2**
magazine	87.7	89.5	92.2	92.5	92.9	92.5	**94.5 ± 0.5**
software	86.5	85.7	87.2	90.4	90.9	**91.7**	90.9 ± 0.2
sports	84.0	83.2	85.7	89.0	89.0	89.5	**91.2 ± 0.7**
IMDb	86.2	83.2	85.5	86.6	87.0	88.0	**88.5 ± 0.6**
MR	74.5	75.5	76.7	76.1	76.7	75.7	**77.1 ± 0.9**
AVG	84.5	84.3	86.1	88.2	88.4	88.6	**89.4 ± 0.1**

Table 3: MDTC classification accuracies on the FDU-MTL dataset.

and discriminator are MLPs with one hidden layer of the same size as their input ($128 + 64$ for classifier and $128+2$ for discriminator). ReLU is used as the activation function. For the FDU-MTL dataset, CNN with a single convolutional layer is used as the feature extractor. It uses different kernel sizes $(3, 4, 5)$, and the number of kernels is 200. The input of the convolutional layer is a 100-dimensional vector, obtained by using word2vec (Mikolov et al., 2013), for each word in the input sequence.

4.2 Multi-Domain Text Classification

Comparison Methods We first conduct experiments of multi-domain text classification. The CAN model is compared with a number of state-of-the-art methods, which are listed below:

- MT-CNN: A CNN-based model which shares the lookup table across domains for better word embeddings (Collobert and Weston, 2008).

- MT-DNN: The multi-task deep neural network model with bag-of-words input and MLPs, in which a hidden layer is shared (Liu et al., 2015).

- CMSC-LS, CMSC-SVM, CMSC-Log: The collaborative multi-domain sentiment clas-

sification method combines an overall classifier across domains and a set of domain-dependent classifiers to make the final prediction. The models are trained on least square loss, hinge loss, and log loss, respectively (Wu and Huang, 2015).

- ASP-MTL: The adversarial multi-task learning framework of text classification, which adopts the share-private scheme, adversarial learning, and orthogonality constraints (Liu et al., 2017).

- MAN-L2, MAN-NLL: The multinomial adversarial network for multi-domain text classification (Chen and Cardie, 2018). This model uses two forms of loss functions to train the domain discriminator: least square loss and negative log-likelihood loss.

- MT-BL: The multi-task learning with bidirectional language models for text classification, which adds language modeling and a uniform label distribution-based loss constraint to the domain-specific feature extractors and the shared feature extractor, respectively (Yang and Shang, 2019).

All the comparison methods use the standard partitions of the datasets. Thus, we directly cite

Method	Books	DVD	Electr.	Kit.	AVG
CAN (full)	83.76	84.68	88.34	90.03	86.70
CAN w/o C	82.45	84.45	87.30	89.65	85.96
CAN w/o E	83.60	84.80	87.70	89.40	86.38
CAN w/o CE	82.98	84.03	87.06	88.57	85.66

Table 4: Ablation study analysis on the Amazon review dataset.

the results from (Chen and Cardie, 2018; Yang and Shang, 2019) for fair comparisons.

Results We conduct MDTC experiments following the setting of (Chen and Cardie, 2018): A 5-fold cross-validation is implemented for the Amazon review dataset. All data is divided into 5 folds per domain: three of the five folds are used as the training set, one is the validation set, and the remaining one is treated as the test set. The 5-fold average test accuracy is reported. All reports are based on 5 runs.

Table 2 and Table 3 show the experimental results on the Amazon review dataset and the FDU-MTL dataset, respectively. From Table 2, it can be seen that our model yields state-of-the-art results not only for the average classification accuracy, but also on each individual domain.

From the experimental results on the FDU-MTL dataset, reported in Table 3, we can see that the CAN model obtains the best accuracies on 10 of 16 domains and achieves the best result in terms of the average classification accuracy. The experimental results on these two MDTC benchmarks illustrate the efficacy of our model.

Ablation Study The CAN model adopts the conditional domain discriminator and entropy conditioning. In this section, we investigate how these two strategies impact the performance of our model on the Amazon review dataset. In particular, three ablation variants are evaluated: (1) CAN w/o C, the variant of the proposed CAN model without conditioning the domain discriminator on label predictions, which utilizes the standard domain discriminator and entropy conditioning; (2) CAN w/o E, the variant of the proposed CAN model without the entropy conditioning, which hence imposes equal importance to different instances; (3) CAN w/o CE, the variant of the proposed CAN model which only uses standard adversarial training for domain alignment. The results of the ablation study are shown in Table 4, where we can see that all variants produce inferior results. Thus, it indicates that both strategies contribute to the CAN model.

4.3 Unsupervised Multi-Source Domain Adaptation

In the MDTC scenario, the model requires labeled training data from each domain. However, in real-world applications, many domains may have no labeled data at all. Therefore, it is important to evaluate the performance of MDTC models on unseen domains (Wright and Augenstein, 2020).

In the unsupervised multi-source domain adaptation setting, we have multiple source domains with both labeled and unlabeled data and one target domain with only unlabeled data. The CAN has the ability to learn domain-invariant representations on unlabeled data, and thus it can be generalized to unseen domains. Since the target domain has no labeled data at all, the domain discriminator is updated only on unlabeled data in this setting. When conducting text classification on the target domain, we only feed the shared feature to $\mathcal{C}$ and set the domain-specific feature vector to 0.

We conduct the experiments on the Amazon review dataset. In the experiments, three of the four domains are regarded as the source domains, and the remaining one is used as the target one. The evaluations are conducted on the target domain. In order to validate CAN's effectiveness, we compare CAN with several domain-agnostic methods: (1) the MLP model; (2) the marginalized denoising autoencoder (mSDA) (Chen et al., 2012); (3) the domain adversarial neural network (DANN) (Ganin et al., 2016). These methods ignore the differences among domains. And certain state-of-the-art unsupervised multi-source domain adaptation methods: (4) the multi-source domain adaptation neural network (MDAN(H) and MDAN(S)) (Zhao et al., 2017); (5) the multinomial adversarial network (MAN-L2 and MAN-NLL) (Chen and Cardie, 2018). When training the domain-agnostic methods, the data in the multiple source domains are combined together as a single source domain.

In Table 5, we observe that the CAN model outperforms all the comparison methods on three out of four domains. In terms of the average classification accuracy, the CAN method achieves superior performance. This suggests that our model has a good ability to generalize on unseen domains.

5 Conclusion

In this paper, we propose conditional adversarial networks (CANs) for MDTC. This approach can perform alignment on joint distributions of shared

Domain	MLP	mSDA	DANN	MDAN(H)	MDAN(S)	MAN-L2	MAN-NLL	CAN(Proposed)
Books	76.55	76.98	77.89	78.45	78.63	78.45	77.78	**78.91**
DVD	75.88	78.61	78.86	77.97	80.65	81.57	82.74	**83.37**
Elec.	84.60	81.98	84.91	84.83	**85.34**	83.37	83.75	84.76
Kit.	85.45	84.26	86.39	85.80	86.26	85.57	86.41	**86.75**
AVG	80.46	80.46	82.01	81.76	82.72	82.24	82.67	**83.45**

Table 5: Unsupervised multi-source domain adaptation results on the Amazon review dataset.

features and label predictions to improve the system performance. The CAN approach adopts the shared-private paradigm, trains domain discriminator by conditioning it on discriminative information conveyed by the label predictions to encourage the shared feature extractor to capture more discriminative information, and exploits entropy conditioning to guarantee the transferability of the learned features. Experimental results on two MDTC benchmarks demonstrate that the CAN model can not only boost the average classification accuracy for MDTC but also promote the generalization ability when tackling unseen domains.

References

Javed A Aslam and Virgil Pavlu. 2007. Query hardness estimation using jensen-shannon divergence among multiple scoring functions. In *European conference on information retrieval*, pages 198–209. Springer.

J. Blitzer, M. Dredze, and F. Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proc. of the annual meeting of the association of computational linguistics*.

K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. 2016. Domain separation networks. In *Advances in Neural Information Processing Systems*.

Leon Brillouin. 2013. *Science and information theory*. Courier Corporation.

M. Chen, Z. Xu, K.Q. Weinberger, and F. Sha. 2012. Marginalized denoising autoencoders for domain adaptation. In *Proc. of International Conference on International Conference on Machine Learning*.

X. Chen and C. Cardie. 2018. Multinomial adversarial networks for multi-domain text classification. In *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Yejin Choi and Claire Cardie. 2008. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 793–801.

R. Collobert and J. Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proc. of the inter. conference on Machine learning*.

Mark Dredze and Koby Crammer. 2008. Online methods for multi-domain learning and adaptation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 689–697. Association for Computational Linguistics.

Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

X. Glorot, A. Bordes, and Y. Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proc. of the international conference on machine learning*.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. 2014. Generative adversarial nets. In *Advances in neural information proc. systems*.

Yves Grandvalet and Yoshua Bengio. 2005. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536.

Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. 2018. Conditional generative adversarial network for structured domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1335–1344.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

D.P. Kingma and J. Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lingling Li, Yuan Wu, Yi Yang, Lian Li, and Bin Wu. 2018. A new strategy to detect lung cancer on ct images. In *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, pages 716–722. IEEE.

S. Li and C. Zong. 2008. Multi-domain sentiment classification. In *Proc. of the Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*.

P. Liu, X. Qiu, and X. Huang. 2017. Adversarial multi-task learning for text classification. In *Proc. of the Annual Meeting of the Association for Computational Linguistics.*

X. Liu, J. Gao, X. He, L. Deng, K. Duh, and Y.Y. Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. 2018. Conditional adversarial domain adaptation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 1647–1657.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781.*

Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784.*

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Avishek Saha, Piyush Rai, Hal Daumã, Suresh Venkatasubramanian, et al. 2011. Online learning of multiple tasks and their relationships. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 643–651.

Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. 2019. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8050–8058.

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 1201–1211. Association for Computational Linguistics.

Christos Troussas, Maria Virvou, Kurt Junshean Espinosa, Kevin Llaguno, and Jaime Caro. 2013. Sentiment analysis of facebook statuses using naive bayes classifier for language learning. In *IISA 2013*, pages 1–6. IEEE.

Duy-Tin Vo and Yue Zhang. 2015. Target-dependent twitter sentiment classification with rich automatic features. In *Twenty-Fourth International Joint Conference on Artificial Intelligence.*

Dustin Wright and Isabelle Augenstein. 2020. Transformer based multi-source domain adaptation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7963–7974.

F. Wu and Y. Huang. 2015. Collaborative multi-domain sentiment classification. In *IEEE International Conference on Data Mining.*

Yuan Wu and Yuhong Guo. 2020. Dual adversarial co-learning for multi-domain text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6438–6445.

Yuan Wu, Diana Inkpen, and Ahmed El-Roby. 2020. Dual mixup regularized learning for adversarial domain adaptation. In *European Conference on Computer Vision*, pages 540–555. Springer.

Yuan Wu, Diana Inkpen, and Ahmed El-Roby. 2021. Mixup regularized adversarial networks for multi-domain text classification. *arXiv preprint arXiv:2102.00467.*

Qi Yang and Lin Shang. 2019. Multi-task learning with bidirectional language models for text classification. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328.

Yu Zhang and Dit-Yan Yeung. 2012. A convex formulation for learning task relationships in multi-task learning. *arXiv preprint arXiv:1203.3536.*

H. Zhao, S. Zhang, G. Wu, J.P. Costeira, J. Moura, and G.J. Gordon. 2017. Multiple source domain adaptation with adversarial training of neural networks. *arXiv preprint arXiv:1705.09684.*

A Appendix

A.1 Proofs for CAN

Assume there exist M domains, for each domain $\mathcal{D}_i$, we have a joint distribution defined as:

$$P_i(f, c) \triangleq P(f = \mathcal{F}_s(\mathbf{x}), c = \mathcal{C}_i | \mathbf{x} \in \mathcal{D}_i) \tag{10}$$

where $\mathcal{C}_i = \mathcal{C}([\mathcal{F}_s(\mathbf{x}), \mathcal{F}_d^i(\mathbf{x})])$ is the prediction probability of the given instance $\mathbf{x}$, $[\cdot, \cdot]$ is the concatenation of two vectors. The objective of $\mathcal{D}$ is to minimize $\mathcal{J}_{\mathcal{D}}$:

$$\mathcal{J}_{\mathcal{D}} = -\sum_{i=1}^{M} \mathbb{E}_{(f,c)\sim P_i}[\log \mathcal{D}_i([f, c])] \tag{11}$$

where $\mathcal{D}_i([f,c])$ is the probability of the vector $([f,c])$ coming from the i-th domain. Therefore, we have:

$$\sum_{i=1}^{M} \mathcal{D}_i([f,c]) = 1 \qquad (12)$$

Lemma 1. *For any given $\mathcal{F}_s$, $\{\mathcal{F}_d^i\}_{i=1}^{M}$ and $\mathcal{C}$, the optimum conditional domain discriminator $\mathcal{D}^*$ is:*

$$\mathcal{D}_i^*([f,c]) = \frac{P_i(f,c)}{\sum_{j=1}^{M} P_j(f,c)} \qquad (13)$$

Proof. For any given $\mathcal{F}_s$, $\{\mathcal{F}_d^i\}_{i=1}^{M}$ and $\mathcal{C}$, the optimum

$$\mathcal{D}^* = \arg\min_{\mathcal{D}} \mathcal{J}_{\mathcal{D}}$$

$$= \arg\min_{\mathcal{D}} -\sum_{i=1}^{M} \mathbb{E}_{(f,c)\sim P_i}[\log \mathcal{D}_i([f,c])]$$

$$= \arg\max_{\mathcal{D}} \sum_{i=1}^{M} \int_{(f,c)} P_i(f,c) \log \mathcal{D}_i([f,c]) d(f,c)$$

$$= \arg\max_{\mathcal{D}} \int_{(f,c)} \sum_{i=1}^{M} P_i(f,c) \log \mathcal{D}_i([f,c]) d(f,c)$$

Here, we utilize the Lagrangian Multiplier for $\mathcal{D}^*$ under the condition (12). We have:

$$L(\mathcal{D}_1,...,\mathcal{D}_M,\lambda) = \sum_{i=1}^{M} P_i \log \mathcal{D}_i - \lambda(\sum_{i=1}^{M} \mathcal{D}_i - 1) \qquad (14)$$

Let $\nabla L = 0$, we have:

$$\begin{cases} \nabla_{\mathcal{D}_i} \sum_{j=1}^{M} P_j \log \mathcal{D}_j - \lambda \nabla_{\mathcal{D}_i}(\sum_{j=1}^{M} \mathcal{D}_j - 1) = 0 \\ \sum_{i=1}^{M} \mathcal{D}_i = 1 \end{cases}$$

From the two above equations, we have:

$$\mathcal{D}_i^*(f,c) = \frac{P_i(f,c)}{\sum_{j=1}^{M} P_j(f,c)} \qquad (15)$$

$\square$

Theorem 1. *Let $\widetilde{P}(f,c) = \frac{\sum_{i=1}^{M} P_i(f,c)}{M}$, when $\mathcal{D}$ is trained to its optimum $\mathcal{D}^*$, we have:*

$$\mathcal{J}_{\mathcal{D}^*} = M \log M - \sum_{i=1}^{M} KL(P_i(f,c)||\widetilde{P}(f,c)) \qquad (16)$$

where $KL(\cdot)$ is the Kullback-Leibler (KL) divergence of each joint distribution $P_i(f,c)$ to the centroid $\widetilde{P}(f,c)$

Proof. Let $\widetilde{P}(f,c) = \frac{\sum_{i=1}^{M} P_i(f,c)}{M}$. We have:

$$\sum_{i=1}^{M} KL(P_i(f,c)||\widetilde{P}(f,c)) =$$

$$\sum_{i=1}^{M} \mathbb{E}_{(f,c)\sim P_i}[\log \frac{P_i(f,c)}{\widetilde{P}(f,c)}]$$

When $\mathcal{D}$ is updated to $\mathcal{D}^*$, we have:

$$\mathcal{J}_{\mathcal{D}^*} = -\sum_{i=1}^{M} \mathbb{E}_{(f,c)\sim P_i}[\log \mathcal{D}_i^*([f,c])]$$

$$= -\sum_{i=1}^{M} \mathbb{E}_{(f,c)\sim P_i}[\log \frac{P_i(f,c)}{\sum_{j=1}^{M} P_j(f,c)}]$$

$$= -\sum_{i=1}^{M} \mathbb{E}_{(f,c)\sim P_i}[\log \frac{P_i(f,c)}{\sum_{j=1}^{M} P_j(f,c)} + \log M]$$

$$+ M \log M$$

$$= M \log M - \sum_{i=1}^{M} \mathbb{E}_{(f,c)\sim P_i}[\log \frac{P_i(f,c)}{\frac{\sum_{j=1}^{M} P_j(f,c)}{M}}]$$

$$= M \log M - \sum_{i=1}^{M} \mathbb{E}_{(f,c)\sim P_i}[\log \frac{P_i(f,c)}{\widetilde{P}(f,c)}]$$

$$= M \log M - \sum_{i=1}^{M} KL(P_i(f,c)||\widetilde{P}(f,c))$$

$\square$

In our model, a mini-max game is implemented to achieve the optimum:

$$\min_{\mathcal{F}_s,\{\mathcal{F}_d^i\}_{i=1}^{M},\mathcal{C}} \max_{\mathcal{D}} \quad \mathcal{J}_{\mathcal{C}} + \lambda \mathcal{J}_{\mathcal{D}} \qquad (17)$$

Therefore, by the non-negativity and convexity of the KL-divergence, we can have the corollary:

Corollary 1. *When $\mathcal{D}$ is trained to its optimum $\mathcal{D}^*$, $\mathcal{J}_{\mathcal{D}^*}$ is $M \log M$. The optimum can be obtained if and only if $P_1(f,c) = P_2(f,c) = ... = P_M(f,c) = \widetilde{P}(f,c)$.*
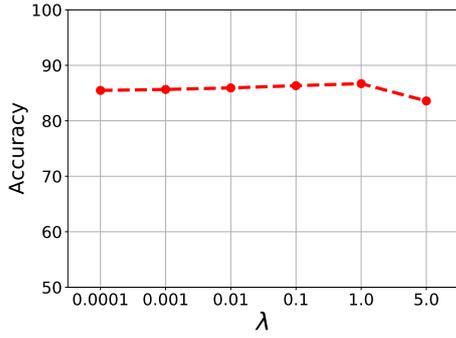
26

Figure 3: The parameter sensitivity analysis.

A.2 Parameter Sensitivity Analysis

The proposed CAN approach has one hyperparameter λ, which is used to balance $\mathcal{J}_{\mathcal{C}}$ and $\mathcal{J}_{\mathcal{D}}^{E}$. We conduct parameter sensitivity analysis on the Amazon review dataset. The λ is evaluated in the range $\{0.0001, 0.001, 0.01, 0.1, 1.0, 5.0\}$. The experimental results are shown in Figure 3. The average classification accuracies across the four domains are reported. It can be noted that from 0.0001 to 1.0, the performance increases with λ increasing, the performance change is very small. Then the accuracy reaches the optimum at the point $\lambda = 1.0$, while the further increase of λ will dramatically deteriorate the performance. This suggests that the selection of λ has an influence on the system performance.

27