Blending-target Domain Adaptation by Adversarial Meta-Adaptation Networks

Ziliang Chen[1], Jingyu Zhuang[1], Xiaodan Liang[1,2], Liang Lin[1,2*]

[1]Sun Yat-sen University   [2]DarkMatter AI Research

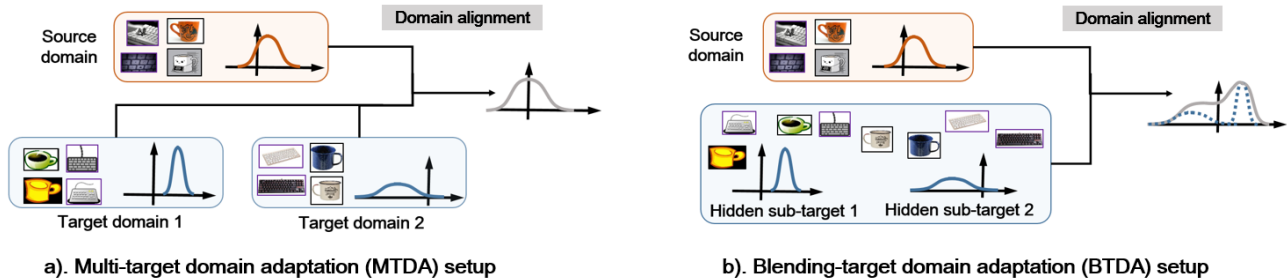c.ziliang@yahoo.com, zhuangjy6@mail2.sysu.edu.cn, xdliang328@gmail.com, linliang@ieee.org

Figure 1. The comparison of MTDA and BTDA (color orange and blue denote source and target) setups. In MTDA (a), target domains are explicitly separated and we are informed by which target an unlabeled sample originates from. In BTDA (b), sub-target IDs are encrypted. If we treat them as a combined single target, transfer learning will lead to the adaptation on a multi-target mixture instead of each hidden target (gray distribution curves in (a), (b)). It implies category-shifted adaptation and negative transfer in practice. Best viewed in color.

Abstract

*(Unsupervised) Domain Adaptation (**DA**) seeks for classifying target instances when solely provided with source labeled and target unlabeled examples for training. Learning domain-invariant features helps to achieve this goal, whereas it underpins unlabeled samples drawn from a single or multiple explicit target domains (Multi-target DA). In this paper, we consider a more realistic transfer scenario: our target domain is comprised of multiple sub-targets implicitly blended with each other, so that learners could not identify which sub-target each unlabeled sample belongs to. This Blending-target Domain Adaptation (**BTDA**) scenario commonly appears in practice and threatens the validities of most existing DA algorithms, due to the presence of domain gaps and categorical misalignments among these hidden sub-targets.*

To reap the transfer performance gains in this new scenario, we propose Adversarial Meta-Adaptation Network (AMEAN). AMEAN entails two adversarial transfer learning processes. The first is a conventional adversarial transfer to bridge our source and mixed target domains. To circumvent the intra-target category misalignment, the second process presents as "learning to adapt": It deploys an unsupervised meta-learner receiving target data and their ongoing feature-learning feedbacks, to discover target clus-ters as our "meta-sub-target" domains. These meta-sub-targets auto-design our meta-sub-target DA loss, which empirically eliminates the implicit category mismatching in our mixed target. We evaluate AMEAN and a variety of DA algorithms in three benchmarks under the BTDA setup. Empirical results show that BTDA is a quite challenging transfer setup for most existing DA algorithms, yet AMEAN significantly outperforms these state-of-the-art baselines and effectively restrains the negative transfer effects in BTDA.

1. Introduction

[1] Despite achieving a growing number of successes, deep supervised learning algorithms remain restrictive in a variety of new application scenarios due to their vulnerabilities towards *domain shifts* [15]: when evaluated on newly-emerged unlabeled target examples drawn from a distribution non-identical with the training source density, supervised learners inevitably perform inferior. In terms of visual data, this issue stems from diverse domain-specific appearance variabilities, *e.g.* differences in background and

camera poses, occlusions and volatile illumination conditions, *etc*. Hence the variabilities are highly relevant in most machine vision implementations and obstacle their advancements. To address these problems, (unsupervised) Domain Adaptations (DAs) choose suitable statistical measures between source and target domains, *e.g.*, maximum mean discrepancy (MMD) [22] and adversarial-network distance [10, 35], to learn domain-invariant features in pursuit of consistent cross-domain model performances. The related studies increasingly attract a large amount of interests from the areas of domain-adaptive perception [18, 7, 6], autonomous steering [43, 42] and robotic vision [5, 3, 30].

Most existing DA approaches indeed have evidenced impressive performances in a laboratory, whereas the domain shifts that frequently occur in reality, are far from being settled through these techniques. One explanation is the ideal target-domain premise these DA techniques start from. Particularly, DAs are conventionally established on a "single-target" preset, namely, all target examples are drawn from an identical distribution. Some recent researches focus on *Multi-target DA* (MTDA) [2, 44], where target examples stem from multiple distributions, whereas we exactly know which target they belong to (See Fig.1.(a)).

In this paper, we argue these target-domain preconditions always taken for granted in most previous transfer learning literatures. After revisiting widespread presences, we discover that target unlabeled examples are often too diverse to well suit a "single-target" foresight. For instance, virtual-to-real researches [5, 43] encourage robots and driver agents trained on a simulation platform to adaptively perform in the real-world environment. However, the target real-world environment includes extensive arrays of scenarios and continuously changes as time goes by. Another case is an encrypted dataset stored in a cloud server [12], where the unlabeled examples are derived from multiple origins whereas due to a privacy protection, users have no access to identify these origins. These facts imply the existence of multiple sub-target domains while unlike standard MTDA, these sub-targets are blended with each other so that learners are not able to identify which sub-target each unlabeled example belongs to (See Fig.1.(b)). This so-called *blending-target domain adaptation* (**BTDA**) scenario regularly occurs in more other circumstances and during adaptation process, it commonly arouses notorious negative transfer effects [29] by two reasons:

- Hidden sub-target domains are organized as a mixture distribution, whereas without the knowledge of sub-target ID, it is quite difficult to align the category to reduce their mismatching across the sub-targets.

- Regardless of the domain gaps among the hidden targets, existing DA approaches will suffer from the category misalignments among these sub-targets.

. Due to the blending-target preset, BTDA can not be solved by existing multi-target transfer learning methods.

To reap transfer performance gains and simultaneously prevent the negative transfer effects in BTDA scenario, we propose *Adversarial Meta-Adaptation Network* (**AMEAN**) attempting to solve this problem under the context of visual recognition. AMEAN evolves from popular adversarial adaptation frameworks [9, 17] and opts for minimizing the discrepancy between our source and mixed target domains. But distinguished from these existing pipelines, our AMEAN is inspired by meta-learning and AutoML [1, 39], which concurrently deploys an *unsupervised meta-learner* to learn deep target cluster embeddings by receiving the mixed target data and their ongoing-learned features as feedbacks. The incurred clusters are treated as meta-sub-target domains. Hence, our AMEAN auto-designs its multi-target adversarial adaptation loss functions and to this end, dynamically train itself to obtain domain-invariant features from a source to a mixed target and among the multiple meta-sub-target domains derived from the mixed target. This bi-level optimization endows more diverse and flexible adaptation within the mixed target and effectively mitigates its latent category mismatching.

Our contributions mainly present in three aspects:

1. On account of practical cross-domain applications, we consider a new transfer scenario termed Blending-target Domain Adaptation (BTDA), which is common in reality and more difficult to settle.

2. We propose AMEAN, a adversarial transfer framework incorporating meta-learner dynamically inducing meta-sub-targets to auto-design adversarial adaptation losses, which effectively achieve transfers in BTDA.

3. Our experiments are conducted on three widely-applied DA benchmarks. Our results show that BTDA setup definitely brings more transfer risks towards existing DA algorithms, while AMEAN significantly outperforms the state-of-the-art and present more robust in BTDA setup.

2. Related Work

Before introducing BTDA problem setup, we would like to briefly revisit (unsupervised) Domain Adaptation (DA) under the modern visual learning background.

Single-source-single-target DA. DAs are derived from [31, 14], where shallow models are deployed to achieve data transfer across visual domains. The development of deep learning enlightens the tunnel to learn nonlinear transferable feature mappings in DAs. Up-to-date deep DA methods have been branched into two mainstreams: explicit and implicit statistical measure matching. The former employs MMD [22, 24], CMD [45], JMMD [23, 25], *etc*, as the domain regularizer to chase for consistent model performances both on source and target datasets. The latter designates domain discriminators to perform adversarial learning, where the feature extractor is trained to optimize the

transferable feature spaces. It includes amounts of avenues to present diverse adversarial manners, *e.g.*, CoGAN [21], DDC [17], RevGred [9, 10], ADDA [35], VADA [34], GTA [33], *etc*. Beside of the two branches, there are some approaches in virtue of other ideologies, *e.g.*, reconstruction [11], semi-supervised learning [32] to optimize a domain-aligned feature space. It is worth noting that, these methods agree with the "single-source-single-target" precondition when they learns transferable features for DA .

Multi-source DA (MSDA). MSDA aims at boosting the target-adaptive accuracy of a model by introducing multiple source domains in a transfer process. It is a historical topic [41] and refers to a part of DA theories [27] [4]. Some recent work place the problem under the deep visual learning background. For instance, [40] invented a adversarial reweighting strategy to infer a source-ensemble target classifier; [46] developed an old-fashion theory to suit deep MSDA and provided a target error upper bound; [26] stacked multi-DA layers in a network to obtain robust multi-source domain alignment.

Multi-target DA (MTDA). Similar to MSDA, the goal of MTDA is to enhance data transfer efficacy by bridging the cross-target semantic. [2] used a semantic disentangler to facilitate an adversarial MTDA approach; [44] considered the visible semantic gap between multiple targets and propose a dictionary learning algorithm to suit this problem. MTDA is still a fresh area and awaits more explorations.

3. Blending-target DA (BTDA): Problem Setup

Preliminaries. Let's consider an m-class visual recognition problem. Suppose a source dataset includes n_s labeled examples $\mathcal{S} = \{(\boldsymbol{x}_i^{(s)}, \boldsymbol{y}_i^{(s)})\}_{i=1}^{n_s}$, where $\boldsymbol{x}_i^{(s)} \in \mathbb{R}_+^d$ denotes the ith source image lying on a d-dimensional data space and $\boldsymbol{y}_i^{(s)}$ is an m-dimensional one-hot vector corresponding to its label. Besides, a target set includes n_t unlabeled examples $\mathcal{T} = \{\boldsymbol{x}_i^{(t)}\}_{i=1}^{n_t}$ where $\boldsymbol{x}_i^{(t)} \in \mathbb{R}_+^d$ denotes the ith target image. $\mathcal{S}$ and $\mathcal{T}$ underly distributions $P_\mathcal{S}(\boldsymbol{x}, \boldsymbol{y})$ and $P_\mathcal{T}(\boldsymbol{x})$, in which $P_\mathcal{T}(\boldsymbol{x}) = \int P_\mathcal{T}(\boldsymbol{x}, \boldsymbol{y})d\boldsymbol{y}$ indicates target labels unobservable during training. DA seeks for learning a classifier along with a domain-invariant feature extractor across $\mathcal{S}$ and $\mathcal{T}$, which is capable to predict the correct labels of given images sampled from $P_\mathcal{T}(\boldsymbol{x}, \boldsymbol{y})$. As of now DA assumes all unlabeled images derived from a single target distribution $P_\mathcal{T}$.

Here we turn to consider the multi-target DA (MTDA) setup: every unlabeled target instance $\boldsymbol{x}^{(t)}$ underly k distributions $\{P_{\mathcal{T}_j}(\boldsymbol{x}^{(t)})\}_{j=1}^k$, as they are drawn from the mixture $P_\mathcal{T}(\boldsymbol{x}^{(t)}) = \sum_{j=1}^k \pi_j P_{\mathcal{T}_j}(\boldsymbol{x}^{(t)})$ where $\forall j \in [k], \pi_j \in [0, 1] \& \sum_j^k \pi_j = 1$. However, *The learning goal of MTDA is to simultaneously adapt k targets $\{\mathcal{T}_j\}_{j=1}^k$ instead of their mixed target $\mathcal{T}$*. Since the multi-target proportions $\{\pi_j\}_{j=1}^k$ are known in MTDA, target set $\mathcal{T}_j$ is explicitly provided by

drawing from the posterior $\frac{\pi_j P_{\mathcal{T}_j}(\boldsymbol{x}^{(t)})}{\sum_{j'=1}^k \pi_j P_{\mathcal{T}_{j'}}(\boldsymbol{x}^{(t)})}$. Hence existing DA algorithms can address the problem by training k target-specific DA models respectively, and using the jth-target model to classify the examples from the jth target.

From MTDA to BTDA. Like MTDA, BTDA is also established on a target mixture distribution and expected to adapt k targets $\{\mathcal{T}_j\}_{j=1}^k$. However, multi-target proportions $\{\pi_j\}_{j=1}^k$ in BTDA are unobservable. In other words, ***BTDA learners are solely provided with a mixed target set $\mathcal{T}$ drawn from a k-mixture density and required to classify a mixed target test set drawn from the same mixture***. If we directly leverage existing DA techniques to transfer category information from $\mathcal{S}$ to $\mathcal{T}$, the learning objective will guide domain-invariant features to adapt a mixed target set $\mathcal{T}$ instead of k target $\{\mathcal{T}_j\}_{j=1}^k$. Since sub-targets from k distributions could be quite distinct in visual realism, adapting to a mixed target probably would result in drastic category mismatching and arouses serious negative transfer effects.

4. Adversarial MEta-Adaptation Network

To reap positive transfer performance in BTDA setup, we propose Adversarial MEta-Adaptation Network (AMEAN). AMEAN is coupled of two adversarial learning processes, which are parallely executed to obtain domain-invariant features by transferring data from source $\mathcal{S}$ to mixed target $\mathcal{T}$, and among k "meta-sub-targets" $\{\hat{\mathcal{T}}_j\}_{j=1}^k$ within the mixed target $\mathcal{T}$. The pipeline of AMEAN is concisely illustrated in Fig.2 and we elaborate the methodology as follows.

4.1. Adaptation from a source to a mixed target

Suppose that C is a m-slot softmax classifier proposed on a transferable feature space and F denotes the feature extractor we long to optimize in BTDA. We employ a source-target domain discriminator D_{st} to deploy an adversarial learning scheme, where F is trained to match a source set $\mathcal{S}$ and a mixed target set $\mathcal{T}$ (the unification of $\{\mathcal{T}_j\}_{j=1}^k$) at the feature level, while D_{st} is demanded to separate the source and the target features.

$$\min_{F,C} \max_{D_{st}} V_{\text{st}}(F, D_{st}, C) =$$

$$\lambda \Big(\underbrace{\mathbb{E}_{\boldsymbol{x} \sim \mathcal{S}} \log \big[D_{st}\big(F(\boldsymbol{x})\big) \big] + \mathbb{E}_{\boldsymbol{x} \sim \mathcal{T}} \log \big[1 - D_{st}\big(F(\boldsymbol{x})\big) \big]}_{\text{Adversarial DA loss}} \Big)$$

$$\underbrace{- \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y}) \sim \mathcal{S}} \boldsymbol{y}^T \log C\big(F(\boldsymbol{x})\big)}_{\text{Classification loss}}$$

(1)

. This minimax objective can be optimized in two ways:

1. C, F, D_{st} are jointly trained by inducing a reversed gradient layer [9] [10] across D_{st} and F, which inversely back-propagates the gradients from a source-target domain discriminator D_{st}.
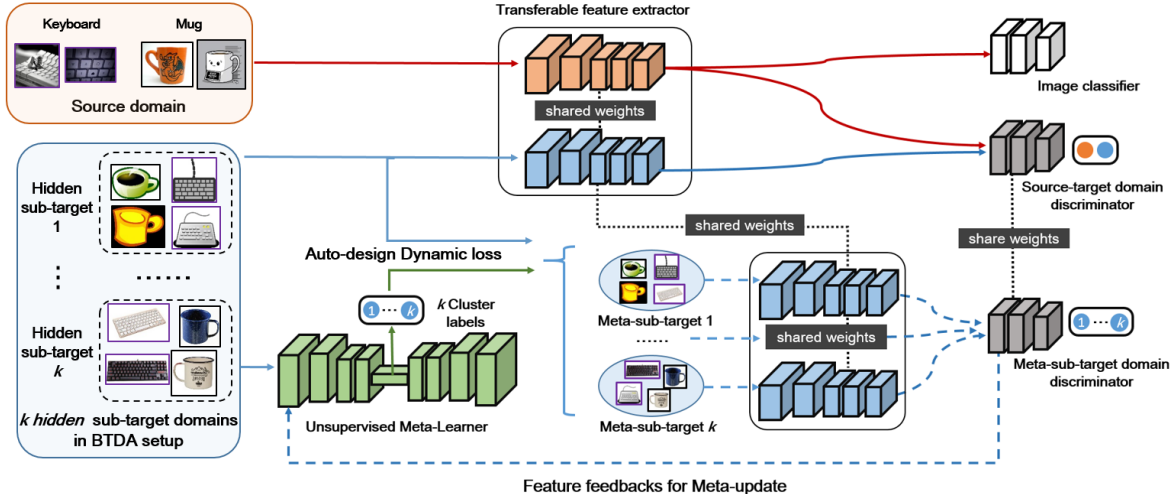
Figure 2. The learning pipeline of our Adversarial MEta-Adaptation Network (AMEAN). AMEAN receives source samples with ground truth and unlabeled samples drawn from an unknown target mixture to synchronously execute two transfer learning processes. In the first process, we propose an adversarial learning bridging our source $\mathcal{S}$ and the mixed target $\mathcal{T}$, so as to learn a feature extractor F that maps them into a common feature space along with a classifier training. In the second stream, AMEAN uses an unsupervised meta-learner U to accept target data and their representations (student feedbacks) and then, learns to separate the mixed target domain into k clusters as meta-sub-targets $\{\hat{\mathcal{T}}_j\}_{j=1}^k$. These meta-sub-targets are iteratively updated to auto-design the multi-target adaptation objectives (Eq.6 7), which operates to progressively eliminate the category mismatching behide the mixed target. Best viewed in color.

2. C, F and D_{st} are updated by an alternative optimization like GAN [13] [34].

Our experiments show them both well-suited in AMEAN.

4.2. Meta-adaptation among meta-sub-targets

The pure source-to-target transfer is blind to the domain gaps and category misalignment among $\{\mathcal{T}_j\}_{j=1}^k$, which the negative transfer mainly ascribes to. Here we interpret the key module of AMEAN to address this problem: an unsupervised meta-learner trained to obtain k embedded clusters [38] as our "meta-sub-targets", which play the roles of automatically and dynamically outputting multi-target adversarial adaptation loss functions (Eq.6, 7) in pursuit of reducing the category mismatching inside the mixed target $\mathcal{T}$.

Meta-learning for dynamic loss design. Meta-learning ("learning to learn") [1] become a thriving topic in machine learning community. It refers to all learnable technique to improve model generalization, which includes fast adaptation to rare categories [37] and unseen tasks [8], as well as auto-tuning the hyper-parameters accompanied with training, *e.g.*, learning rate [1], architecture [47], loss function [39], *etc*. Inspired by them, our meta-learner U is an unsupervised net learning to find deep clustering embeddings that incurs k clusters based on the ongoing feature extractor feedbacks. It induces k meta-sub-targets $\{\hat{\mathcal{T}}_j(U)\}_{j=1}^k$ to take place of $\{\mathcal{T}_j\}_{j=1}^k$. Via this auxiliary, meta-learner has access to auto-design the adversarial multi-target DA loss with respect to the k meta-sub-targets and, dynamically alter the losses according to the change of these meta-sub-targets as our meta-learner updates the k clusters.

Two major concerns are probably raised to our methodology: 1). Are $\{\hat{\mathcal{T}}_j(U)\}_{j=1}^k$ and $\{\mathcal{T}_j\}_{j=1}^k$ similar ? 2). Does the clustering finally lead to target samples of the same category staying together? Towards the first concern, our answer is not and unnecessary. The technical difficulty in BTDA arises from the category misalignment instead of the hidden sub-target domains. As long as a DA algorithm performs appropriate category alignment in a mixed target, it is unnecessary to explicitly discover these hidden sub-target domains. AMEAN receives ongoing learned features that implicitly conceive label information from Eq 1 , to adaptively organizes its meta-adaptation objective. In our ablation, This manner shows powerful to overcome the category misalignment in $\mathcal{T}$. The second concern raises a dilemma to our meta-transfer: as learned features become more classifiable, target features of the same class will be closer and closer, then clustering will more probably select them as a newly-updated meta-target domain. It causes category shift if we apply them to learn AMEAN. To remove this hidden threat, our meta-learner simultaneously receives $\boldsymbol{x}^{(t)}$ and $F(\boldsymbol{x}^{(t)})$ to learn deep clustering embeddings, where $\boldsymbol{x}^{(t)}$ denotes the primitive state of a sample without DA and $F(\boldsymbol{x}^{(t)})$ implies the adaptation feedback from the ongoing learned feature. Since $\boldsymbol{x}^{(t)}$ is feature-agnostic, the induced clusters inherit the merit of meta-learning and concurrently gets rid of this classifiable feature dilemma.

Discovering k meta-sub-targets via deep k-clustering. Our unsupervised meta-learner U is derived from the base model in DEC [38], a denoising auto-encoder (DAE) composed of encoder U_1 and decoder U_2. Distinguished from the DAE in DEC, U takes a target couple $\left(\boldsymbol{x}^t, F(\boldsymbol{x}^{(t)})\right)$ as

4

inputs and learns to satisfy the self-reconstruction as well as obtain deep clustering embedding $U_1\big(\boldsymbol{x}^{(t)}, F(\boldsymbol{x}^{(t)})\big)$. More specifically, suppose $\{\mu_j\}_{j=1}^k$ is the k cluster centroids, and we define a soft cluster assignment $\{q_{i,j}\}_{j=1}^k$ to $\boldsymbol{x}_i^{(t)} \in \mathcal{T}$.

$$q_{i,j} = \frac{(1 + \frac{||U_1(\boldsymbol{x}_i^{(t)}, F(\boldsymbol{x}_i^{(t)})) - \mu_j||^2}{\alpha})^{-\frac{\alpha+1}{2}}}{\sum_{j'}^k (1 + \frac{||U_1(\boldsymbol{x}_i^{(t)}, F(\boldsymbol{x}_i^{(t)})) - \mu_{j'}||^2}{\alpha})^{-\frac{\alpha+1}{2}}} \quad (2)$$

where α indicates the degree of freedom in a Student's t-distribution. Eq.2 is aimed to learn U by iteratively inferring deep cluster centroids $\{\mu_j\}_{j=1}^k$. This EM-like learning operates with the help of auxiliary distributions $\{p_{i,j}, \forall i \in [n_t]\}$, which are computed by raising q_i to the second power and normalize it by the frequency in per cluster, *i.e.*,

$$p_{i,j} = \frac{q_{i,j}^2 / f_j}{\sum_{j'}^k q_{i,j'}^2 / f_{j'}}, \quad s.t. \ f_j = \sum_i^{n_t} q_{i,j} \quad (3)$$

where f_j denotes the jth cluster frequency. Compared with q_i, p_i endows more emphasis on data points with high confidence and thus, is more appropriate to supervise the soft cluster inference. We employ KL divergence to restrict q_i and p_i for the meta-learner clustering network learning:

$$\min_{U_1, U_2, \{\mu_j\}_{j=1}^k} \mathbb{E}_{\boldsymbol{x}_i \sim \mathcal{T}} L_{\text{rec}}(\boldsymbol{x}_i; F) - \sum_{j=1}^k p_{i,j} \log \frac{p_{i,j}}{q_{i,j}} \quad (4)$$

where $L_{\text{rec}}(\boldsymbol{x}; F)$ denotes a l_2 self-reconstruction *w.r.t.* a target feedback pair $\big(\boldsymbol{x}, F(\boldsymbol{x})\big)$ and the second term denotes a KL divergence term for clustering. Parameter learning and k cluster centroid ($\{\mu_j\}_{j=1}^k$) update are facilitated by back-propagation with a SGD solver. (See more in Appendix.A)

After meta-learner U converges, we apply the incurred clustering assignments to separate $\mathcal{T}$ into k meta-sub-target domains, *i.e.*, $\boldsymbol{x}_i^{(t)}$ in a mixed target $\mathcal{T}$ will be classed into meta-sub-target $\hat{\mathcal{T}}_j(U)$ if $q_{i,j}$ is the maximum in $\{q_{i,j'}\}_{j'=1}^k$:

$$\forall j \in [k], \ \hat{\mathcal{T}}_j(U) = \{\boldsymbol{x}_i \in \mathcal{T} \& j = \arg\max_{j'} q_{i,j'}^*\} \quad (5)$$

Meta-sub-target adaptation. Given k meta-sub-target domains $\{\hat{\mathcal{T}}_j(U)\}_{j=1}^k$, AMEAN auto-designs the k-sub-target DA losses to re-align the features in $\mathcal{T}$. More detailedly, AMEAN designates a k-slot softmax classifier D_{mt} sharing parameters with D_{st}, as meta-sub-target domain discriminator. In order to obtain k meta-sub-target adaptations, features extracted by F seek to "maximally confuse" the discriminative decision of D_{mt}:

$$\min_F \max_{D_{mt}} V_{mt}(F, D_{mt}) = \sum_{j=1}^k \mathbb{E}_{\boldsymbol{x} \sim \hat{\mathcal{T}}_j(U)} \mathbf{1}_j^T \log \big[D_{mt}\big(F(\boldsymbol{x})\big)\big] \quad (6)$$

where $\mathbf{1}_j$ indicates a k-dimensional one-hot vector implying that sample $\boldsymbol{x}$ belongs to $\hat{\mathcal{T}}_j(U)$. In the case of joint

Algorithm 1 AMEAN (Stochastic version)

Input: Source $\mathcal{S}$; Mixed target $\mathcal{T}$; feature extractor F; classifier C; domain discriminators D_{st}, D_{mt}; meta-learner $U = \{U_1, U_2\}$.
Output: well-trained F^*, D_{st}^*, D_{mt}^*, C^*.

1: **while** not converged **do**
2:     *Meta-sub-target Discovery* (Meta-update):
3:       Initiate $\{\mu_j\}_{j=1}^k$ and U;
4:     **while** not converged **do**
5:       Sample a mini-batch X_t from $\mathcal{T}$; Construct $\{q_{i,j}\}^k$ by Eq.2 and $\{p_{i,j}\}^k$ by Eq.3 for each $\boldsymbol{x}_i$ in the mini-batch.
6:       Update U and $\{\mu_j\}_{j=1}^k$ in Eq.4 with SGD solver.
7:     **end while**
8:     Divide $\mathcal{T}$ into k meta-sub-targets $\{\hat{\mathcal{T}}_j(U)\}_{j=1}^k$ by Eq.5.
9:     *Collaborative Advesarial Meta-Adaptation*:
10:     **for** 1:M **do**
11:       Sample a mini-batch X_s from $\mathcal{S}$ and k mini-batches $\{X_t^{(j)}\}_{j=1}^k$ from $\{\hat{\mathcal{T}}_j(U)\}_{j=1}^k$ respectively; $X_t = \cup X_t^{(j)}$.
12:       **if** alternating domain adaptation **then**
13:         Update D_{st} and D_{mt} with Eq.9 .
14:         Update F and C with Eq.10 .
15:       **else** Update D_{st}, D_{mt}, F and C with Eq.8 .
16:       **end if**
17:     **end for**
18: **end while**
19: **return** $F^* = F$; $C^* = C$; $D_{st}^* = D_{st}$; $D_{mt}^* = D_{mt}$.

parameter learning, due to the mutual architectures of D_{st} and D_{mt}, Eq.6 is implemented by the same reversed gradient layer originally for source-to-target transfer. However, if subnets F and D_{mt} are alternatively trained, Eq.6 is solely used to update D_{mt} while we prefer to optimize feature extractor F by maximizing the cross-entropy of $D_{mt}\big(F(\boldsymbol{x})\big)$:

$$\min_F \widetilde{V}_{\text{mt}}(F, D_{mt}) = \sum_{j=1}^k \mathbb{E}_{\boldsymbol{x} \sim \hat{\mathcal{T}}_j(U)} D_{mt}\big(F(\boldsymbol{x})\big)^T \log \big[D_{mt}\big(F(\boldsymbol{x})\big)\big] \quad (7)$$

It implies that F learns to "confuse" multi-target domain discriminator D_{mt}, namely, D_{mt} could not identify which meta-sub-target an unlabeled example belongs to.

4.3. Collaborative Advesarial Meta-Adaptation

In order to learn domain-invariant features as well as resist the negative transfer from a mixed target, the previous transfer processes should be combined to battle the domain shifts. In particular, we retrain our meta-learner to update meta-DA losses $V_{\text{mt}}(F, D_{\text{mt}})$ and $\widetilde{V}_{\text{mt}}(F, D_{\text{mt}})$ per M iteration during feature learning. After that, if we employ a reverse gradient layer as the adversarial DA implementation (Eq.1), the collaborative learning objective is formulated as

$$\max_{D_{\text{st}}, D_{\text{mt}}} \min_{F, C} V_{\text{joint}}(D_{\text{st}}, D_{\text{mt}}, F, C)$$
$$= V_{\text{st}}(F, D_{\text{st}}, C) + \gamma V_{\text{mt}}(F, D_{\text{mt}}) \quad (8)$$

where γ indicates the balance factor between two transfer processes. Eq.8 suits joint learning *w.r.t.* $F, D_{\text{st}}, D_{\text{mt}}, C$,

Table 1. Classification accuracy (ACC %)on Digit-five in BTDA setup. **BLUE**, **RED** indicate the baseline suffer from *absolute negative transfer* (ANT%) or *relative negative transfer* (RNT%), respectively. Best viewed in color.

Models	mt→mm,sv,up,sy ACC^{ANT}	RNT	mm→mt,sv,up,sy ACC^{ANT}	RNT	sv→mm,mt,up,sy ACC^{ANT}	RNT	sy→mm,mt,sv,up ACC^{ANT}	RNT	up→mm,mt,sv,sy ACC^{ANT}	RNT	Avg ACC^{ANT}	RNT
Backbone-1:												
Source only	26.9	0	56.0	0	67.2	0	73.8	0	36.9	0	52.2	0
ADDA	43.7	-8.0	$55.9^{(-0.1)}$	-3.3	$40.4^{(-26.8)}$	-21.7	$66.1^{(-6.7)}$	-6.5	$34.8^{(-0.1)}$	-13.3	$48.2^{(-4.0)}$	-10.5
DAN	31.3	-7.5	$53.1^{(-2.9)}$	-3.1	$48.7^{(-18.5)}$	-9.5	$63.3^{(-10.5)}$	-3.9	$27.0^{(-9.9)}$	-11.0	$44.7^{(-7.5)}$	-7.0
GTA	44.6	-9.2	$54.5^{(-1.5)}$	-2.1	$60.3^{(-6.9)}$	-3.9	74.5 (+0.7)	-1.1	41.3	-2.0	55.0	-3.7
RevGrad	52.4	-8.9	64.0	-4.1	$65.3^{(-1.9)}$	-4.1	$66.6^{(-6.2)}$	-7.5	44.3	-6.3	58.5	-6.2
AMEAN	**56.2 (+3.8)**	-	**65.2 (+1.2)**	-	**67.3 (+0.1)**	-	$71.3^{(-2.5)}$	-	**47.5 (+3.2)**	-	**61.5 (+3.0)**	-
Backbone-2:												
Source only	47.4	0	58.1	0	73.8	0	74.5	0	50.6	0	60.8	0
VADA	76.0	-5.3	72.3	-2.3	75.6	-2.5	81.3	-3.8	56.4	-8.7	72.3	-4.5
DIRT-T	73.5	-7.1	76.1	-1.5	75.9	-5.5	78.5	-3.1	47.0	-7.5	70.2	-5.0
AMEAN	**85.1 (+9.1)**	-	**77.6 (+1.5)**	-	**77.4 (+1.5)**	-	**84.1 (+2.8)**	-	**75.5 (+19.1)**	-	**80.0 (+7.7)**	-

while in an alternating adversarial manner, it would be more appropriate to iteratively update $\{F, C\}$ and $\{D_{st}, D_{mt}\}$ by switching the optimization objectives between

$$\max_{D_{st}, D_{mt}} V_{alter}(D_{st}, D_{mt}) = V_{st}(F, D_{st}, C) + V_{mt}(F, D_{mt})$$
(9)

$$\min_{F, C} V_{alter}(F, C) = V_{st}(F, D_{st}, C) + \gamma \widetilde{V}_{mt}(F, D_{mt})$$
(10)

In a summary, the stochastic learning pipeline of AMEAN is described by Algorithm.1 .

5. Experiments

In this section, we elaborate comprehensive experiments in the BTDA setup and compare AMEAN with state-of-the-art DA baselines. Code is available at http:// URL .

5.1. Setup

Benchmarks. *Digit-five* [40] is composed of five domain sets drawn from **mt** (*MNIST*) [20], **mm** (*MNIST-M*) [10], **sv** (*SVHN*) [28], **up** (*USPS*) and **sy** (*Synthetic Digits*) [10], respectively. There are 25000 for training and 9000 for testing in **mt**, **mm**, **sv**, **sy**, while the entire *USPS* is chosen as a domain set **up**. *Office-31* [31] is a famous visual recognition benchmark comprising 31 categories and totally 4652 images in three separated visual domains **A** (*Amazon*), **D** (*DSLR*), **W** (*Webcam*), which indicate images taken by web camera and digital camera in distinct environments. *Office-Home* [36] consists of four visual domain sets, i.e., Artistic (**Ar**), Clip Art (**Cl**), Product (**Pr**) and Real-world (**Rw**) with 65 categories and around 15, 500 images in total.

Baselines. Since MSDA, MTDA approaches require domain remarks, they obviously do not suit the BTDA setup. Therefore we compare our AMEAN with existing (*single-source-single-target*) DA baselines and evaluate their classification accuracies by transferring class information from source $\mathcal{S}$ to mixed target $\mathcal{T}$. State-of-the-art DA baselines include: Deep Adaptation Network (**DAN**) [22], Residual Transfer Network (**RTN**) [24], Joint Adaptation Network (**JAN**) [25], Generate To Adapt (**GTA**) [33], Adversarial Discriminative Domain Adaptation (**ADDA**) [35], Reverse Gradient (**RevGred**) [9] [10], Virtual Adversarial

Domain Adaptation (**VADA**) [34] and its variant **DIRT-T** [34]. DAN, RTN and JAN proposed MMD-based regularizer to pursue cross-domain distribution matching in a feature space; ADDA, RevGred, GTA and VADA are domain adversarial training paradigms encouraging domain-invariant feature learning by "cheating" their domain discriminators. DIRT-T is built upon VADA by introducing a network to guide the dense target feature regions away from the decision boundary. Beyond these approaches, we also report the **Source-only** results based on F and C that are merely trained on source labeled data to classify target examples.

Implementation setting. In digit recognition, we evaluate AMEAN on two different backbones. The first is derived from a LeNet architecture with F, D_{st}, D_{mt}, C jointly trained through a reversed gradient layer (Eq.8); the second employs a GAN-based alternating learning scheme [13] that switches the optimization between Eq.9 , 10 . For a fair comparison, all baselines in Digit-five experiment are based on these backbones. In Office-31 and Office-Home, we evaluate all baselines with AlexNet[19] and ResNet-50 [16], where our AMEANs are trained by Eq.8. Our meta-learner employ the same architecture in all experiments, *i.e.*, a four-layered fully-connected DAE. More implementation details are deferred in our Appendix.A.

5.2. Evaluation Criteria.

Our experimental evaluation is aimed to answer two fundamental questions in this paper:

1. *Does BTDA bring more transfer learning risks to existing DA algorithms* ?

2. *Is our AMEAN able to reduce these transfer risks* ?

As a primal metric, classification accuracies on a mixed target ($Acc_{\mathbf{BTDA}}$) are provided to evaluate DA baselines in the BTDA setup, where their adaptations from $\mathcal{S}$ to a mixed $\mathcal{T}$ are performed to cultivate a classifier that predicts labels on a mixed target test set. To answer the questions above, we also consider two additional metrics, *i.e.*, *absolute negative transfer* (ANT) and *relative negative transfer* (RNT):

- **Absolute negative transfer (ANT).** Given a DA baseline, if its performance is inferior to its Source-only, it

Table 2. Classification accuracy (ACC %) on Office-31 in BTDA setup. **BLUE**, **RED** indicate the baseline suffer from *absolute negative transfer* (ANT%) or *relative negative transfer* (RNT%), respectively. Best viewed in color.

Backbones	Models	A→D,W ACC^{ANT}	RNT	D→A,W ACC^{ANT}	RNT	W→A,D ACC^{ANT}	RNT	Avg ACC^{ANT}	RNT
AlexNet	Source only	62.4	0	60.8	0	57.2	0	60.1	0
	DAN	68.2	-0.2	58.7$^{(-2.1)}$	-5.0	55.6$^{(-1.6)}$	-4.9	60.8	-3.4
	RTN	71.6	-1.1	56.3$^{(-4.5)}$	-4.6	52.2$^{(-5.0)}$	-6.1	59.9$^{(-0.2)}$	-4.1
	JAN	73.7	-0.3	62.1	-4.9	58.4	-3.6	64.7	-3.0
	RevGrad	74.1	+0.9	58.6$^{(-2.2)}$	-4.3	55.0$^{(-2.2)}$	-3.4	62.6	-2.2
	AMEAN (ours)	**74.5 (+0.4)**	-	**62.8 (+0.7)**	-	**59.7 (+1.3)**	-	**65.7 (+1.0)**	-
ResNet-50	Source only	68.6	0	70.0	0	66.5	0	68.4	0
	DAN	78.0	-2.1	64.4$^{(-5.6)}$	-6.8	66.7	-1.8	69.7	-3.6
	RTN	84.3	+2.3	67.5$^{(-2.5)}$	-5.5	64.8$^{(-0.2)}$	-5.5	72.2	-2.9
	JAN	84.2	-1.2	74.4	-0.8	72.0	-2.8	76.9	-1.6
	RevGrad	78.2	-3.3	72.2	-2.7	69.8	-2.8	73.4	-2.9
	AMEAN (ours)	**90.1 (+5.8)**	-	**77.0 (+2.6)**	-	**73.4 (+1.4)**	-	**80.2 (+3.4)**	-

Table 3. Classification accuracy (ACC %) on Office-Home in BTDA setup. **BLUE**, **RED** indicate the baseline suffer from *absolute negative transfer* (ANT%) or *relative negative transfer* (RNT%), respectively. Best viewed in color.

Backbones	Models	Ar→Cl,Pr,Rw ACC^{ANT}	RNT	Cl→Ar,Pr,Rw ACC^{ANT}	RNT	Pr→Ar,Cl,Rw ACC^{ANT}	RNT	Rw→Ar,Cl,Pr ACC^{ANT}	RNT	Avg ACC^{ANT}	RNT
AlexNet	Source only	33.4	0	37.6	0	32.4	0	39.3	0	35.7	0
	DAN	39.7	-3.7	43.2	-3.0	39.4	-3.4	47.8	-2.2	42.5	-3.1
	RTN	42.8	-2.0	45.2	-2.5	40.6	-2.4	49.6	-2.7	44.6	-2.3
	JAN	43.5	-2.9	46.5	-3.6	40.9	-6.6	49.1	-2.1	45.0	-4.6
	RevGrad	42.1	-3.3	45.1	-4.4	41.1	-4.5	48.4	-5.6	44.2	-4.4
	AMEAN (ours)	**44.6 (+1.1)**	-	**47.6 (+1.1)**	-	**42.8 (+1.7)**	-	**50.2 (+1.1)**		**46.3 (+1.3)**	-
ResNet-50	Source only	47.6	0	42.6	0	44.2	0	51.3	0	46.4	0
	DAN	55.6	-0.8	56.6	+0.9	48.5	-5.1	56.7	-6.3	54.4	-2.6
	RTN	53.9	-1.8	56.7	-1.3	47.3	-3.8	51.6	-2.8	52.4	-2.6
	JAN	58.3	-0.4	60.5	+2.3	52.2	-2.2	57.5	-7.0	57.1	-1.9
	RevGrad	58.4	-3.1	58.1	-2.2	52.9	-4.5	62.1	-3.0	57.9	-3.2
	AMEAN (ours)	**64.3 (+5.9)**	-	**65.5 (+5.0)**	-	**59.5 (+6.1)**	-	**66.7 (+4.6)**		**64.0 (+6.1)**	-

implies that this DA algorithm not only fails to benefit but also damages the classifier, *i.e.*, suffers from ANT.

- **Relative negative transfer (RNT).** RNT aims to measure how much performance drops if a DA baseline alters from MTDA to BTDA setups. In MTDA, each DA baseline performs adaptation from source $\mathcal{S}$ to each explicit target $\mathcal{T}_j$ ($\forall j \in [k]$). It results in k target-specific domain-adaptive models with their accuracies $\{Acc_j\}_{j=1}^k$ on the target test sets respectively. Towards this end, we compute the MTDA weighted averaged accuracy by $Acc_{\mathbf{MTDA}} = \sum_{j=1}^k \alpha_j \, Acc_j$ where α_j indicates the ratio of sub-target $\mathcal{T}_j$ in a mixed target $\mathcal{T}$. Hence we have RNT $= Acc_{\mathbf{BTDA}} - Acc_{\mathbf{MTDA}}$.

More details of the metrics can be found in our Appendix.B.

5.3. Results.

The evaluations based on classification accuracy (ACC, *i.e.*, Acc_{BTDA}), ANT and RNT have been conducted in Tables.1 - 3 (we highlight ANT, RNT in **BLUE**, **RED**).

In **Digit-five** (Table.1), ANT frequently occurs in BTDA setups. In sv→mm,mt,up,sy and sy→mm,mt,up,sv, few DA algorithms are exempt for these performance degradation under backbone-1 (*i.e.,* -26.8 for ADDA). This situation is ameliorated on backbone-2. However, all DA baselines suffer from RNT. Especially when an adaptation process starts from a simple source to complex targets, *i.e.*, mt→mm,sv,up,sy, the accuracy drop from MTDA to BTDA

is significant. In **Office-31.** (Table.2), due to D and W sharing similar visual appearances, most of their categories are aligned well in the mixed target. Therefore ANT and RNT are suppressed in A→ D,W. But in the other transfer tasks, ANT and RNT still haunt the performances of DA baselines. In **Office-Home** (Table.3), all DA baselines get rid of ANT, whereas they remain inferior to their performances in MTDA setup (suffer RNT). Observe that, deeper models, *e.g.*, ResNet-50, may encourage the baselines to resist ANT. But the deeper models do not help reduce RNTs across DA algorithms. These evidences sufficiently verify the hardness of BTDA and answer the first question.

Though BTDA is a challenging transfer setup, **AMEAN** presents as a ideal solver. As shown in Table.1 -3 , AMEAN achieved the state-of-the-art in 29 out of 30 BTDA transfer cases, and its average accuracy exceeds the second best by $1.0 \sim 7.7\%$. AMEAN almost achieve positive transfer in all transfer cases, and has reaped huge transfer gains in some of them, *e.g.*, +37.7% in mt→mm,sv,up,sy, +21.5% in A→ D,W, +31.3% in Ar→Cl,Pr,Rw, *etc*. More importantly, AMEAN obtains more impressive performances from deeper architectures, which demonstrates its superiority to address BTDA problem.

5.4. Analysis.

Adaptation visualization. For the BTDA transfer task **mt → mm,sv,up,sy** in Digit-five, we visualize the classifi-
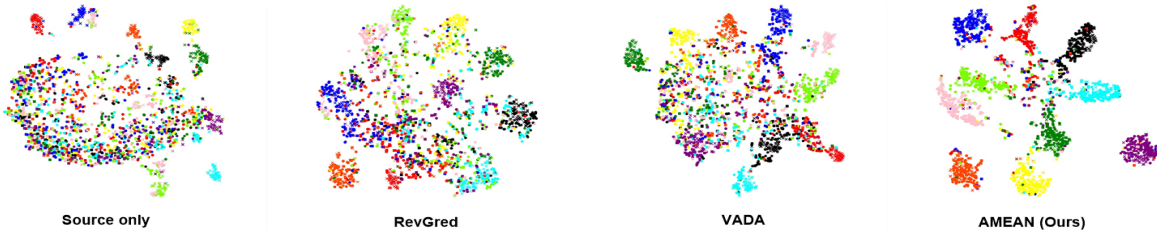
Figure 3. t-SNE visualizations of the features learned by Source-only, RevGred, VADA and AMEAN on Digit-five in BTDA setup. Shapes and colors indicate different domains and categories, respectively. Best viewed in color.
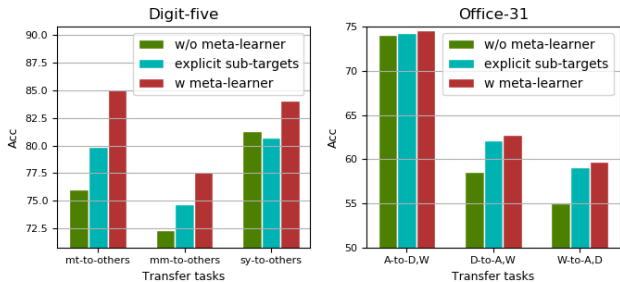

Figure 4. Ablation studies of our meta-learner across three transfer tasks in Digit-five (Backbone-2) and in Office-31 (AlexNet).

cation activations from Source-only, RevGred, VADA and AMEAN in BTDA setup. As can be seen in Fig.3 , Source-only barely captures any classification patterns. In a comparison, RevGred and VADA show better classifiable visualization patterns than Source-only's. But their activations remain pretty messy and most of them are misaligned in their classes. It demonstrates that BTDA is a very challenging scenario for existing DA algorithms. Finally, the activations from AMEAN show clear classification margins. It illustrates the superior transferability of our AMEAN.

Ablation study. The crucial component of our AMEAN is the meta-learner for auto-designing V_{mt}. Hence our ablation focuses on this model-driven auto-learning technique. In particular, we evaluate three adaptation manners derived from our AMEAN: adaptation without meta-learner (w/o meta learner); adaptation without meta-learner but using explicit sub-target ($\{\mathcal{T}_j\}_{j=1}^k$) to guide the transfer in BTDA (explicit sub-targets); adaptation with meta-learner (w meta-learner, AMEAN). As illustrated in Fig.4 , explicit sub-target information is not persistently helpful to BTDA. In **sy**-to-others, explicit sub-target information even draws back the source-to-target transfer gain. By contrast, meta-learner plays a key role to enhance the adaptation towards digit and real-world visual domains and obtain state-of-the-art in BTDA. Surprisingly, the dynamical meta-sub-targets even drive the adaptation model exceed those trained with the explicit sub-target domains.

To further investigate the meta-adaptation dynamic provided by AMEAN, we ablate the multi-target DA by following different target separation strategies: 1). explicit-sub-target (EST); 2).static deep k-clustering (k-C); 3) AMEAN. Note that, 2) ablates the auto-loss-design dynamic in V_{mt},

Table 4. The ablation of V_{mt} constructed by Explicit Sub-Target (EST), deep k-Clustering (k-C) and AMEAN (ours).

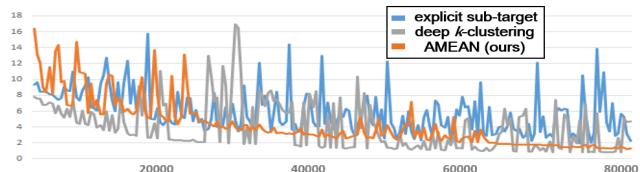|	mt→mm,sv,up,sy	mm→mt,sv,up,sy	D→A,W	W→A,D
EST	79.9	74.7	62.2	58.4
k-C	81.6	74.5	61.3	59.1
ours	**85.1**	**77.6**	**62.8**	**59.7**

Figure 5. The ablation of V_{mt} based on mt→mm,sv,up,sy in Digit-five, where V_{mt} is constructed in different manners over iterations. AMEAN performs a faster and more stable optimization process.

as it solely uses the initial clusters to divide the mixed target and keep a static V_{mt} along model training. The comparison of 2) and 3) helps to unveil whether our auto-loss strategy facilitates AMEAN. As shown in Table 4 , 1) and 2) disregard label information given by a source and their V_{mt} still suffer from a risk of class mismatching. Our auto-loss manner adaptively changes V_{mt} by receiving label information from the features previously learned by Eq.1 and thus, achieve better performance in BTDA. More importantly, it also encourages a fast and more stable adaptation during the minimax optimization process (see Fig 5).

6. Concluding Remarks

In this paper, we concern a realistic adaptation scenario, where our target domain is comprised of multiple hidden sub-targets and learners could not identify which sub-target each unlabeled example comes from. This Blending-target domain adaptation (BTDA) conceives category mismatching risk and if we apply existing DA algorithm in BTDA, it will lead to negative transfer effect. To take on this uprising challenge, we propose Adversarial MEta-Adaptation Network (AMEAN). AMEAN starts from the popular adversarial adaptation methods, while newly employs a meta-learner network to dynamically devise a multi-target DA loss along with learning domain-invariant features. The AutoML merits are inherited by AMEAN to reduce the class mismatching in the mixed target domain. Our experiments focus on verifying the threat of BTDA and the efficacy of AMEAN in BTDA. Our evaluations show the significance of BTDA problem as well as the superiority of our AMEAN.

References

[1] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, pages 3981–3989, 2016.

[2] Anonymous. Unsupervised multi-target domain adaptation: An information theoretic approach. In *Submitted to International Conference on Learning Representations*, 2019. under review.

[3] J. C. Balloch, V. Agrawal, I. Essa, and S. Chernova. Unbiasing semantic segmentation for robot perception using synthetic data feature transfer. *arXiv preprint arXiv:1809.03676*, 2018.

[4] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

[5] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, et al. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4243–4250. IEEE, 2018.

[6] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. *arXiv preprint arXiv:1612.05424*, 2016.

[7] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool. Domain adaptive faster r-cnn for object detection in the wild. 2018.

[8] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. 2017.

[9] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

[10] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. *Domain-Adversarial Training of Neural Networks*. 2017.

[11] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016.

[12] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning*, pages 201–210, 2016.

[13] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *International Conference on Neural Information Processing Systems*, pages 2672–2680, 2014.

[14] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 999–1006. IEEE, 2011.

[15] A. Gretton, A. J. Smola, J. Huang, M. Schmittfull, K. M. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. 2009.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. 2015.

[17] J. Hoffman, E. Tzeng, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *Domain Adaptation in Computer Vision Applications*, pages 173–187. Springer, 2017.

[18] J. Hoffman, D. Wang, F. Yu, and T. Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[21] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016.

[22] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pages 97–105, 2015.

[23] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. *arXiv preprint arXiv:1605.06636*, 2016.

[24] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016.

[25] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. 2017.

[26] M. Mancini, L. Porzi, S. R. Bulò, B. Caputo, and E. Ricci. Boosting domain adaptation by discovering latent domains. *arXiv preprint arXiv:1805.01386*, 2018.

[27] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation with multiple sources. In *Advances in neural information processing systems*, pages 1041–1048, 2009.

[28] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. *Nips Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[29] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[30] A. A. Rusu, M. Vecerik, T. Rothörl, N. Heess, R. Pascanu, and R. Hadsell. Sim-to-real robot learning from pixels with progressive nets. *arXiv preprint arXiv:1610.04286*, 2016.

[31] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. *Computer Vision–ECCV 2010*, pages 213–226, 2010.

[32] K. Saito, Y. Ushiku, and T. Harada. Asymmetric tri-training for unsupervised domain adaptation. *arXiv preprint arXiv:1702.08400*, 2017.

[33] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. *ArXiv e-prints, abs/1704.01705*, 2017.

[34] R. Shu, H. H. Bui, H. Narui, and S. Ermon. A dirt-t approach to unsupervised domain adaptation. *arXiv preprint arXiv:1802.08735*, 2018.

[35] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. *arXiv preprint arXiv:1702.05464*, 2017.

[36] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proc. CVPR*, pages 5018–5027, 2017.

[37] Y. X. Wang, R. Girshick, M. Hebert, and B. Hariharan. Low-shot learning from imaginary data. 2018.

[38] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487, 2016.

[39] H. Xu, H. Zhang, Z. Hu, X. Liang, R. Salakhutdinov, and E. Xing. Autoloss: Learning discrete schedules for alternate optimization. 2018.

[40] R. Xu, Z. Chen, W. Zuo, J. Yan, and L. Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3964–3973, 2018.

[41] J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive svms. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 188–197. ACM, 2007.

[42] L. Yang, X. Liang, T. Wang, and E. Xing. Real-to-virtual domain unification for end-to-end autonomous driving. 2018.

[43] Y. You, X. Pan, Z. Wang, and C. Lu. Virtual to real reinforcement learning for autonomous driving. 2017.

[44] H. Yu, M. Hu, and S. Chen. Multi-target unsupervised domain adaptation without exactly shared categories. 2018.

[45] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger, and S. Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning. *arXiv preprint arXiv:1702.08811*, 2017.

[46] H. Zhao, S. Zhang, G. Wu, J. ao P. Costeira, J. M. F. Moura, and G. J. Gordon. Multiple source domain adaptation with adversarial learning, 2018.

[47] B. Zoph and Q. V. Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.