HHS Public Access

Author manuscript

Proc IEEE Int Conf Data Min. Author manuscript; available in PMC 2023 April 12.

Published in final edited form as:

Proc IEEE Int Conf Data Min. 2022 ; 2022: 1299–1304. doi:10.1109/icdm54844.2022.00171.

Robust Unsupervised Domain Adaptation from A Corrupted Source

Shuyang Yu[*], **Zhuangdi Zhu**[*], **Boyang Liu**[*], **Anil K. Jain**[*], **Jiayu Zhou**[*]

[*]Department of Computer Science and Engineering Michigan State University

Abstract

Unsupervised Domain Adaptation (UDA) provides a promising solution for learning without supervision, which transfers knowledge from relevant source domains with accessible labeled training data. Existing UDA solutions hinge on clean training data with a short-tail distribution from the source domain, which can be fragile when the source domain data is corrupted either inherently or via adversarial attacks. In this work, we propose an effective framework to address the challenges of UDA from corrupted source domains in a principled manner. Specifically, we perform knowledge ensemble from multiple domain-invariant models that are learned on random partitions of training data. To further address the distribution shift from the source to the target domain, we refine each of the learned models via mutual information maximization, which adaptively obtains the predictive information of the target domain with high confidence. Extensive empirical studies demonstrate that the proposed approach is robust against various types of poisoned data attacks while achieving high asymptotic performance on the target domain.

Keywords

Unsupervised Domain Adaptation; Robust Learning; Poison Data Attack

I. Introduction

Deep learning techniques have been thriving over the last decade as a powerful tool for predictive modeling in a variety of domains, including computer vision [1], autonomous vehicles [2], and healthcare [3], to name just a few. The over-parameterization design of deep models gives them ultrahigh model flexibility, which gives them the power to capture complex mappings between input data points and target labels. The success of deep learning-based predictive modeling, however, hinges on massive training data with accurate labels, which hinders its application to tasks with limited training label supervision, where collecting accurate labels can be economically prohibitive. Longitudinal studies, strict enrollment conditions, data coding errors, and high costs associated with the data collection often result in only very small datasets being available for supervised learning [4].

Domain adaptation (DA) has emerged as an effective solution, which transfers knowledge learned from a related but different domain (i.e. the source domain) to assist the learning

yushuyan@msu.edu .

of the target domain. In particular, a challenging and practical problem along this line is *unsupervised domain adaptation* (UDA), in which the target domain has access to only a few *unlabeled* training samples. While UDA has been extensively studied for typical machine learning settings, most existing UDA methods are usually built upon an implicit assumption that source domain data is clean. Under this assumption, UDA methods are prone to performance degradation when the source domain samples are corrupted, either unintentionally during data collection or deliberately by vicious attackers. Consequently, models learned on the corrupted source data can be easily under attack even on the source domain, not to mention confronting the challenges of domain distribution shift when adapting to the target domain. Such model performance degradation can be exacerbated under adversarial attacks. For instance, as illustrated in Figure 1, a minimal corruption in source domain samples shifts the model's hypothesis plane drastically when performing domain adaptation, especially due to the lack of labeled supervision in the target domain.

Given the challenge of UDA under the corrupted source domain, in this work, we propose a simple yet effective solution for robust UDA that addresses various types of data corruption. Specifically, inspired by the principle of Median of Means (MoM) estimators [5], we alleviate the impacts of corrupted training samples by ensemble learning on a group of lightweight models with domain-invariant features, which is shown to be effective in confronting poisoned data. To further address the distribution shift inherent in domain adaptation, we refine the learned models by maximizing the mutual information between the latent feature representations and the posterior distributions. Eventually, the final ensemble model can attain the predictive knowledge of the target domain with high confidence.

The merits of our proposed approach are multi-fold: i) It is a principled and effective solution in defending contaminated training samples. ii) The proposed solution to UDA is generally robust against agnostic types of data corruption. In particular, our approach can successfully tackle notorious *backdoor attacks*, where both the training samples and corresponding labels may be maliciously modified by attackers. iii) The proposed learning framework can be flexibly combined with existing UDA approaches that are orthogonal to our work to improve their robustness under corrupted data.

II Related Work

Domain Adaptation (DA) has been applied to a number of practical applications, including semantic segmentation [6], objective detection [7], etc. In this work, we work on the problem setting of **unsupervised domain adaptation** (UDA), which is more challenging than semi-supervised domain adaptation [8] where a few labeled samples of the target domain are available to assist learning. Among various UDA approaches, *domain invariant representations* reside at their core. A plethora of work has been proposed to learn feature representations that are *discriminative* for prediction while being *invariant* among domains. Earlier work leveraged the idea of minimizing the Maximum Mean Discrepancy (MMD) to achieve feature invariance [9]. Adversarial training approaches emerged to minimize the discrepancy of the latent feature distributions between different domains [10]. Moment matching was also widely utilized for learning latent representations [11], which can be combined with generative adversarial learning for improving such domain-invariance [12].

Another direction towards solving UDA is based on data reconstruction [13]. Most existing approaches did not tackle the issue of source domain corruption.

Learning with noisy data has been extensively studied in traditional, non-domain adaptation settings. Numerous robust learning methods have been proposed for tackling feature corruption, label corruption, and data poisoning attacks [14], [15]. However, the problem of learning with noisy data for DA is not well studied. Most of the existing robust DA methods are limited to one or two particular types of noise in data. [16] addressed domain adaptation under missing classes by performing a unilateral alignment. [17], [18] solves DA in a scenario where only the labels are noisy, with input features untouched. [19] proposed a marginalized Stacked denoising autoencoders (mSDA) to address feature corruption for DA. [20] developed an offline curriculum learning approach to tackle the label noise of DA, and adopted a proxy distribution based margin discrepancy to alleviate feature noise.

Median of Means (MoM) Estimators [5] are robust estimators utilizing the median of the predictions. [21] showed that MoM has a theoretical advantage over classical ERM-based approaches given long-tailed data with outliers, which can be very effective for solving general noisy data problems. [22], [23] applied MoM for robust predictive learning. In this paper, we leverage MoM to solve UDA with data corruption.

III Problem Setting

Unsupervised Domain Adaptation (UDA) addresses learning in a *target* domain without any label supervision via leveraging knowledge obtained from a *source* domain. Denote $\mathscr{P}_s^{xy} := \mathscr{P}_s(X) \times \mathscr{P}_s(Y)$ as the distribution of the source domain, and $\mathscr{P}_t^{xy} = \mathscr{P}_t(X) \times \mathscr{P}_t(Y)$ as the distribution of the target domain, respectively. One can access *labeled* samples from the source domain, denoted as $\mathscr{D}_s := \left\{x_s^i, y_s^i\right\}_{i=1}^{N_s} \subset \mathscr{P}_s^{xy}$. Accordingly, let $\mathscr{D}_t := \left\{x_t^j\right\}_{j=1}^{N_t} \subset \mathscr{P}_t\left(X\right)$ be the set of *unlabeled* samples accessible in the target domain, Denote the loss function for the target domain as $\mathscr{L}: \triangle^{\mathscr{Y}} \times \mathscr{Y} \to \mathbb{R}^+$, where $\triangle^{\mathscr{Y}}$ is the simplex over the label space, with $|\mathscr{Y}| = C$ denoting the number of unique labels. Let Θ be the parameter space of the learning model, and $f(\cdot\,; \theta)$ be the post-activation, prediction output of model $\theta \sim \Theta$. The objective for UDA is to optimize the learning model performance on the target domain:

$$\theta^* = \arg\min_{\theta \in \Theta} \mathbb{E}_{x,y \sim \mathscr{P}_t^{xy}}\left[\mathscr{L}\left(f\left(x; \theta\right), y\right)\right]. \tag{1}$$

In practice, the learning model is derived based on accessible samples from both domains, i.e. $\theta \leftarrow \Phi(\mathscr{D}_s, \mathscr{D}_t)$, where Φ is the learning procedure. Without loss of generality, in this work, we focus on single domain adaptation, and our learning framework can be readily extended to address multi-domain adaptation problems.

UDA with Source Domain Corruption tackles domain adaptation from a *corrupted* source domain. One can consider that there is a one-to-one mapping between the clean source

domain $\mathscr{P}_s^{xy}$ and the corrupted source domain $\widetilde{\mathscr{P}}_s^{xy}$. The input feature x_s^i can be disrupted with probability p_e:

$$p_e := \mathbb{E}_{x_s^i, \tilde{x}_s^i \sim \langle \mathscr{P}_s(\mathscr{X}), \widetilde{\mathscr{P}}_s(\mathscr{X}) \rangle} \left[\mathbb{I}(\tilde{x}_s^i \neq x_s^i) \right].$$

Accordingly, labels of noisy samples are transformed based on an unknown transition probability matrix $\mathscr{T} \in \mathbb{R}^{C \times C}$, where C is the cardinality of label types. Each entry $\mathscr{T}(i, j)$ in $\mathscr{T}$ denotes the probability that a label $i \in [C]$ is flipped to $j \in [C]$ after data corruption:

$$\mathscr{T}(i, j) = \mathbb{E}_{y_s^i, \tilde{y}_s^i \sim \langle \mathscr{P}_s(\mathscr{Y}), \widetilde{\mathscr{P}}_s(\mathscr{X}) \rangle} \left[\mathbb{I}(\tilde{y}_s = j \mid y_s = i) \right].$$

Denote $\widetilde{\mathscr{D}}_s = \{\tilde{x}_s^i, \tilde{y}_s^i\}_{i=1}^{N_s}$ the noisy samples from $\widetilde{\mathscr{P}}_s^{xy}$, the model learned under corrupted source domain is hence derived by noisy source domain samples instead: $\theta \leftarrow \Phi(\widetilde{\mathscr{D}}_s, \mathscr{D}_t)$. Such data corruption can be unconsciously introduced during data collection by human mistakes or sensor malfunction, or maliciously triggered via malicious attacks. It is a challenging yet practical problem setting, potentially undermining most existing UDA approaches that do not consider the risk of noisy source domains (as illustrated in Figure 1).

IV. Methodology

A. Preliminaries of Median of Means

Given a model θ, there exists a gap between the empirical risk $\hat{E}(\theta) := \frac{1}{|\mathscr{D}|} \sum_{x \sim \mathscr{D}} \mathscr{L}(f(x; \theta), y)$, and the true risk $E(\theta) := \mathbb{E}_{x, y \sim P^{xy}}[\mathscr{L}(f(x; \theta), y)]$, which can be exacerbated when the data are heavily tailed or contain contaminated samples. Therefore, models that are learned to soley minimize $\hat{E}(\theta)$ can be sensitive to outliers. Median of Means estimators alleviate such issue by finding a more *proper* approximation of the true risk, compared with an empirical risk minimizer (ERM). Formally, let $\{x^i\}_{i=1}^N$ be N i.i.d. samples from an unknown distribution $\mathscr{P}$. Let the MoM estimator associated with a parameter $\delta \in [e^{1-N/2}, 1]$, then one can *evenly* separate $\{x^i\}_{i=1}^N$ into K blocks, where $K = \lfloor \ln(\delta^{-1}) \rfloor$. The MoM estimator $\mu_{MoM}(\delta)$ is then defined as the *median* of the K arithmetic *mean* of each block X_k:

$$\mu_{MoM}(\delta) = \text{median} \left(\frac{1}{|Z_k|} \sum_{x_i \sim X_k} x_i; \right)_{k=1}^K.$$

The MoM estimator can probably attain subgaussian properties under mild assumptions on the variance of input features. Particularly, $\forall N \geq 4$, one can derive that [24]:

$$\mathbb{P}\left(\left| \mu_{MoM}(\delta) - \mathbb{E}_{\mathscr{P}}[x] \right| > C \sqrt{\frac{1 + \ln(\delta^{-1})}{N}} \right) \leq \delta.$$

Unlike the ERM estimator $\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x^i$, MoM estimator is robust to data with outliers or heavy-tailed inputs. Inspired by MoM, we aim to approximate and minimize the ***centroid*** of the excessive risks by ensemble learning, which is resemblant to the *median* of means when we treat x_i as a sample-wise loss value.

B. Robust UDA via Ensemble Learning

We now elaborate on our learning paradigm. We first randomly split the source domain data $\widetilde{\mathscr{D}}_s := \{\tilde{x}_s^i, \tilde{y}_s^i\}_{i=1}^{N_s}$ into K even blocks $\left\{\widetilde{\mathscr{D}}_s^k\right\}_{k=1}^{K}$, and apply the same random split for the unlabeled target domain data: $\left\{\mathscr{D}_t^k\right\}_{k=1}^{K}$. Next, we learn K separate models with parameters $\{\theta\}_{k=1}^{K}$, while each optimizing towards a domain-adaptation objective using one pair of the ⟨source, target⟩ domain data block, respectively, to minimize the empirical risk:

$$\min_{\{\theta_k \sim \Theta\}_{k=1}^{K}} \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}_{x_s, y_s \sim \widetilde{\mathscr{D}}_s^k, x_t \sim \mathscr{D}_t^k} J_{\mathrm{DA}}\left(x_s, y_s, x_t, \theta_k\right), \tag{2}$$

in which $J_{\mathrm{DA}}(x_s, y_s, x_t; \theta)$ is the domain-adaptation risk function. One highlight of our work is that, we do not constrain the specific form of J_{DA}, hence a variety of UDA approaches proposed by prior arts can be flexibly integrated into our learning framework, by applying different forms of J_{DA} as in need. In practice, J_{DA} is usually derived by adversarial learning to attain a saddle-point solution that captures domain-invariant latent representations [25], [10]. Without the loss of generality, we present one form of J_{DA} as below, although any other legitimate objective forms are also applicable: $J_{\mathrm{DA}}(x_s, y_s, x_t; \theta) :=$

$$\max_{D: \mathscr{X} \to [0,1)} \Bigg[\underbrace{\{\log\,(1 - D(g(x_s; \theta)))\} + \log\,(D(g(x_t; \theta)))}_{(\mathrm{A})} + \underbrace{\mathscr{L}(f(x_s; \theta), y_s)}_{(\mathrm{B})} \Bigg], \tag{3}$$

in which D is a discriminator model inspired by adversarial generative training [26], and $g(\cdot\,; \theta)$ is the latent feature map of model θ. The term (A) in Equation 3 encourages learning a domain-invariant feature representation, while the term (B) in Equation 3 reinforces the predictive power of the model using labeled supervision from the source domain.

Once the K models have been learned, the ***centroid*** prediction of arbitrary sample x can be derived by their ensemble voting:

$$\begin{aligned} \bar{y} &= \text{ensemble}\,\left(x; \{\theta_k\}_{k=1}^{K}\right) \\ &= \arg\max_{y \sim \mathscr{Y}} \sum_{k=1}^{K} \mathbb{I}\left(\arg\max_{c \sim \mathscr{Y}} f(x; \theta_k)_c = y\right), \end{aligned} \tag{4}$$

where $\mathbb{I}$ is an indicator function; $f(x; \theta_k)$ is the posterior distribution output of model θ_k, and $f(x; \theta_k)_c$ indicates the predictive probability of input feature belonging to class c. Therefore, $\bar{y}$

of an input feature x is the most voted label by the K models, which alleviates the influences of potentially contaminated models induced by data corruption.

C. Hypothesis Adaptation by Information Maximization

Up to now, one can derive a conceptual robust model by using the ensemble results from multiple models. To reinforce the performance of models before the final ensemble, we can adapt their hypothesis to the target domain by further leveraging the unlabeled target domain samples. More concretely, we refine each learned model θ_t by maximizing the mutual information between its latent feature representations and its posterior distribution. using the following information maximization objective:

$$\begin{aligned}&\min_{\theta_k} J_{\mathrm{IM}}\left(\mathscr{D}_t;\theta_k\right)\\&:=\underbrace{\mathbb{E}_{x_t\sim\mathscr{D}_t}[H(f(x;\theta_k))]}_{A}-\underbrace{H(\text{softmax }(\mathbb{E}_{x\sim\mathscr{D}_t}[f(x_t;\theta_k)]))}_{B},\end{aligned}\tag{5}$$

where $H(p)$ is the entropy for input $p\sim\triangle^{\mathscr{Y}}$.

This refinement objective aligns with a common perception that, an ideal model shall be confident in its sample-wise predictions (minimize term A), and be diversified on domain-wise predictions (maximize term B). A resemblant strategy has been applied by prior work to address source-free DA [27]. In our setting, optimizing toward this objective shows significant benefits in weakening the impacts of source data corruption, which can adaptively tune the potentially contaminated model to fit in the target domain hypothesis.

Moreover, when refining a model θ_k using target domain samples, we can obtain the pseudo label $\hat{y}_t=\arg\max_{c\sim[C]} f(x_t;\theta)_c$ for each sample x_t, as well as the class-wise *centroid* representation $\bar{g}_k$:

$$\forall k\in[C],\bar{g}_k=\mathbb{E}_{x_t\sim\mathscr{D}_t,\hat{y}_t=k}\left[g(x_t;\theta_k)\right].\tag{6}$$

where $g(x_t;\theta_k)$ is the latent feature representation of x_t, i.e. the penultimate layer output of model θ. We find it beneficial to correct the pseudo labels of x_t by finding the nearest centroid: $\bar{y}_t:=\arg\min_{k\in[C]}\cos\left(g(x_t;\theta_k),\bar{g}_k\right)$, then use the corrected pseudo labels $\bar{y}_t$ to adjust the model. More concretely, this augmented objective J_{PL} is derived as follows:

$$\min_{\theta_k} J_{\mathrm{PL}}:=\mathbb{E}_{x_t\sim\mathscr{D}_t,\bar{y}_t}\left[-\log\left(f(x_t;\theta_k)_{\bar{y}_t}\right)\right].\tag{7}$$

Based on the above building blocks, we now summarize our robust domain adaptation approach in Algorithm 1, in which K models are independently learned using separated training blocks, then refined to adapt their model hypothesis into the target domain by optimizing Equation 5 and Equation 7, where α and β are the constant *w.r.t* the gradient of Equation 5 and Equation 7, respectively. Note that for each learning batch i, we iteratively

adjust the centroid $\bar{g}_k$ using the updated model. Eventually, their ensemble voting is used as the final prediction for the target domain.

V. Evaluation

In this section, we conduct extensive experiments[1] on multiple benchmark datasets to investigate the following question: whether our approach is effective for unsupervised domain adaptation, given a corrupted source domain data?

A. Experiment Setup

Dataset: We conducted experiments using the following datasets: 1) **Digit datasets**: the UDA tasks from MNIST [28] to USPS [29] (M → U), and from USPS to MNIST (U → M), respectively. 2) **Image datasets:** the UDA task from CIFAR 10 [30] to STL [31] with the non-overlapping class of these two detests removed. Hence, these two domains are redefined as 9-class classification tasks. We also downscale the original image dimension of STL to the image dimension of CIFAR10.

Algorithm 1

Robust Unsupervised Domain Adaptation

Inputs: labeled source domain dataset $\mathscr{D}_s$; unlabeled target domain dataset $\mathscr{D}_t$; constant K, DA risk function $J_{\mathrm{DA}}: \mathscr{X} \times \mathscr{X} \times \mathscr{Y} \to \mathbb{R}^+$; K models $\{\theta_k\}_{k=1}^K \sim \Theta$ training steps E_1, adaptation steps E_2; constant α, $\beta > 0$.

Randomly split $\mathscr{D}_s$, $\mathscr{D}_t$ into K blocks of pairs: $\left\{\mathscr{D}_s^k, \mathscr{D}_t^k\right\}_{k=1}^K$, $s.t.\ \forall k, \left|\mathscr{D}_s^k\right| \le \left\lceil \frac{|\mathscr{D}_s|}{K} \right\rceil, \left|\mathscr{D}_t^k\right| \le \left\lceil \frac{|\mathscr{D}_t|}{K} \right\rceil$.

for $k \sim [K]$ in parallel **do**

  for $1 \le i \le E_1$ **do**

    $\theta_k \leftarrow \theta_k - \eta * \nabla_{\theta_k} \mathbb{E}_{D_s^k, D_t^k}[J_{\mathrm{DA}}(x_s, y_s, x_t)]$.

  end for

end for

for $k \sim [K]$ in parallel **do**

  for $1 \le i \le E_2$ **do**

    $\theta_k \leftarrow \theta_k - \eta(\alpha \nabla_{\theta_t} J_{\mathrm{IM}}(\mathscr{D}_t; \theta_k) + \beta J_{\mathrm{PL}}(\mathscr{D}_t; \theta_k))$.

  end for

end for

Return ensemble $\{\theta_k\}_{k=1}^K$.

Compared Approaches: We compare our method against the following approaches: 1) **DANN** is a representative UDA method based on generative-adversarial learning [25]. 2) **CDAN** is short for conditional adversarial domain adaptation, which conditions the model posterior on the discriminative information from the classifier [32].

[1]The code is available at https://github.com/illidanlab/RobustUDA

Implementation: We choose ***backdoor*** attacks as our corruption method because it is a more challenging attack than feature noise or label noise attacks that existing robust DA methods managed to solve. We implement two kinds of backdoor attacks:

1. ***BadNet* Attack** : *BadNet* [33] is one of the most common backdoor attacks. According to a set poison ratio, we add a 5×5 trigger to the upper right corner of each poisoned sample from the source domain. These poisoned samples are also assigned with attacker-specified target labels. Then these poisoned source samples are fed into DNNs along with the remaining clean source samples and a few unlabeled target samples for training. The network is evaluated both on the clean target samples and poisoned target samples which are corrupted the same way as source samples.
2. **Clean Label Backdoor Attack (*CLBD*):** Compared with *BadNet*, CLBD [14] does not change the label of poison samples, but adds a learned adversarial perturbation to each base image. We craft the poison samples on a pre-trained Resnet-18 model using the CIFAR10 dataset, then modify them with a trigger. Note that the poison ratio for $CLBD$ represents the fraction of examples poisoned from a single class, instead of the entire source training samples.

For the digit tasks, we utilize the LeNet-5 [34] network, while for image tasks, we adopt the Resnet-18 network.

Evaluations are performed *w.r.t.* the following criteria:

1) **Target clean accuracy (Clean acc)** refers to the accuracy evaluated on the clean target dataset. 2) **Target poison accuracy (Poison acc)** refers to the accuracy evaluated on the poisoned target data with clean labels. 3) **Attack success rate (Success rate)** refers to the accuracy evaluated on the poisoned target data with poisoned labels. This criterion can help us find out whether hidden backdoors are activated by attacker-specified trigger patterns.

B. Results and Discussions

For the digits tasks (M $\leftrightarrow$ U), we apply *BadNet* attacks and vary the poison ratio from 0.01 to 0.03. For image adaptations between CIFAR10 and STL, we fix the poison ratio to be 0.02 for *BadNet* attacks and 0.5 for *CLBD* attacks.

Effects of MoM on defending poison data attacks: For digit adaptation tasks, we evaluate the accuracy and attack success rates *w.r.t.* different poison ratios for two different base DA approaches: DANN and CDAN, respectively. As shown in Table I, our proposed MoM method is consistently robust given different base DA algorithms. When the poison ratio is 0, there are no poisoning attacks on source data, hence the poison acc and success rate for poison ratio = 0 is evaluated on poisoned testing samples, with a model trained on clean samples. We use this result as a reference for the following experiments. The performance of MoM for image task under *BadNet* and *CLBD* attacks is shown in Figure 3. Block number = 1 refers to training without applying MoM, which we use as the baselines for our proposed algorithm. *We found that by applying MoM, we significantly bring down the attack success rate and improve the target poison test accuracy while maintaining*

the target clean sample accuracy. The results for both tasks can be further improved by adaptation with IM or PL which will be covered later.

Effects of different block numbers for MoM: We also investigate how the number of blocks would affect the performance of our approach. *We observe that increasing the number of blocks within a certain range is beneficial for improving the performance.* The best block number is related to the poison ratio and can be task dependent. For instance, adaptation tasks between CIFAR10 and STL need more blocks to achieve a low attack success rate, compared with digits adaptations. Meanwhile, we show that adaptation with IM or PL (Section IV-C) is more beneficial for enhancing the robustness of our approach, instead of keeping increasing the block number.

Effects of defending poison data attack using adaptation: To further improve the results, we refine our model with adaptation method IM and PL (section IV-C). IM and PL are verified to be effective to not only further decrease the attack success rate but also increase the target poison accuracy. To the best of our knowledge, our proposed method is the most robust DA method given corrupted source samples compared with existing methods. For the digits tasks, we evaluate our proposed MoM + adaptation algorithm with poison ratio=0.03 and block number=20, using two models: DANN and CDAN, respectively, as shown in Table II. For image task, the accuracy and attack success rates for *BadNet* attacks and *CLBD* attacks with the best block size 40 are shown in Table III. Both IM and PL can be used to further improve the results for defending *BadNet* and *CLBD* attacks while IM shows the best results.

VI. Conclusion

In this work, we tackled the problem of unsupervised domain adaptation under corrupted source domain samples. Inspired by the Median of Means estimators, we proposed a principled and robust ensemble learning algorithm powered by hypothesis transfer via information maximization, which can defend corrupted training samples with high performance on the target domain. Extensive empirical studies showed that our UDA approach is robust against agnostic data corruption, which can serve as a general framework to improve the robustness of orthogonal UDA approaches. We leave more complex scenarios, such as corrupted multi-domain adaptation, to our future work.

Acknowledgement

This material is based in part upon work supported by the National Science Foundation under Grant IIS-2212174, IIS-1749940, Office of Naval Research N00014-20-1-2382, and National Institute on Aging (NIA) RF1AG072449.

REFERENCES

[1]. Moeslund TB and Granum E, “A survey of computer vision-based human motion capture,” Computer vision and image understanding, vol. 81, no. 3, pp. 231–268, 2001.

[2]. Heimberger M, Horgan J, Hughes C, McDonald J, and Yogamani S, “Computer vision in automated parking systems: Design, implementation and challenges,” Image and Vision Computing, vol. 68, pp. 88–101, 2017.

[3]. Esteva A, Robicquet A, Ramsundar B, Kuleshov V, DePristo M, Chou K, Cui C, Corrado G, Thrun S, and Dean J, “A guide to deep learning in healthcare,” Nature medicine, vol. 25, no. 1, pp. 24–29, 2019.

[4]. Asgari M, Kaye J, and Dodge H, “Predicting mild cognitive impairment from spontaneous spoken utterances,” Alzheimer’s & Dementia: Translational Research & Clinical Interventions, vol. 3, no. 2, pp. 219–228,2017

[5]. Nemirovskij AS and Yudin DB, “Problem complexity and method efficiency in optimization,” 1983.

[6]. Vu T-H, Jain H, Bucher M, Cord M, and Pérez P, “Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2517–2526.

[7]. Baltrušaitis T, Mahmoud M, and Robinson P, “Cross-dataset learning and person-specific normalisation for automatic action unit detection,” in 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), vol. 6. IEEE, 2015, pp. 1–6.

[8]. Saito K, Kim D, Sclaroff S, Darrell T, and Saenko K, “Semi-supervised domain adaptation via minimax entropy,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 8050-8058

[9]. Tzeng E, Hoffman J, Zhang N, Saenko K, and Darrell T, “Deep domain confusion: Maximizing for domain invariance,” arXiv preprint arXiv:1412.3474, 2014

[10]. Tzeng E, Hoffman J, Saenko K, and Darrell T, “Adversarial discriminative domain adaptation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7167–7176.

[11]. Peng X, Bai Q, Xia X, Huang Z, Saenko K, and Wang B, “Moment matching for multi-source domain adaptation,” in Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 1406–1415.

[12]. Li C-L, Chang W-C, Cheng Y, Yang Y, and Póczos B, “Mmd gan: Towards deeper understanding of moment matching network,” Advances in neural information processing systems, vol. 30, 2017.

[13]. Hoffman J, Tzeng E, Park T, Zhu J-Y, Isola P, Saenko K, Efros A, and Darrell T, “Cycada: Cycle-consistent adversarial domain adaptation,” in International conference on machine learning. PMLR, 2018, pp. 1989–1998.

[14]. Turner A, Tsipras D, and Madry A, “Clean-label backdoor attacks,” 2018.

[15]. Levine A. and Feizi S, “Deep partition aggregation: Provable defense against general poisoning attacks,” arXiv preprint arXiv:2006.14768, 2020.

[16]. Wang Q, Michau G, and Fink O, “Missing-class-robust domain adaptation by unilateral alignment,” IEEE Transactions on Industrial Electronics, vol. 68, no. 1, pp. 663–671, 2020.

[17]. Yu X, Liu T, Gong M, Zhang K, Batmanghelich K, and Tao D, “Label-noise robust domain adaptation,” in International Conference on Machine Learning. PMLR, 2020, pp. 10913–10924.

[18]. Yu Q, Hashimoto A, and Ushiku Y, “Divergence optimization for noisy universal domain adaptation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 2515-2524.

[19]. Chen M, Xu Z, Weinberger K, and Sha F, “Marginalized denoising autoencoders for domain adaptation,” arXiv preprint arXiv:1206.4683, 2012.

[20]. Han Z, Gui X-J, Cui C, and Yin Y, “Towards accurate and robust domain adaptation under noisy environments,” arXiv preprint arXiv:2004.12529, 2020.

[21]. Lugosi G. and Mendelson S, “Risk minimization by median-of-means tournaments,” Journal of the European Mathematical Society, vol. 22, no. 3, pp. 925–965, 2019.

[22]. Prasad A, Balakrishnan S, and Ravikumar P, “A robust univariate mean estimator is all you need,” in International Conference on Artificial Intelligence and Statistics. PMLR, 2020, pp. 4034–4044

[23]. Lecué G. and Lerasle M, “Robust machine learning by median-of-means: theory and practice,” The Annals of Statistics, vol. 48, no. 2, pp. 906–931, 2020.

[24]. Devroye L, Lerasle M, Lugosi G, and Oliveira RI, “Sub-gaussian mean estimators,” The Annals of Statistics, vol. 44, no. 6, pp. 2695–2725, 2016.

[25]. Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, and Lempitsky V, “Domain-adversarial training of neural networks,” The journal of machine learning research, vol. 17, no. 1, pp. 2096–2030, 2016.

[26]. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, and Bengio Y, “Generative adversarial nets,” Advances in neural information processing systems, vol. 27, 2014.

[27]. Liang J, Hu D, and Feng J, “Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation,” in International Conference on Machine Learning. PMLR, 2020, pp. 6028–6039.

[28]. LeCun Y, Cortes C, and Burges C, “Mnist handwritten digit database,” 2010.

[29]. Asuncion A. and Newman D, “Uci machine learning repository,” 2007

[30]. Krizhevsky A, Hinton G. et al., “Learning multiple layers of features from tiny images,” 2009.

[31]. French G, Mackiewicz M, and Fisher M, “Self-ensembling for visual domain adaptation,” arXiv preprint arXiv:1706.05208, 2017.

[32]. Long M, Cao Z, Wang J, and Jordan MI, “Conditional adversarial domain adaptation,” Advances in neural information processing systems, vol. 31, 2018.

[33]. Gu T, Dolan-Gavitt B, and Garg S, “Badnets: Identifying vulnerabilities in the machine learning model supply chain,” arXiv preprint arXiv:1708.06733, 2017

[34]. LeCun Y, Bottou L, Bengio Y, and Haffner P, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

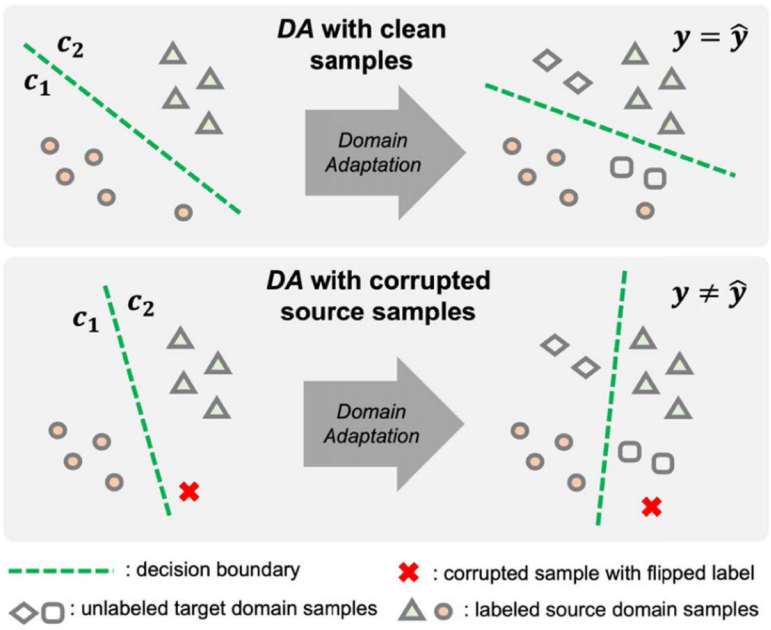


Fig. 1:

Source domain data corruption may lead to failure in many existing domain adaptation approaches.

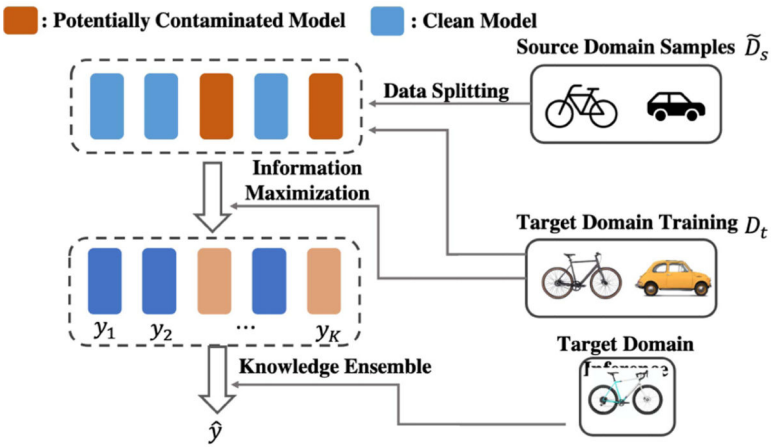
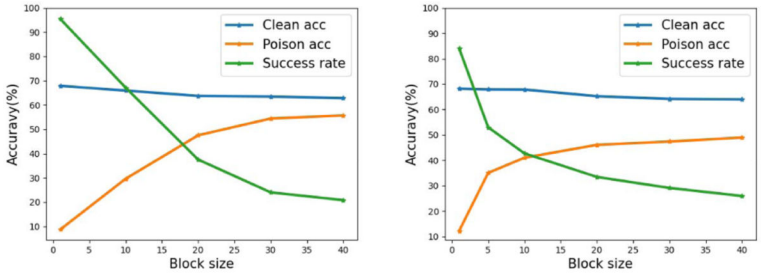


Fig. 2:
Process of robust domain adaptation learning.



(a) *BadNet* attacks using DANN (b) *CLBD* attacks using DANN

Fig. 3:

Clean test accuracy, poison test accuracy and attack success rate for MOM *w.r.t.* different block number. Increasing the number of blocks within a certain range is beneficial for improving the performance.

TABLE I:

Accuracy(%) and attack success rates(%) for MoM using under *BadNet* attacks. ↑ indicates that a larger value is desirable, and vice versa. **Bold** numbers are best performers. Bnum indicates block number. By applying MoM, we can significantly bring down the attack success rate and also improve the target poison test accuracy, while maintaining the target clean sample accuracy.

DA	Task	Poison ratio	Bnum	Clean acc ↑	Poison acc ↑	Success rate ↓
DANN	M→U	0 (clean)	1	88.89	11.26	9.62
		0.01	1	88.79	8.57	92.33
			10	86.25	**12.21**	**8.77**
		0.02	1	89.34	8.77	97.06
			15	83.86	**11.61**	**28.65**
		0.03	1	88.44	8.62	95.17
			20	82.76	**11.21**	**35.87**
	U→M	0 (clean)	1	95.54	9.56	9.89
		0.01	1	95.38	10.02	94.89
			10	85.00	**10.24**	**12.88**
		0.02	1	93.30	10.21	97.82
			10	83.22	**10.50**	**18.09**
		0.03	1	95.02	10.10	99.12
			20	75.31	**10.57**	**18.87**
CDAN	M→U	0 (clean)	1	93.47	12.41	9.57
		0.01	1	93.52	8.52	94.27
			10	87.84	**12.76**	**9.97**
		0.02	1	94.07	8.57	97.46
			15	85.45	**12.01**	**19.18**
		0.03	1	94.02	8.82	99.15
			20	82.71	**10.96**	**38.22**
	U→M	0 (clean)	1	92.96	9.68	10.04
		0.01	1	96.29	10.17	93.49
			10	83.09	**10.48**	**15.62**
		0.02	1	93.4	10.1	97.27
			15	77.71	**10.56**	**17.51**
		0.03	1	97.22	10.13	98.39
			20	74.39	**10.53**	**20.76**

TABLE II:

Accuracy(%) and attack success rates(%) for digit task. Our adaptation method is consistently robust given different DA algorithms. IM and PL are verified to be effective to not only further decrease attack success rate but also increase the target poison test accuracy.

DA model	Task	Clean acc ↑	Poison acc ↑	Success rate ↓	Adaptation
DANN	M→U	82.76	11.21	35.87	/
		80.67	11.71	19.03	IM
		81.22	**11.96**	**16.49**	IM+PL
	U→M	75.31	10.57	18.87	/
		78.95	10.62	10.11	IM
		79.46	**10.76**	**9.89**	IM+PL
CDAN	M→U	82.71	10.96	38.22	/
		81.42	**12.51**	**7.08**	IM
		81.81	12.16	10.66	IM+PL
	U→M	74.39	10.53	20.76	/
		78.90	10.50	11.58	IM
		80.67	**10.56**	**10.60**	IM+PL

TABLE III:

Accuracy(%) and attack success rates(%) using base approach DANN for the task CIFAR10 → STL under *BadNet* and *CLBD* attack. Our adaptation method is consistently robust given different kinds of tasks and corruptions.

Attack	Clean acc ↑	Poison acc ↑	Success rate ↓	Adaptation
	62.00	55.76	19.28	/
BadNet	62.00	**56.76**	**16.64**	IM
	60.85	50.79	18.38	IM+PL
	63.98	48.97	25.96	/
CLBD	61.99	50.38	**23.45**	IM
	62.41	**50.71**	24.37	IM+PL