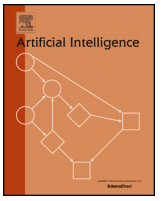
Artificial Intelligence 347 (2025) 104383

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint



Adversarially robust unsupervised domain adaptation

Lianghe Shi, Weiwei Liu,*

School of Computer Science, Wuhan University, Bayi Road, 430072, Wuhan, China

ARTICLE INFO

Keywords:
Adversarial robustness
Unsupervised domain adaptation
Generalization error bound
Rademacher complexity

ABSTRACT

Unsupervised domain adaptation (UDA) has been successfully applied in many contexts with domain shifts. However, we find that existing UDA methods are vulnerable to adversarial attacks. A direct modification of the existing UDA methods to improve adversarial robustness is to feed the algorithms with adversarial source examples. However, empirical results show that traditional discrepancy fails to measure the distance between adversarial examples, leading to poor alignment between adversarial examples of source and target domains and inefficient transfer of the robustness from source domain to target domain. And the traditional theoretical bounds do not always hold in adversarial scenarios. Accordingly, we first propose a novel adversarial discrepancy (AD) to narrow the gap between adversarial robustness and UDA. Based on AD, this paper provides a generalization error bound for adversarially robust unsupervised domain adaptation through the lens of Rademacher complexity, theoretically demonstrating that the expected adversarial target error can be bounded by empirical adversarial source error and AD. We also present the upper bounds of Rademacher complexity, with a particular focus on linear models and multi-layer neural networks under ℓ_r attack ($r \geq 1$). Inspired by this theory, we go on to develop an adversarially robust algorithm for UDA. We further conduct comprehensive experiments to support our theory and validate the robustness improvement of our proposed method on challenging domain adaptation tasks.

1. Introduction

The key assumption of classical machine learning—that training and test data come from the same distribution—may not always hold in many real-world applications [17]. For example, one may collect the training and test data from different domains, while distributional shifts in covariates can lead to changes in the data-generating distribution [54,7]. To address this problem, unsupervised domain adaptation (UDA) has been developed to train a model that performs well on an unlabeled target domain by leveraging labeled data from a similar yet distinct source domain.

Remarkable algorithmic advances [25,69,12,41] of UDA show that the generalization ability of a model can be transferred between different domains. However, little theoretical research has been done to explore the adversarial robustness of UDA, despite its vital role in security-critical scenarios. Adversarial robustness refers to the invariance of a model to small perturbations of its input [47], while adversarial accuracy refers to a model's prediction accuracy on adversarial examples generated by an attacker. The surprising data in Table 1 shows that existing vanilla UDA methods obtain 0% adversarial accuracy on task $A \to D$ of Office-31 [44] under PGD-20 attack [35] with ℓ_∞-norm-bounded perturbation of radius $\epsilon = 3/255$.

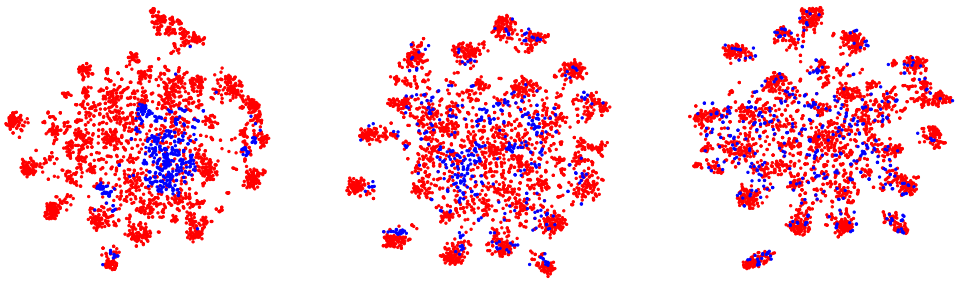
* Corresponding author.
E-mail addresses: shilianghe007@gmail.com (L. Shi), weiwei.liu@whu.edu.cn (W. Liu).

https://doi.org/10.1016/j.artint.2025.104383
Received 23 October 2023; Received in revised form 2 June 2025; Accepted 7 June 2025

Available online 13 June 2025

Table 1
Results (%) of UDA methods on task $A \to D$ of Office-31 [44]. $\mathcal{A}_{cle}$ denotes the accuracy on clean examples, while $\mathcal{A}_{adv}$ indicates the accuracy on adversarially perturbed examples. All methods are based on ResNet-18 [22]. The adversarial examples are generated by PGD-20 attack [35] with ℓ_∞-norm bounded perturbation of radius $\varepsilon = 3/255$. DANN stands for Domain-Adversarial Neural Network [18]. MDD stands for Margin Disparity Discrepancy [69]. BSP stands for Batch Spectral Penalization [12]. MCC stands for Minimum Class Confusion [25]. MCD stands for Maximum Classifier Discrepancy [46]. DIRT-T stands for Decision-boundary Iterative Refinement Training with a Teacher [55].

Method	MCD	DIRT-T	DANN	MDD	BSP	MCC
$\mathcal{A}_{cle}$	75.9	78.3	78.5	79.1	79.7	91.2
$\mathcal{A}_{adv}$	0.2	0.0	0.0	0.0	0.0	0.0



(a) Source adversarial (b) Source adversarial & $d_{H\triangle H}$ (c) Source adversarial & AD (Ours)

Fig. 1. The visualization of embedded features on the task A→D, red and blue points denote the source and target domains, respectively. Best viewed in color. (a) shows the result of standard adversarial training with only source data; (b) shows the result of DANN trained with adversarial source examples; (c) shows the result of our method. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

To gain robustness, a popular and useful strategy [20] is adversarial training, which adds adversarial examples to the training data. In many prior works of UDA [18,69,12], researchers minimize both source error and discrepancy through an adversarial network. Hence, a direct modification of these existing UDA methods is to feed the learner with adversarial source examples generated corresponding to the classifier. However, the traditional discrepancies measure the distance between the clean examples of source and target domains, we empirically find that they are no longer appropriate metrics in the robustness setting. In Fig. 1(b), the t-SNE [34] graphs show that optimizing traditional discrepancy leads to poor alignment between adversarial examples of source and target domains. The poor alignment leads to inefficient transfer of the robustness from the source domain to the target domain. We also provide a toy example (Example 4.1) to theoretically show the traditional generalization bounds do not always hold in adversarial robustness scenarios, which suggests we need a new discrepancy metric.

Accordingly, this paper first introduces a novel adversarial discrepancy (AD) with nice theoretical properties, bridging the gap between adversarial robustness and UDA. Based on AD, we derive a learning bound for adversarially robust unsupervised domain adaptation (ARUDA) based on Rademacher complexity. Moreover, we present concrete bounds of Rademacher complexity for the linear hypotheses and multi-layer neural networks. Our theoretical results show that the expected adversarial target error can be upper bounded by the empirical adversarial source error and AD, which means the adversarial robustness can also be transferred between different domains. Hence, we can minimize the empirical adversarial source error and AD to train a robust classifier on the target domain. Motivated by our theoretical results, we propose an adversarially robust representation learning algorithm (illustrated in Fig. 3) for ARUDA to minimize the bound. Our method evidently boosts the adversarial robustness on the relatively difficult tasks D → A of Office-31, where our method improves the adversarial accuracy of baselines by 19.6% under PGD-20 attack with ℓ_∞-norm bounded perturbation of radius $\varepsilon = 3/255$. Extensive numerical experiments verify our theoretical bounds and show that the robustness of the model can be transferred from the source domain to the target domain.

Contributions. We summarize our main contributions as follows:

- We show that the existing UDA methods are vulnerable to adversarial attacks. To bridge the gap between adversarial robustness and UDA theory, we introduce a novel adversarial discrepancy metric (AD) for distributions and provide a generalization error bound for ARUDA through the lens of Rademacher complexity.
- We provide novel upper bounds of Rademacher complexity for linear hypothesis classes and **multi-layer** neural networks.
- Motivated by the theoretical results, we propose an adversarially robust representation learning algorithm for ARUDA to gain robustness on the target domain.

- We conduct extensive experiments on popular datasets and verify our theoretical bounds.

2. Related work

Unsupervised Domain Adaptation. [6] has shown one of the pioneering theoretical works in UDA. They introduce the $H \bigtriangleup H$-divergence, which is a substitution of traditional distribution discrepancy and can be estimated from finite samples, to characterize the difference between two domains. [36] further extends the divergence to general loss functions. Based on the theory presented in [6], [18] introduce a popular baseline domain-adversarial neural network (DANN) that achieves promising performance on UDA tasks. DANN aims to learn the features that are discriminative for the learning task on the source domain and indiscriminate with respect to the shift between domains. Following this line, [57] proposes an architecture that employs asymmetric encodings for target and source data. [46] utilizes two classification heads to simultaneously make predictions for the samples. And the domain gap between the source domain and target domain is measured by the discrepancy of the prediction results of these two classifiers on the target domain. [55] proposes the Virtual Adversarial Domain Adaptation (VADA) model, which combines domain adversarial training with a penalty term that punishes violation of the cluster assumption. [48] unifies pixel-level and feature-level adversarial learning for domain adaptation. [46] considers the classifier instead of the features and proposes an original adversarial learning method by maximizing the classifier discrepancy. [12] presents a general approach based on singular values to improve both transferability and discriminability. [41] develops a robust contrastive prototype adaptation strategy to align each pseudo-labeled target data point. [60] proposes to use depth distribution density to support semantic segmentation and further improve the performance of the prediction model. However, none of these works consider the adversarial robustness of UDA.

[23] and [64] both focus on improving clean accuracy using adversarial samples, while our method aims to improve adversarial robustness. [2] proposes a UDA method to improve adversarial robustness. However, their model depends on ImageNet adversarially pre-trained models and gains robustness by instilling robustness from the pre-trained models, while our proposed method does not use any adversarially pre-trained model. We have different settings compared to [2]. The work of [52] investigates the adversarial robustness of gradual domain adaptation, where the theory is based on the classical domain discrepancy, while we propose the novel adversarial discrepancy for robust domain adaptation. [72,71] analyzes the adversarial robustness in OOD generalization and contrastive learning models. Another work related to ARUDA is [31] where the authors propose a series of methods called SSAT. SSAT generates the adversarial examples by maximizing the self-supervised loss on the target domain and then minimizing the discrepancy between clean source examples and adversarial target examples. Compared to [2,31], our work provides theoretical analysis for ARUDA and develops a theoretically motivated discrepancy to align the domains under adversarial attack.

Adversarial Robustness. [56] shows that DNNs are fragile to adversarial attacks. [20] observes that adding adversarial examples into the training datasets improves adversarial robustness, and the strategy is called adversarial training. An explicit formulation of the min-max optimization appears in [24] and [51]. Following this line of research, [35,37,28,9] propose iterative variants of the gradient attack with improved adversarial learning frameworks. One notable work is TRADES [68], which balances the trade-off between standard and robust accuracy. On the other hand, some works focus on the sample complexity and generalization of adversarial training. [53] analyzes the curriculum adversarial training algorithm from the perspective of domain discrepancy. [15] demonstrates sample complexity bounds on PAC-learning in the presence of an evasion adversary. [33] theoretically investigates the tradeoff between robustness and fairness of the model. [49] shows that although in a simple natural data model such as the Gaussian model, the sample complexity of adversarial training can be significantly larger than that of "standard" training. They postulate that the difficulty of training robust classifiers stems from this inherently larger sample complexity. What's more, the sample complexity is studied in many works to give a generalization bound for adversarial training [61,38,19,70,63].

Adversarial Rademacher Complexity. [65] introduces the adversarial Rademacher complexity to measure the robust generalization gap. They derive the upper bounds for linear hypothesis class and two-layer neural networks. However, the proofs cannot be extended to multi-layer cases. [27] uses a method called tree transform to derive the adversarial Rademacher complexity upper bound and [65] uses the SDP relaxation proposed by [42]. [19,62] studies the adversarial Rademacher complexity for deep neural networks, but their analysis is based on standard adversarial attack setting [20], while we extend the adversarial Rademacher complexity bound to the domain adaptation problem.

Robustness Transfer. [50] discusses how robustness is transferred in transfer learning, demonstrating that the target model can inherit the robustness from an adversarially pre-trained source model. One recent work [11] further fine-tunes multiple layers to inherit the robustness of the source model. [13] studies the transferability of the adversarial attacks, showing that the black-box adversarial examples targeting one network can also successfully attack the other networks. Nevertheless, the adversarial robustness of UDA is poorly understood. Accordingly, this paper systematically studies ARUDA from both theoretical and algorithmic perspectives.

3. Preliminaries

3.1. Unsupervised domain adaptation

A domain can be envisioned as a tuple $\langle \mathcal{D}, f \rangle$ [6], consisting of a distribution $\mathcal{D}$ on input space $\mathcal{X}$ and a labeling function $f : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{Y}$ denotes the output or label space. The input space $\mathcal{X}$ is a subset of a d-dimensional space, $\mathcal{X} \subset \mathbb{R}^d$. The label space is $\{0, 1\}$ in binary classification, $\{1, ..., k\}$ in multiclass classification, and some measurable subset of $\mathbb{R}$ in regression.

There are two domains in UDA: the source domain and the target domain [36,17]. We use $\langle P, f_P \rangle$ to denote the source domain and $\langle Q, f_Q \rangle$ for the target domain. Following [6], we assume that the learner is provided with source samples $S = \{(x_i^s)\}_{i=1}^m$ drawn

independent and identically distributed (i.i.d.) according to the source distribution P with labels $\{f_P(x_i^s)\}_{i=1}^m$ and target samples $\mathcal{T} = \{x_i^t\}_{i=1}^n$ drawn i.i.d. according to the target distribution Q. We use $\widehat{P}$ and $\widehat{Q}$ to denote the corresponding empirical distribution for samples S and $\mathcal{T}$ respectively. Let X denote the $d \times m$ matrix with data point x_i as the column. The (p,q)-*group norm* of a matrix M is defined as $\|M\|_{p,q} = \|(\|M_1\|_p, \ldots, \|M_n\|_p)\|_q$, where M_i, $i \in \{1, \cdots, n\}$ is the i-th column of matrix M. For a real number $p \geq 1$, the p-norm or ℓ_p-norm of x is defined by $\|x\|_p = (|x_1|^p + |x_2|^p + \ldots + |x_n|^p)^{1/p}$.

A hypothesis is defined as a function $h : \mathcal{X} \to \mathcal{Y}$. We use $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ to denote a loss function and $\epsilon_D(h, f)$ to indicate the expected error for any two functions $h, f : \mathcal{X} \to \mathcal{Y}$ and any distribution D over $\mathcal{X}$:

$$\epsilon_D(h, f) = \mathop{\mathbb{E}}_{x \sim D} [\ell(h(x), f(x))] .$$

3.2. Adversarial error

We use $N(x)$ to represent a neighborhood of x. This paper focuses on the set $N(x) = \{a \in \mathcal{X} : \|a - x\|_r \leq \varepsilon, r \geq 1\}$, which is called ℓ_r-attack in AT. Similar to the definition of standard error $\epsilon_D(h, f_D)$, we define the expected adversarial error for any two functions $h, f : \mathcal{X} \to \mathcal{Y}$ and any distribution D over $\mathcal{X}$ as:

$$\epsilon_D^{adv}(h, f) = \mathop{\mathbb{E}}_{x \sim D} \left[\sup_{a \in N(x)} \ell(h(a), f(x)) \right] .$$

The adversarial source error of a hypothesis is $\epsilon_P^{adv}(h, f_P)$, while the empirical adversarial source error is $\widehat{\epsilon}_P^{adv}(h, f_P)$. We use the parallel notations $\epsilon_Q^{adv}(h, f_Q)$, $\widehat{\epsilon}_Q^{adv}(h, f_Q)$ for the target domain.

The ARUDA problem aims to find a hypothesis from the hypothesis set H with a small expected adversarial error $\epsilon_Q^{adv}(h, f_Q)$ according to the target distribution Q. To train a robust model with resistance to the adversarial examples, AT aims to find a hypothesis h that yields the minimal adversarial error:

$$\min_{h \in H} \mathop{\mathbb{E}}_{x \sim Q} \left[\sup_{a \in N(x)} \ell(h(a), f(x)) \right] .$$

However, AT can not be directly applied to the target domain of UDA, since it requires labeled data.

3.3. Rademacher complexity

Rademacher complexity is used to measure the complexity of a hypothesis set [59]. Here we present the definition.

Definition 3.1. (Rademacher Complexity) Let H be a set of real-valued functions defined over a set $\mathcal{X}$. For any fixed collection of points $X := (x_1, ..., x_m)$, the empirical Rademacher complexity of H is given by:

$$\widehat{\mathcal{R}}_X(H) = \frac{2}{m} \mathop{\mathbb{E}}_{\sigma} \left[\sup_{h \in H} \sum_{i=1}^{m} \sigma_i h(x_i) \right] . \tag{1}$$

The expectation is taken over $\sigma = (\sigma_1, ..., \sigma_m)$, where σ_i, $i \in \{1, ..., m\}$ is an independent uniform random variable taking values in $\{-1, +1\}$.

The following lemma [4] uses the Rademacher complexity to connect the population and empirical error.

Lemma 3.2. ***(Rademacher Bound)*** *Suppose that $\mathcal{K}$ is a class of functions mapping $\mathcal{X}$ to [0,1]. Then, for any $\alpha > 0$, with probability at least $1 - \alpha$ over samples X of size m, the following holds for all $k \in \mathcal{K}$:*

$$\mathbb{E}_{x \sim D}\, k(x) \leq \mathbb{E}_{x \sim \widehat{D}}\, k(x) + \widehat{\mathcal{R}}_X(\mathcal{K}) + 3\sqrt{\frac{\log \frac{2}{\alpha}}{2m}}, \tag{2}$$

where $\mathbb{E}_{x \sim D}\, k(x)$ is the expectation of a function k, while $\mathbb{E}_{x \sim \widehat{D}}\, k(x)$ is its empirical average over the samples X drawn according to the empirical distribution $\widehat{D}$.

4. Theoretical analysis for ARUDA

The learner of UDA is trained with labeled data only on the source domain. To obtain generalization ability in the target domain, the distributions of P and Q should be similar. Thus, the measurement of distance between distributions is a key technical tool in generalization error bound for UDA. A straightforward divergence for distribution is the variation distance:

$$d_1(P, Q) = 2 \sup_{B \in \mathfrak{B}} |P[B] - Q[B]|,$$

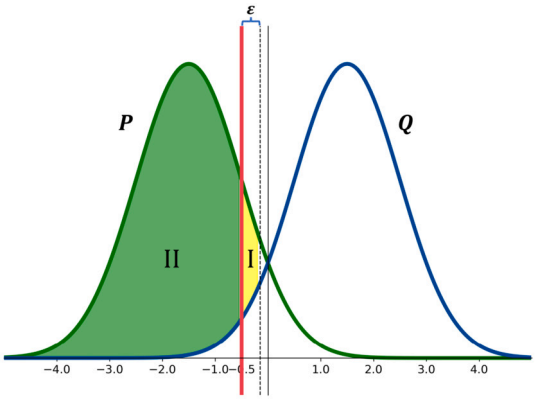


Fig. 2. The illustration of the toy example of Gaussian model. We use the curves in green and blue to represent the source and target distributions respectively. Best viewed in color.

where $\mathfrak{B}$ is the set of measurable subsets under P and Q. The disadvantage of variation divergence is that it cannot be accurately estimated from finite samples. To overcome this shortcoming, [6] proposes the $H \triangle H$ divergence to measure the discrepancy:

$$d_{H\triangle H}(P,Q) = \sup_{h,h'\in H} |\mathop{\mathbb{E}}_{x\sim P} \mathbb{1}[h(x)\neq h'(x)] - \mathop{\mathbb{E}}_{x\sim Q} \mathbb{1}[h(x)\neq h'(x)]|,$$

where $\mathbb{1}$ is the indicator function: the expression $\mathbb{1}[h(x)\neq h'(x)]$ evaluates to 1 if $h(x)\neq h'(x)$ and to 0 otherwise. Based on this divergence, [6] provides the following generalization error bound:

$$\forall h\in H,\ \ \epsilon_Q(h) \le \epsilon_P(h) + d_{H\triangle H}(P,Q) + \min_{h^*\in H}\{\epsilon_P(h^*)+\epsilon_Q(h^*)\}. \tag{3}$$

Following [6], many works propose various discrepancy measurements [69,26,21] and provide generalization error bounds for UDA. Although these theories have been influential in advancing algorithm designs, a gap remains between adversarial robustness and UDA. We next present a toy example of the Gaussian model to illustrate the gap.

Example 4.1 *(Gaussian model).* We consider the input space $\mathcal{X}=\mathbb{R}$ and the label space $\mathcal{Y}=\{-1,+1\}$. The source distribution P is a Gaussian distribution over $\mathcal{X}$: $x\sim\mathcal{N}(-\mu,\sigma^2)$, and the target distribution Q is symmetrical to P: $x\sim\mathcal{N}(+\mu,\sigma^2)$, where $\mu>0$. We consider a threshold hypothesis class, $H \triangleq \{x\mapsto sgn(x-b-\frac{1}{2}) : b\in Z\}$, where Z denotes the set of integers and sgn is the sign function. For simplicity, we assume that the labeling functions of the source domain and the target domain are the same, i.e., the covariate shift setting. And the thresholds of the labeling functions approach to $-\infty$, i.e., $b_P=b_Q=-\infty$. We use 0-1 loss as the loss function and assume the perturbation radius $\epsilon\le 0.5$. Then, we can find a hypothesis $h\in H$ with $b_h=-1$ such that:

$$\underbrace{\epsilon_Q^{adv}(h)-\epsilon_P^{adv}(h)}_{A} > \underbrace{d_{H\triangle H}(P,Q)}_{B} + \underbrace{\min_{h^*\in H}\{\epsilon_P^{adv}(h^*)+\epsilon_Q^{adv}(h^*)\}}_{C}, \tag{4}$$

which illustrates that the bound of (3) may not hold under adversarial robustness scenarios. We visualize this example in Fig. 2. The first term A in (4) equals the area of $I+II$ in Fig. 2, while the second term B equals the area of II and the third term C equals 0. The detailed calculation can be found in Appendix B.1.

4.1. Adversarial discrepancy

To fill the gap between adversarial robustness and UDA, we propose a novel adversarial discrepancy to measure the distance between distributions in adversarial robustness scenarios.

Definition 4.2. (Adversarial Discrepancy, AD). Let $\ell : \mathcal{Y}\times\mathcal{Y}\to\mathbb{R}$ be a loss function over $\mathcal{Y}$. Given a hypothesis class H mapping $\mathcal{X}$ to $\mathcal{Y}$, the adversarial discrepancy $disc_{adv}$ between two distributions P and Q over $\mathcal{X}$ is:

$$disc_{adv}(P,Q) = \sup_{h,h'\in H}\left|\mathop{\mathbb{E}}_{x\sim P}\left[\sup_{a\in N(x)}\ell\left(h(a),h'(x)\right)\right] - \mathop{\mathbb{E}}_{x\sim Q}\left[\sup_{a\in N(x)}\ell\left(h(a),h'(x)\right)\right]\right|. \tag{5}$$

This definition is more suitable in adversarial scenarios, and we claim that AD is a pseudometric in that it satisfies all properties of a metric, except that there may exist pairs $P\neq Q$ satisfying $disc_{adv}(P,Q)=0$. The detailed proof can be found in Appendix B.3.

Moreover, AD offers an important advantage in that it can be estimated from finite samples. The following theorem shows that for a bounded loss function ℓ, AD between a distribution and its empirical distribution can be bounded in terms of the empirical Rademacher complexity. The proof is provided in Appendix B.4.

Proposition 4.3. *Let ℓ be a loss function bounded by $\mathcal{M} > 0$. Let P be a distribution over $\mathcal{X}$, while $\widehat{P}$ is the corresponding empirical distribution for samples $S = (x_1, ..., x_m)$. Then, for any $\alpha > 0$, with probability at least $1 - \alpha$ over samples S of size m drawn according to P, we have:*

$$disc_{adv}(P, \widehat{P}) \leq \widehat{\mathcal{R}}_S(\widetilde{L}_H) + 3\mathcal{M}\sqrt{\frac{\log\frac{2}{\alpha}}{2m}}, \tag{6}$$

where

$$\widetilde{L}_H \triangleq \{x \to \sup_{a \in N(x)} \ell(h(a), h'(x)) | h, h' \in H\}.$$

We can then claim that AD between two distributions can be estimated from finite samples, thereby resulting in fully data-dependent learning guarantees. The proof is attached in Appendix B.5.

Proposition 4.4. ***(Discrepancy Estimation Bound).*** *Let ℓ be a loss function bounded by $\mathcal{M} > 0$. Let P, Q be distributions over $\mathcal{X}$, while $\widehat{P}, \widehat{Q}$ are the corresponding empirical distributions for samples S and $\mathcal{T}$ respectively. Then, for any $\alpha > 0$, with probability at least $1 - 2\alpha$ over samples S of size m drawn according to P and samples $\mathcal{T}$ of size n drawn according to Q:*

$$disc_{adv}(P, Q) \leq disc_{adv}(\widehat{P}, \widehat{Q}) + \widehat{\mathcal{R}}_S(\widetilde{L}_H) + \widehat{\mathcal{R}}_{\mathcal{T}}(\widetilde{L}_H) + 3\mathcal{M}\left(\sqrt{\frac{\log\frac{2}{\alpha}}{2m}} + \sqrt{\frac{\log\frac{2}{\alpha}}{2n}}\right). \tag{7}$$

4.2. Generalization error bounds for ARUDA

The following theorem shows that the adversarial target error can be controlled by the adversarial source error and the proposed AD. The proof is presented in Appendix B.6.

Theorem 4.5. ***(Generalization Error Bound for ARUDA)*** *Let ℓ be a loss function that is symmetric and obeys the triangle inequality. Then, for any hypothesis $h \in H$, the following inequality holds:*

$$\epsilon_Q^{adv}(h, f_Q) \leq \epsilon_P^{adv}(h, f_P) + \lambda + disc_{adv}(P, Q), \tag{8}$$

where

$$\lambda = \epsilon_P^{adv}(h^*, f_P) + \epsilon_Q^{adv}(h^*, f_Q), \tag{9}$$

and

$$h^* = \underset{h \in H}{argmin}\ \epsilon_P^{adv}(h, f_P) + \epsilon_Q^{adv}(h, f_Q). \tag{10}$$

Remark 4.6. In this theorem, we prove that the adversarial target error is upper bounded by the adversarial source error, the combined adversarial error λ, and the proposed AD. **Accordingly, the adversarial robustness can be transferred between domains**, and AD has a substantial impact on the adversarially robust generalization performance of the target domain. As we show in Example 4.1, the traditional divergence is not suitable for the adversarial robustness scenario. In contrast, our proposed AD successfully fills this gap.

The bound of (8) is relative to the **expected** adversarial error and AD. Based on Proposition 4.4, we further derive an **empirical** version of the generalization error bound for ARUDA. The proof is provided in Appendix B.7.

Corollary 4.7. *Let ℓ be a loss function that is symmetric and obeys the triangle inequality. Then, for any hypothesis $h \in H$ and $\alpha > 0$, with probability at least $1 - 3\alpha$ over samples S of size m drawn according to P and samples $\mathcal{T}$ of size n drawn according to Q, the following inequality holds:*

$$\epsilon_Q^{adv}(h, f_Q) \leq \widehat{\epsilon}_P^{adv}(h, f_P) + \lambda + disc_{adv}(\widehat{P}, \widehat{Q}) + 2\widehat{\mathcal{R}}_S(\widetilde{L}_H) + \widehat{\mathcal{R}}_{\mathcal{T}}(\widetilde{L}_H) + 3\mathcal{M}(2\sqrt{\frac{\log\frac{2}{\alpha}}{2m}} + \sqrt{\frac{\log\frac{2}{\alpha}}{2n}}). \tag{11}$$

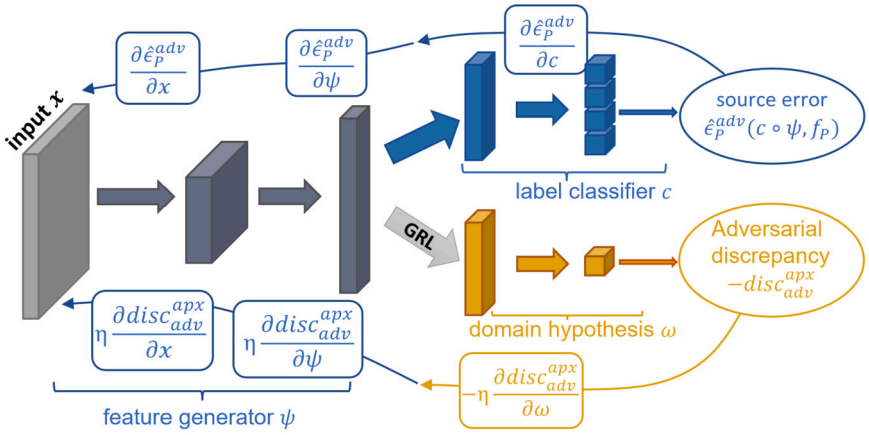


Fig. 3. The illustration of our proposed ARUDA algorithm. We use two curves and the gradients inside the rectangles to represent gradient backward propagation.

5. Algorithm

Based on Corollary 4.7, we propose a theoretically motivated ARUDA algorithm. Corollary 4.7 tells us that, the expected adversarial target error $\epsilon_Q^{adv}(h, f_Q)$ is bounded by the empirical adversarial source error $\hat{\epsilon}_P^{adv}(h, f_P)$ and $disc_{adv}(\widehat{P}, \widehat{Q})$. Hence, to find a classifier with a small adversarial target error in a given class, the algorithm should minimize the trade-off between $\hat{\epsilon}_P^{adv}(h, f_P)$ and $disc_{adv}(\widehat{P}, \widehat{Q})$, namely

$$\hat{\epsilon}_P^{adv}(h, f_P) + \eta \, disc_{adv}(\widehat{P}, \widehat{Q}), \tag{12}$$

where η is the trade-off parameter. Motivated by [18], strategy to control $disc_{adv}(\widehat{P}, \widehat{Q})$ is to find a representation of those examples in which the source and target domains are as indistinguishable as possible.

Formally, we use a robust feature generator $\psi : \mathbb{R}^d \to \mathbb{R}^e$ to learn a transformation from the original input space into a representation space. Based on the representation, a classifier $c \in \mathfrak{C}$ learns a function $c : \mathbb{R}^e \to \mathcal{Y}$, which predicts the output label of the input sample. Let $H = \{c \circ \psi | c \in \mathfrak{C}, \psi \in \Psi\}$ be the hypothesis class composed by feature generator and classifier, and $H_\psi = \{c \circ \psi | c \in \mathfrak{C}\}$ be the hypothesis class with a given feature generator $\psi \in \Psi$. Given a feature generator $\psi \in \Psi$, applying Corollary 4.7 to H_ψ, we derive that the expected adversarial target error $\epsilon_Q^{adv}(c \circ \psi, f_Q)$ can be controlled mainly by the empirical adversarial source error $\hat{\epsilon}_P^{adv}(c \circ \psi, f_P)$ and $disc_{adv}^{H_\psi}(\widehat{P}, \widehat{Q})$. Here, we use $disc_{adv}^{H_\psi}(\widehat{P}, \widehat{Q})$ to denote the AD corresponding to hypothesis class H_ψ:

$$\begin{aligned} disc_{adv}^{H_\psi}(\widehat{P}, \widehat{Q}) &= \sup_{c', c'' \in \mathfrak{C}} \left| \hat{\epsilon}_P^{adv}(c' \circ \psi, c'' \circ \psi) - \hat{\epsilon}_Q^{adv}(c' \circ \psi, c'' \circ \psi) \right| \\ &= \sup_{c', c'' \in \mathfrak{C}} \left| \mathbb{E}_{x \sim \widehat{P}} \left[\sup_{a \in N(x)} \ell \left(c' \circ \psi(a), c'' \circ \psi(x) \right) \right] - \mathbb{E}_{x \sim \widehat{Q}} \left[\sup_{a \in N(x)} \ell \left(c' \circ \psi(a), c'' \circ \psi(x) \right) \right] \right|, \end{aligned}$$

where c' and c'' play the role of discriminator [6]. Since our algorithm aims to train a feature generator ψ that minimizes the AD $disc_{adv}^{H_\psi}(\widehat{P}, \widehat{Q})$ and a classifier c that minimizes source error $\hat{\epsilon}_P^{adv}(c \circ \psi, f_P)$ based on the representation, the minimization problem (12) can be rewritten as:

$$\min_{c \in \mathfrak{C}, \psi \in \Psi} \hat{\epsilon}_P^{adv}(c \circ \psi, f_P) + \eta \, disc_{adv}^{H_\psi}(\widehat{P}, \widehat{Q}). \tag{13}$$

We empirically find that the training process is unstable due to the optimization of two separate networks c' and c''. To overcome this issue, we first present an approximation of AD:

$$disc_{adv}^{apx}(\widehat{P}, \widehat{Q}) = \sup_{c', c'' \in \mathfrak{C}} \left| \mathbb{E}_{x \sim \widehat{P}} \left[\sup_{a \in N(x)} \ell \left(c' \circ \psi(a), c'' \circ \psi(a) \right) \right] - \mathbb{E}_{x \sim \widehat{Q}} \left[\sup_{a \in N(x)} \ell \left(c' \circ \psi(a), c'' \circ \psi(a) \right) \right] \right|. \tag{14}$$

We have the following upper bound of the difference between AD and the approximated AD as below.

Proposition 5.1. *Let P and Q be any distributions over $\mathcal{X}$. Assume the loss function ℓ is L_l-Lipschitz and the hypothesis $c \circ \psi$ is L_c-Lipschitz. We have*

$$|disc_{adv}^{H_\psi}(P, Q) - disc_{adv}^{sur}(P, Q)| \leq 2\varepsilon L_l L_c.$$

Since P and Q are two arbitrary distributions over $\mathcal{X}$, we can substitute the empirical source and target distributions into the bound and use the result to extend our previous to the approximated AD. The detailed proof of Proposition 5.1 can be found in Appendix B.

Then we introduce a new hypothesis class Ω as below.

Definition 5.2 *(Hypothesis class Ω).* For a hypothesis class $\mathfrak{C}$, the hypothesis class Ω is the set of hypotheses:

$$\omega \in \Omega \iff \omega(x) = \ell(c'(x), c''(x)),$$

where c', c'' are some hypotheses in $\mathfrak{C}$.

By substituting ω into the approximation of AD (14), we obtain the following equivalent form:

$$disc_{adv}^{apx}(\widehat{P}, \widehat{Q}) = \sup_{\omega \in \Omega} \left| \frac{1}{m} \sum_{i=1}^{m} \left[\sup_{a_i^s \in N(x_i^s)} \omega(\psi(a_i^s)) \right] - \frac{1}{n} \sum_{i=1}^{n} \left[\sup_{a_i^t \in N(x_i^t)} \omega(\psi(a_i^t)) \right] \right|. \tag{15}$$

Algorithm 1 Calculation of Adversarial Discrepancy.

Require: the clean samples x_i^s, x_i^t; the fixed feature extractor ψ; the step size α, β
Ensure: adversarial discrepancy $disc_{adv}^{apx}(\widehat{P}, \widehat{Q})$
for $j = 1$ to m **do**
  randomly initialize Δx_i
  for $k = 1$ to n **do**
    $a_i = x_i + \Delta x_i$
    $\Delta x_i \leftarrow \Delta x_i + \alpha \nabla_{\Delta x_i} \omega(\psi(a_i))$
    $\Delta x_i \leftarrow \Delta x_i.clip(-\epsilon, \epsilon)$
  end for
  $a_i = x_i + \Delta x_i$
  $\omega \leftarrow \omega + \beta \nabla_w (\frac{1}{m} \sum_{i=1}^{m} \omega(\psi(a_i^s)) - \frac{1}{n} \sum_{i=1}^{n} \omega(\psi(a_i^t)))$
end for
return $\frac{1}{m} \sum_{i=1}^{m} \omega(\psi(a_i^s)) - \frac{1}{n} \sum_{i=1}^{n} \omega(\psi(a_i^t))$

The overall optimization problem can be rewritten as:

$$\min_{c \in \mathfrak{C}, \psi \in \Psi} \widehat{\epsilon}_P^{adv}(c \circ \psi, f_P) + \eta \, disc_{adv}^{apx}(\widehat{P}, \widehat{Q}). \tag{16}$$

As we can see in Eq. (16), there are two terms to be optimized. The first term is the source adversarial error $\widehat{\epsilon}_P^{adv}(c \circ \psi, f_P)$ which is used to train a robust source classifier. The second term $disc_{adv}^{apx}(\widehat{P}, \widehat{Q})$ robustly measures the distance between two empirical distributions $\widehat{P}$ and $\widehat{Q}$. Intuitively, the smaller the distance, the more indistinguishable the domains. The role of sup_y in (15) is to improve the robustness of the feature generator. Minimizing this term can align the source and target domains under adversarial attacks. The pseudo-code of calculating adversarial discrepancy in Eq. (15) can be found in Algorithm 1.

The problem in (16) contains the optimization of three hypotheses, ψ, c, and w, which motivates us to implement (16) in an adversarial network (see Fig. 3). The discriminator w is implemented by a branch neural network after the feature generator ψ. With the aid of the gradient reversal layer (GRL) [18], we can directly perform stochastic gradient descent (SGD) to maximize $disc_{adv}^{apx}$ over $\omega \in \Omega$ and minimize $\widehat{\epsilon}_P^{adv}(c \circ \psi, f_P) + \eta \, disc_{adv}^{apx}(\widehat{P}, \widehat{Q})$ over ψ and c.

6. Experiments

In this section, we show the robustness improvement of the proposed method and validate the theoretical bounds of ARUDA. We first present the setup of the experiments.

6.1. Setup

Datasets. Office-31 [44] is a standard domain adaptation dataset containing 31 object categories in three domains: Amazon (**A**), DSLR (**D**) and Webcam (**W**). There are a total of 4,652 images in 31 unbalanced classes. We evaluate all methods on all six transfer tasks. **Office-Home** [58] is a more difficult dataset than Office-31, comprising around 15,500 images in 65 classes from four domains: (**Ar**) artistic images in the form of sketches, paintings, etc.; (**Cl**) collection of clipart images; (**Pr**) images of objects without a background; and (**Rw**) images of objects captured with a regular camera. We evaluate all methods on all twelve transfer tasks. **VisDA-2017** [40] is a simulation-to-real dataset for domain adaptation which focuses on the simulation-to-reality shift with over 280,000 images across 12 categories. The training images are generated from the same object under different circumstances, while the validation images are collected from MSCOCO.

Table 2
Adversarial accuracy (%) of various methods on Office-31. All methods are based on ResNet-50. The adversarial examples are generated by PGD-20 attack with ℓ_∞-norm bounded perturbation of radius $\epsilon = 3/255$. **Avg.** denotes the average accuracy over all tasks. **The baseline methods are trained with adversarial examples**.

Method	A→D	A→W	D→W	D→A	W→A	W→D	Avg.
CPGA^{adv} [41]	7.6	12.3	21.9	7.8	17.8	51.2	19.8
BSP^{adv} [12]	31.7	36.0	80.8	20.0	35.0	93.0	49.4
MDD^{adv} [69]	26.9	43.8	86.5	16.5	33.1	95.8	50.4
RST	34.1	37.3	85.0	28.7	29.7	89.2	50.7
DANN^{adv} [18]	38.6	39.9	88.4	19.2	33.2	96.2	52.6
MCC^{adv} [25]	57.4	62.9	90.3	23.9	38.2	**98.6**	61.9
AD (ours)	**59.8**	**64.8**	**94.3**	**43.5**	**42.5**	98.2	**67.2**

Table 3
Adversarial accuracy (%) of various methods on Office-Home. All methods are based on ResNet-50. The adversarial examples are generated by PGD-20 attack with ℓ_∞-norm bounded perturbation of radius $\epsilon = 3/255$. Avg. denotes the average accuracy over all tasks. **The baseline methods are trained with adversarial examples.** *We use A, C, P and R to denote Ar, Cl, Pr and Rw respectively in this table.*

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg.
CPGA^{adv} [41]	23.2	22.8	29.7	13.1	29.5	24.9	11.9	27.3	29.5	19.1	30.3	39.0	25.0
MCC^{adv} [25]	25.9	28.8	32.8	13.2	34.3	25.5	13.0	25.5	32.3	24.2	34.8	48.4	28.2
BSP^{adv} [12]	25.7	25.6	32.8	15.7	34.3	29.0	15.0	28.0	34.3	25.5	38.3	46.8	29.3
DANN^{adv} [18]	28.1	25.2	33.9	17.3	36.1	31.3	14.3	30.8	37.1	27.7	39.8	48.8	30.9
MDD^{adv} [69]	26.0	30.0	38.5	14.3	36.9	28.4	13.4	29.7	36.3	29.7	38.1	53.1	31.2
RST	31.1	34.8	**41.6**	**22.7**	39.7	**36.8**	19.7	28.6	41.2	32.7	39.0	**55.0**	35.2
AD (ours)	**34.5**	**35.6**	40.2	**22.7**	**42.1**	34.9	**22.1**	**36.6**	**41.3**	**33.0**	**45.3**	54.7	**36.9**

UDA methods trained with adversarial source examples. The traditional UDA methods used in our experiments are: Domain-Adversarial Neural Network **(DANN)** [18], Margin Disparity Discrepancy **(MDD)** [69], Batch Spectral Penalization **(BSP)** [12], Minimum Class Confusion **(MCC)** [25], and Contrastive Prototype Generation and Adaptation **(CPGA)** [41]. Based on the original frameworks of these methods, **we extend these methods through feeding with adversarial source examples.** In the tables, we use a superscript ^{adv} to imply this modification. By comparing the performance between these adversarial UDA methods and the proposed method, we validate the superiority of the proposed adversarial discrepancy AD under the ARUDA setting.

Adversarial training based on pseudo labels. Recently, there has been a series of works utilizing unlabeled data to improve adversarial robustness [10,67,1], which first trains a model on labeled data, then predicts pseudo labels of unlabeled data, and finally adversarially fine-tunes the model on the pseudo-labeled data. In our experiments, we adopt this method as the baseline method RST.

Implementation details. We implement the methods using Pytorch [39] on two Nvidia GeForce RTX 3090 Ti GPUs. We adopt **ResNet-18**, **ResNet-50**, and **ResNet-101** [22] as feature generators ψ with parameters pre-trained on ImageNet [43]. The classifiers c and ω are randomly initialized three-layer neural networks with widths of 256 and 1024 respectively. Following the standard experimental protocol for UDA [32], we use all labeled source data and unlabeled target data in the training stage. The model is trained on 64-sized batches, where half of each batch is populated by samples from the source domain and the remainder are drawn from the target domain. Moreover, following the strategies developed for DANN [18], we use mini-batch SGD with momentum 0.9. The learning rates of the layers trained from scratch are set to be 10 times those of fine-tuned layers. We adopt a curriculum adversarial training [8] schedule that gradually increases the adversarial radius following a sine-based schedule over the training iterations. Following [35], we use PGD-20 with a single step size of 0.01 to obtain the adversarial examples.

6.2. Results

The results of methods using ResNet-50 on Office-31 are shown in Table 2. Results in **bold** indicate the best performance. From Table 2, we can see that our proposed method achieves the best adversarial accuracy on almost all transfer tasks. Compared with all baselines, our proposed method improves the adversarial accuracy by 5.3% on average over the six tasks of Office-31. It should further be noted that our method evidently boosts the adversarial robustness on the relatively difficult tasks D → A and W → A where the source domain is very small. For example, compared with the runner-up baseline, our method improves the adversarial accuracy by 14.8% on task D → A under PGD-20 attack with ℓ_∞-norm bounded perturbation of radius $\epsilon = 3/255$.

Table 3 presents the adversarial accuracy of all methods using ResNet-50 on the large-scale dataset Office-Home. As the table shows, our method achieves superior adversarial accuracy on most transfer tasks. For example, our proposed approach improves the adversarial accuracy of the runner-up baseline by 5.8% on task Pr → Cl under PGD-20 attack with ℓ_∞-norm bounded perturbation of radius $\epsilon = 3/255$. The results show that the robustness of the model can be transferred from the source domain to the target domain using appropriate strategies. Our algorithm considers the adapted discrepancy measure (AD) compared to the baseline methods and has superior performance than the baselines, which verify the excellence of the proposed AD.

Table 4
Adversarial accuracy (%) of various methods on VisDA-2017. All methods are based on ResNet-101.

Methods	SSAT	DIRT-T^{adv}	DANN^{adv}	MDD^{adv}	RST	MCD^{adv}	MCC^{adv}	AD (ours)
Acc.	8.0	8.1	12.7	13.3	21.2	25.9	26.4	**28.4**

Table 5
Adversarial accuracy (%) of various methods on Office-31. All methods are based on ResNet-18. The training and test adversarial examples are generated by PGD attack with ℓ_∞-norm bounded perturbation of radius $\varepsilon = 4/255$.

Method	A→D	A→W	D→A	D→W	W→A	W→D	Avg.
SSAT	2.0	0.0	2.0	23.3	4.0	36.3	11.3
MCD^{adv}	7.5	7.6	1.6	28.9	3.5	45.8	15.8
DIRT-T^{adv}	8.6	8.6	12.2	34.8	12.8	28.9	17.7
RST	20.5	28.6	15.9	29.4	18.2	54.4	27.8
MDD^{adv}	17.3	25.0	12.1	70.6	22.5	84.1	38.6
MCC^{adv}	17.7	22.8	17.4	74.3	20.1	85.5	39.7
DANN^{adv}	20.5	27.9	14.8	83.8	22.7	85.7	42.6
AD (ours)	**34.1**	**37.4**	**27.9**	**86.9**	**32.6**	**92.0**	**51.8**

Table 6
Adversarial accuracy (%) of various methods on Office-31. All methods are based on ResNet-18. The training and test adversarial examples are generated by PGD attack with ℓ_∞-norm bounded perturbation of radius $\varepsilon = 5/255$.

Method	A→D	A→W	D→A	D→W	W→A	W→D	Avg.
SSAT	0.0	0.0	2.0	2.3	0.2	16.7	3.5
MCC^{adv}	5.2	3.8	3.5	3.8	3.5	4.4	4.0
MCD^{adv}	6.0	7.3	0.9	17.6	1.0	31.0	10.6
RST	15.7	25.8	8.9	4.0	11.3	4.4	11.7
DIRT-T^{adv}	5.0	9.2	10.0	30.7	10.4	27.7	15.5
MDD^{adv}	8.4	21.1	12.5	69.1	17.7	71.5	33.4
DANN^{adv}	15.3	22.4	13.7	76.1	18.1	72.3	36.3
AD (ours)	**23.7**	**26.9**	**23.4**	**82.0**	**27.6**	**86.8**	**45.1**

We also conduct extensive experiments on VisDA, VLCS, and PACS datasets [30] to verify that our method consistently achieve better robustness compared to baseline methods. Table 4 shows the results of methods on the VisDA dataset using ResNet-101 as the backbone. The results on VLCS and PACS datasets can be found in Appendix C.

The method also performs well on other attack types and larger perturbation radii. In Table 5 and 6, we show the results on Office-31 corresponding to larger radii $\varepsilon = 4/255$ and $\varepsilon = 5/255$ respectively. As shown in the tables, our proposed method consistently achieves the stat-of-the-art robustness among the baseline methods. And the superority of our method becomes even more obvious on larger perturbation radius. For example, under the perturbation of radius $\varepsilon = 3/255$, the AD method achieves 57.0% adversarial accuracy in average, which is 116.6% of the accuracy of the runner-up method. When the perturbation of radius increases to $\varepsilon = 5/255$, the AD method achieves 45.1% which is 124.2% of the accuracy of the runner-up. We also note that some methods such as MCC^{adv} and RST degrade rapidly when the radius ε increases from $4/255$ to $5/255$. The reason is intuitive: when the adversarial attacker becomes stronger, the traditional methods using standard discrepancy measurement fail to align the source and target domains towards such a stronger and stronger attacker. On the contrary, our proposed method, with a robust discrepancy measurement, performs well against strong attackers.

To validate that our method also works on other attack types, we also provide the robustness against C&W attack. C&W attack [9] is an effective robustness metric that is widely adopted in extensive literature on adversarial robustness [68,5,14]. Table 7 shows the C&W adversarial robustness of the models trained under PGD attack with radius $\varepsilon = 3/255$. Although MCC^{adv} achieves better robustness in some tasks, our proposed method has the highest average robustness over all tasks. We also test the C&W adversarial accuracy of the models with training radius $\varepsilon = 4/255$ and $\varepsilon = 5/255$. The results are briefly summarized in Table 8, showing that our method consistently achieves better robustness compared to other methods.

6.3. Ablation study

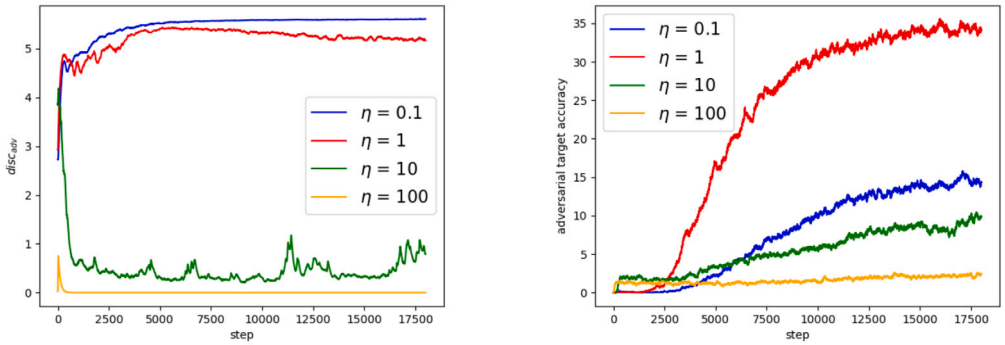
We further conduct an ablation study to assess the effect of each component of our proposed method. For comparison, we use various source errors and discrepancies. In Table 9, the first method presents the results of ResNet-18 trained with only clean source data. The second method uses DANN to train a model; DANN consists of source error and $H \bigtriangleup H$-divergence, which are not adversarially robust. The third method shows the results of standard adversarial training with only source data. Based on it, we add a normal $H \bigtriangleup H$-divergence regularization in the fourth method. The last method presents the proposed method.

Table 7
Adversarial accuracy (%) of various methods on Office-31. All methods are based on ResNet-18. The models are adversarially trained by PGD attack with radius $\epsilon = 3/255$ while the test adversarial examples are generated by C&W attack with ℓ_2-norm perturbation and trade-off parameter $c = 10^{-4}$.

Method	A→D	A→W	D→A	D→W	W→A	W→D	Avg.
SSAT	8.3	4.8	13.9	74.7	13.3	86.2	33.5
MCD^{adv}	25.2	15.9	8.3	78.5	10.8	83.0	37.0
RST	29.9	31.4	13.3	71.2	14.3	82.0	40.4
DIRT-T^{adv}	31.0	28.4	12.8	73.3	13.4	89.8	41.4
MDD^{adv}	31.3	35.2	11.0	76.3	13.1	87.0	42.3
DANN^{adv}	36.5	36.8	13.6	79.7	15.9	91.9	45.7
MCC^{adv}	43.8	37.2	14.1	**80.5**	15.7	**93.2**	47.4
AD (ours)	**45.3**	**44.7**	**14.6**	74.0	**18.3**	89.8	**47.8**

Table 8
Average adversarial accuracy (%) of various methods on six tasks of Office-31. All methods are based on ResNet-18. The models are adversarially trained by PGD attack with $\epsilon = \frac{4}{255}$ and $\frac{5}{255}$, while the test adversarial examples are generated by C&W attack with ℓ_2-norm perturbation and trade-off parameter $c = 10^{-4}$.

Tasks	MCD^{adv}	RST	SSAT	MDD^{adv}	DIRT-T^{adv}	MCC^{adv}	DANN^{adv}	AD
$\epsilon = \frac{4}{255}$	23.8	25.2	31.0	40.8	40.8	42.9	44.5	**47.0**
$\epsilon = \frac{5}{255}$	10.3	10.9	25.9	35.1	39.0	4.8	41.3	**45.7**



(a) Adversarial discrepancy (b) Adversarial target accuracy

Fig. 4. The changes of $disc^{apx}_{adv}$ and adversarial target accuracy of our method on the difficult transfer task D → A of Office-31 over steps. We use a step to represent a batch calculation. Best viewed in color.

Table 9
Ablation study on Office-31. The method is based on ResNet-18. $\mathcal{A}_{adv}$ denotes the accuracy on adversarially perturbed examples generated by PGD-20 attack with ℓ_∞-norm bounded perturbation of radius $\epsilon = 3/255$.

Source Discrepancy	clean -	clean $H \triangle H$	adversarial -	adversarial $H \triangle H$	adversarial AD
$\mathcal{A}_{adv}$	0.0	0.0	34.1	47.3	**56.7**

As shown in Table 9, adversarial training on the source domain enhances the model's robustness on the target domain, as can be concluded by comparing the first and third rows. The last three rows of Table 9 show that regularization is able to improve the model's robustness on the target domain and that our proposed AD regularization is more effective than $H \triangle H$-divergence regularization. These experimental results validate that minimizing the empirical adversarial source error and empirical AD boosts the adversarial robustness of UDA, which verifies our Corollary 4.7.

We visualize by t-SNE [34] in Fig. 1 the representations of task A→D (a difficult task with 31 classes) by ResNet-18. The source and target data are not aligned in (a), better aligned with a $H \triangle H$ divergence in (b) but still have an obvious mismatch on the center class. They are best aligned in (c) compared with (a) and (b), which shows the efficiency of our proposed adversarial discrepancy.

6.4. Sensitivity of regularization hyperparameter η

Fig. 4 visualizes the changes of $disc^{apx}_{adv}$ and the adversarial target accuracy of our method on the difficult transfer task D → A of Office-31 over steps. η is chosen from $\{0.1, 1, 10, 100\}$. As Fig. 4 (a) shows, our method is able to effectively control empirical AD.

Table 10
Results (%) of our method on Office-31 with varying η. The method is based on ResNet-18 and trained with adversarial examples generated by PGD-20 attack [35] with ℓ_∞-norm bounded perturbation of radius $\varepsilon = 3/255$. $\mathcal{A}_{adv}$ denotes the accuracy on adversarially perturbed examples generated by PGD-20 attack with ℓ_∞-norm bounded perturbation of radius $\varepsilon = 3/255$.

η	10^{-2}	10^{-1}	1	10^{1}	10^{2}	10^{3}	10^{4}	10^{5}
$\mathcal{A}_{adv}$	36.9	39.4	**56.7**	40.9	34.6	34.7	7.1	4.1

Moreover, as Fig. 4 (b) shows, when η increases, AD decreases while the adversarial target accuracy first rises and then falls. We also present the results of our proposed method on Office-31 with varying η in Table 10. The highest adversarial accuracies are achieved at $\eta = 1$. A suitable trade-off η is important, and we set $\eta = 1$ for baseline comparisons.

In fact, the hyper-parameter selection problem is an important question in domain adaptation literature. Since we don't have a labeled validation dataset in the unsupervised domain adaptation (UDA) setting, it's infeasible to do supervised validation. Fortunately, there is a series of effective work investigating how to do unsupervised parameter selection in UDA. [66] proposes Deep Embedded Validation (DEV), which embeds adapted feature representation into the validation procedure to obtain unbiased estimation of the target risk with bounded variance. [45] assumes that the source classifier should project target samples of the same class into a dense cluster in feature space and propose a novel unsupervised validation criterion that measures the density of soft neighborhoods. [16] selects the models by linear aggregation and provides theoretical error guarantees. All of these methods can be used to select η in our work.

7. Rademacher complexity for the concrete hypothesis class

In previous sections, although we have shown the generalization error bound for ARUDA, the Rademacher complexity $\widehat{\mathcal{R}}_S(\widetilde{L}_H)$ in Corollary 4.7 remains insufficiently explicit. In this section, we present the concrete bounds of Rademacher complexity for linear hypotheses and multi-layer neural networks. For linear hypotheses, we use Rademacher complexity $\widehat{\mathcal{R}}_S(H)$ with standard loss to bound $\widehat{\mathcal{R}}_S(\widetilde{L}_H)$. For neural networks, we use the technical tool of covering number [59] to bound $\widehat{\mathcal{R}}_S(\widetilde{L}_H)$. We first introduce the definition of covering number and some useful lemmas.

Definition 7.1 *(Covering Number [59]).* Given a metric space $(\mathbb{T}, \rho)$ where $\mathbb{T}$ is a non-empty set and $\rho(\cdot,\cdot)$ is a (pseudo)-metric. A δ-cover of $\mathbb{T}$ with respect to ρ is a set $\{\theta^1, ..., \theta^C\} \subset \mathbb{T}$ such that for each $\theta \in \mathbb{T}$, there exists some $i \in \{1, ..., C\}$ such that $\rho(\theta, \theta^i) \le \delta$. The δ-covering number $C(\delta; \mathbb{T}, \rho)$ is the cardinality of the smallest δ-cover. We use $C_X(\delta; \mathbb{T})$ as shorthand to denote $C(\delta; \mathbb{T}, \rho_X)$.

Accordingly, the covering number can measure the complexity of a non-empty set $\mathbb{T}$ with respect to ρ. The following lemma demonstrates how to bound the covering number of a ball in $\mathbb{R}^d$.

Lemma 7.2. *Let $\mathbb{B}_d^p(\tau)$ be the ℓ_p-norm ball in $\mathbb{R}^d$ with radius τ. The δ-covering number of $\mathbb{B}_d^p(\tau)$ with respect to the ℓ_p-norm obeys the following bound:*

$$C(\delta; \mathbb{B}_d^p(\tau), \|\cdot\|_p) \le \left(1 + \frac{2\tau}{\delta}\right)^d.$$

The proof can be found in Appendix B.8. We then provide a definition of sub-Gaussian process.

Definition 7.3 *(59, Definition 5.16, Sub-Gaussian Process).* A collection of zero-mean random variables $\{X_\theta, \theta \in \mathbb{T}\}$ is a sub-Gaussian process with respect to a metric ρ_X on $\mathbb{T}$ if:

$$\mathbb{E}[e^{\nu(X_\theta - X_{\tilde{\theta}})}] \le e^{\frac{\nu^2 \rho_X^2(\theta,\tilde{\theta})}{2}}, \ \forall \theta, \tilde{\theta} \in \mathbb{T}, \nu \in \mathbb{R}.$$

It can be easily proven that the Rademacher process [59] satisfies the condition in Definition 7.3 with respect to the ℓ_2-norm. The next Lemma derives an upper bound for the expected supremum of a sub-Gaussian process and relates the Rademacher complexity and covering number.

Lemma 7.4 *(Dudley's Entropy Integral Bound [59]). Let $\{X_\theta, \theta \in \mathbb{T}\}$ be a zero-mean sub-Gaussian process with respect to the induced pseudometric ρ_X from Definition 7.3. Then, for any $\delta \in [0, D]$, we have:*

$$\mathbb{E}\left[\sup_{\theta,\tilde{\theta}\in\mathbb{T}} (X_\theta - X_{\tilde{\theta}})\right] \le 2\mathbb{E}\left[\sup_{\substack{\gamma,\gamma'\in\mathbb{T} \\ \rho_X(\gamma,\gamma')\le\delta}} (X_\gamma - X_{\gamma'})\right] + 32\mathcal{J}(\delta/4; D), \tag{17}$$

$D = \sup_{\theta,\widetilde{\theta}\in\mathbb{T}} \rho_X(\theta,\widetilde{\theta})$; $\mathcal{J}(a;b) = \int_a^b \sqrt{lnC_X(u;\mathbb{T})}du$.

Remark 7.5. Given a fixed $\theta_0 \in \mathbb{T}$, since $\mathbb{E}\left[X_{\theta_0}\right] := \mathbb{E}\left[\langle\theta_0,\sigma\rangle\right] = 0$, we have:

$$\mathbb{E}\left[\sup_{\theta\in\mathbb{T}} X_\theta\right] = \mathbb{E}\left[\sup_{\theta\in\mathbb{T}}(X_\theta - X_{\theta_0})\right] \le \mathbb{E}\left[\sup_{\theta,\widetilde{\theta}\in\mathbb{T}}(X_\theta - X_{\widetilde{\theta}})\right]. \tag{18}$$

Combining (17) with (18), we have:

$$\mathbb{E}\left[\sup_{\theta\in\mathbb{T}} X_\theta\right] \le 2\mathbb{E}\left[\sup_{\substack{\gamma,\gamma'\in\mathbb{T}\\ \rho_X(\gamma,\gamma')\le\delta}}(X_\gamma - X_{\gamma'})\right] + 32\mathcal{J}(\delta/4;D), \tag{19}$$

which can be used to draw the upper bound of Rademacher complexity $\widehat{\mathcal{R}}_S(\widetilde{L}_H)$.

7.1. Linear hypothesis class

This subsection provides a characterization of $\widehat{\mathcal{R}}_S(\widetilde{L}_H)$ for ℓ_p-norm bounded linear function classes and perturbations measured in terms of the ℓ_r-norm.

Theorem 7.6 *($\widehat{\mathcal{R}}_S(\widetilde{L}_H)$ under ℓ_r-norm attack for linear models). We denote the bounded linear function class as $H = \{x \to \langle w,x\rangle \mid \|w\|_p \le U\}$ and $1/p + 1/p^* = 1$. We consider a ℓ_q-norm regression loss, $\ell(y_1,y_2) = |y_1 - y_2|^q$. Accordingly, we have:*

$$\widehat{\mathcal{R}}_S(\widetilde{L}_H) \le 2L_q\widehat{\mathcal{R}}_S(H) + \frac{L_q U\varepsilon}{\sqrt{m}} max\left(d^{1-\frac{1}{r}-\frac{1}{p}}, 1\right), \tag{20}$$

where

$$L_q = q(\sup_x \sup_{\|a-x\|_r\le\varepsilon} |h(a) - h'(x)|)^{q-1}, \tag{21}$$

and

$$\widehat{\mathcal{R}}_S(H) \le \begin{cases} \frac{U}{m}\sqrt{2log(2d)}\|X^T\|_{2,\infty} & if\ p = 1 \\ \frac{\sqrt{2}U}{m}\left[\frac{\Gamma(\frac{p^*+1}{2})}{\sqrt{\pi}}\right]^{\frac{1}{p^*}}\|X^T\|_{2,p^*} & if\ 1 < p \le 2 \\ \frac{U}{m}\|X^T\|_{2,p^*} & if\ p \ge 2 \end{cases}. \tag{22}$$

The proof can be found in Appendix B.9.

Remark 7.7. Theorem 7.6 states that $\widehat{\mathcal{R}}_S(\widetilde{L}_H)$ can be bounded by two components: the Rademacher complexity of the standard linear hypotheses and an additional dimension-dependent term. Combining Proposition 4.3 with (20) and (22), we have:

$$disc_{adv}(P,\widehat{P}) = \mathcal{O}\left(\frac{\mathcal{M}\sqrt{log\frac{1}{\alpha}} + L_q U\varepsilon max(d^{1-\frac{1}{r}-\frac{1}{p}}, 1)}{\sqrt{m}}\right). \tag{23}$$

From (23), we can observe that when m approaches infinity, $disc_{adv}(P,\widehat{P})$ tends to zero.

7.2. Multi-layer neural networks

This subsection analyzes $\widehat{\mathcal{R}}_S(\widetilde{L}_H)$ for multi-layer neural networks, which is far more complicated than linear hypotheses. A significant obstacle is the neural network's non-linearity which prevents us from using Hölder's inequality to remove the *sup* in the formula. Our solution is to calculate the covering number of $\widetilde{L}_H$ and then use Dudley's Entropy Integral Bound to bound $\widehat{\mathcal{R}}_S(\widetilde{L}_H)$.

Formally, we consider z-layer fully connected neural networks:

$$H = \{x \to W_z g(...g(W_1 x)), \|W_l\|_F \le U_l, l = 1,...,z\}, \tag{24}$$

where $g(\cdot)$ is an element-wise L_g-Lipschitz function with $g(0) = 0$, $W_l \in \mathbb{R}^{h_l\times h_{l-1}}$ for $l = 1,...,z$, and $h_z = 1$, $h_0 = d$. We denote the Frobenius norm of matrices by $\|\cdot\|_F$.

Theorem 7.8. *Assume the loss function is ℓ_q-norm regression loss and is L_q-Lipschitz, where L_q is defined by (21). Given the hypothesis class H defined in equation (24), we have:*

$$\widehat{\mathcal{R}}_S(\widetilde{L}_H) \leq \frac{256}{\sqrt{m}} L_q L_g^{z-1} \sqrt{2z \sum_{l=1}^{z} h_l h_{l-1} \prod_{l=1}^{z} U_l} \left[max \left\{ 1, d^{\frac{1}{2}-\frac{1}{r}} \right\} (\|X\|_{r,\infty} + \varepsilon) + \|X\|_{2,\infty} \right]. \tag{25}$$

The proof can be found in Appendix B.10.

Remark 7.9. In the work of [3], the authors also provide the bounds for adversarial Rademacher complexity. However, their hypothesis class only involves one hypothesis, while our hypothesis class $\widetilde{L}_H$ involves a combination of two hypotheses. In addition, they only focus on one-layer neural networks, while we investigate **multi-layer neural networks**, which are much more complicated to analyze.

8. Conclusion

In this work, we study the generalization performance of ARUDA via Rademacher complexity. Moreover, we provide concrete bounds of the Rademacher complexity for linear models and multi-layer neural networks. Motivated by our theory, we propose an effective adversarially robust representation learning algorithm for ARUDA. Experimental results verify our theory and the robustness improvement of our proposed algorithm.

CRediT authorship contribution statement

Lianghe Shi: Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation. **Weiwei Liu:** Writing – review & editing, Supervision, Resources, Project administration, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Lianghe Shi reports a relationship with Pennsylvania State University that includes: non-financial support. Lianghe Shi reports a relationship with The University of Texas at Austin that includes: non-financial support. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is supported by the Key R&D Program of Hubei Province under Grant 2024BAB038, the National Key R&D Program of China under Grant 2023YFC3604702, and the Fundamental Research Funds for the Central Universities under Grant 2042025kf0045.

Appendix A. Additional definitions

Definition A.1 *(Covering Number [59]).* Given a metric space $(\mathbb{T}, \rho)$ where $\mathbb{T}$ is a non-empty set and $\rho(\cdot,\cdot)$ is a (pseudo)-metric. A δ-cover of $\mathbb{T}$ with respect to ρ is a set $\{\theta^1, ..., \theta^C\} \subset \mathbb{T}$ such that for each $\theta \in \mathbb{T}$, there exists some $i \in \{1, ..., C\}$ such that $\rho(\theta, \theta^i) \leq \delta$. The δ-covering number $\mathcal{C}(\delta; \mathbb{T}, \rho)$ is the cardinality of the smallest δ-cover. We use $C_X(\delta; \mathbb{T})$ as shorthand to denote $\mathcal{C}(\delta; \mathbb{T}, \rho_X)$.

Definition A.2 *(Packing Number [59]).* Given a metric space $(\mathbb{T}, \rho)$. A δ-packing of $\mathbb{T}$ with respect to ρ is a set $\{\theta^1, ..., \theta^{\mathcal{P}}\} \subset \mathbb{T}$ such that $\rho(\theta^i, \theta^j) > \delta$ for all distinct $i, j \in \{1, ..., \mathcal{P}\}$. The δ-packing number $\mathcal{P}(\delta; \mathbb{T}, \rho)$ is the cardinality of the largest δ-packing.

Definition A.3 *(Sub-Gaussian Process [59]).* A collection of zero-mean random variables $\{X_\theta, \theta \in \mathbb{T}\}$ is a sub-Gaussian process with respect to a metric ρ_X on $\mathbb{T}$ if:

$$\mathbb{E}[e^{\nu(X_\theta - X_{\tilde{\theta}})}] \leq e^{\frac{\nu^2 \rho_X^2(\theta, \tilde{\theta})}{2}}, \ \forall \theta, \tilde{\theta} \in \mathbb{T}, \nu \in \mathbb{R}.$$

Remark A.4. It can be easily proven that the Rademacher process satisfies the condition in Definition 7.3 with respect to the ℓ_2-norm [59].

Appendix B. Proofs

B.1. Proof of Example 4.1

Proof of Example 4.1. For $h(x) = sgn(x + \frac{1}{2})$, we can obtain that:

$$\epsilon_P^{adv}(h, f_P) = \Phi(-\frac{1}{2} + \epsilon + \mu) - \Phi(b_P + \mu),$$
$$\epsilon_Q^{adv}(h, f_Q) = \Phi(-\frac{1}{2} + \epsilon - \mu) - \Phi(b_Q - \mu),$$

where $\Phi(x)$ is the cumulative distribution function (CDF) of the Gaussian distribution $\mathcal{N}(0, \sigma^2)$. If we set $b_P = b_Q = -\infty$, then

$$\epsilon_P^{adv}(h, f_P) = \Phi(-\frac{1}{2} + \epsilon + \mu),$$
$$\epsilon_Q^{adv}(h, f_Q) = \Phi(-\frac{1}{2} + \epsilon - \mu).$$

The divergence $d_{H\triangle H}$ can be calculated as:

$$d_{H\triangle H} = \sup_{b', b'' \in Z} \{(\Phi(b' + \frac{1}{2} + \mu) - \Phi(b'' + \frac{1}{2} + \mu)) - (\Phi(b' + \frac{1}{2} - \mu) - \Phi(b'' + \frac{1}{2} - \mu))\}.$$

We can easily derive that the optimal value is achieved at $b' = -1, b'' = -\infty$. Then we have:

$$d_{H\triangle H} = \Phi(-\frac{1}{2} + \mu) - \Phi(-\frac{1}{2} - \mu).$$

In addition, we can obtain that:

$$\min_{h^* \in H} \{\epsilon_P^{adv}(h^*) + \epsilon_Q^{adv}(h^*)\} = 0,$$

where the optimal value is achieved at $b_{h^*} = -\infty$.

By combining these terms and setting $\epsilon \leq \frac{1}{2}$, we can obtain the conclusion:

$$\epsilon_Q^{adv}(h) > \epsilon_P^{adv}(h) + d_{H\triangle H}(P, Q) + \min_{h^* \in H} \{\epsilon_P^{adv}(h^*) + \epsilon_Q^{adv}(h^*)\}. \quad \square$$

B.2. Proposition Appendix B.1

Based on the definition of AD, we can directly derive that AD upper bounds the difference of adversarial error between two domains.

Proposition B.1. *For any $h, h' \in H$, where H is a set of functions, we have:*

$$\left|\epsilon_P^{adv}(h, h') - \epsilon_Q^{adv}(h, h')\right| \leq disc_{adv}(P, Q). \tag{B.1}$$

Proof of Proposition Appendix B.1. Using the definition of adversarial error and AD, we have:

$$\begin{aligned} \forall h, h' \in H, \quad & \left|\epsilon_P^{adv}(h, h') - \epsilon_Q^{adv}(h, h')\right| \\ & \leq \sup_{h, h' \in H} \left|\epsilon_P^{adv}(h, h') - \epsilon_Q^{adv}(h, h')\right| \\ & = disc_{adv}(P, Q). \quad \square \end{aligned}$$

B.3. Proposition Appendix B.2

We claim that AD is a pseudometric, in that it satisfies all properties of a metric, except that there may exist pairs $P \neq Q$ satisfying $disc_{adv}(P, Q) = 0$.

Proposition B.2. *Adversarial discrepancy (AD) defined in Definition 4.2 is a pseudometric. For any distributions P, Q and T over $\mathcal{X}$, we have:*

$$disc_{adv}(P, T) \leq disc_{adv}(P, Q) + disc_{adv}(Q, T).$$

Proof of Proposition Appendix B.2. AD is clearly symmetric and we only verify the triangle inequality here. For any distribution P, Q and T over $\mathcal{X}$:

$$
\begin{aligned}
& disc_{adv}(P,Q) + disc_{adv}(Q,T) \\
&= \sup_{h,h' \in H} \left\{ \left| \epsilon_P^{adv}(h,h') - \epsilon_Q^{adv}(h,h') \right| \right\} + \sup_{h,h' \in H} \left\{ \left| \epsilon_Q^{adv}(h,h') - \epsilon_T^{adv}(h,h') \right| \right\} \\
&\geq \sup_{h,h' \in H} \left\{ \left| \epsilon_P^{adv}(h,h') - \epsilon_Q^{adv}(h,h') \right| + \left| \epsilon_Q^{adv}(h,h') - \epsilon_T^{adv}(h,h') \right| \right\} \\
&\geq \sup_{h,h' \in H} \left| \epsilon_P^{adv}(h,h') - \epsilon_T^{adv}(h,h') \right| \\
&= disc_{adv}(P,T) \quad \square
\end{aligned}
\tag{B.2}
$$

B.4. Proof of Proposition 4.3

AD offers an important advantage in that it can be estimated from finite samples. The following theorem shows that for a bounded loss function ℓ, AD between a distribution and its empirical distribution can be bounded in terms of the empirical Rademacher complexity.

Proposition 4.3. Let ℓ be a loss function bounded by $\mathcal{M} > 0$. Let P be a distribution over $\mathcal{X}$, while $\widehat{P}$ is the corresponding empirical distribution for samples $S = (x_1, ..., x_m)$. Let $\widetilde{L}_H = \{x \to sup_{a \in N(x)} \ell(h(a), h'(x)) | h, h' \in H\}$ be a class of functions. Then, for any $\alpha > 0$, with probability at least $1 - \alpha$ over samples S of size m drawn according to P, we have:

$$
disc_{adv}(P,\widehat{P}) \leq \widehat{\mathcal{R}}_S(\widetilde{L}_H) + 3\mathcal{M}\sqrt{\frac{\log\frac{2}{\alpha}}{2m}}. \tag{B.3}
$$

Proof of Proposition 4.3. We first scale the loss ℓ to [0,1] by dividing $\mathcal{M}$, and get the new class $\frac{\widetilde{L}_H}{\mathcal{M}}$. Then the class meets the conditions of Lemma 3.2. Applying Lemma 3.2 in $\frac{\widetilde{L}_H}{\mathcal{M}}$, for any $\alpha > 0$, with probability at least $1 - \alpha$, the following inequality holds for all $h, h' \in H$:

$$
\mathop{\mathbb{E}}_{x \sim P}\left[\sup_{a \in N(x)} \frac{\ell(h(a), f(x))}{\mathcal{M}}\right] \leq \mathop{\mathbb{E}}_{x \sim \widehat{P}}\left[\sup_{a \in N(x)} \frac{\ell(h(a), f(x))}{\mathcal{M}}\right] + \widehat{\mathcal{R}}_S(\frac{\widetilde{L}_H}{\mathcal{M}}) + 3\sqrt{\frac{\log\frac{2}{\alpha}}{2m}}.
$$

By the definition of $\epsilon_P^{adv}(h,h')$,

$$
\frac{\epsilon_P^{adv}(h,h')}{\mathcal{M}} \leq \frac{\epsilon_{\widehat{P}}^{adv}(h,h')}{\mathcal{M}} + \widehat{\mathcal{R}}_S(\frac{\widetilde{L}_H}{\mathcal{M}}) + 3\sqrt{\frac{\log\frac{2}{\alpha}}{2m}}.
$$

Multiply $\mathcal{M}$ at both sides of this inequality,

$$
\epsilon_P^{adv}(h,h') \leq \epsilon_{\widehat{P}}^{adv}(h,h') + \widehat{\mathcal{R}}_S(\widetilde{L}_H) + 3\mathcal{M}\sqrt{\frac{\log\frac{2}{\alpha}}{2m}}.
$$

Finally, by Proposition Appendix B.1,

$$
disc_{adv}(P,\widehat{P}) \leq \widehat{\mathcal{R}}_S(\widetilde{L}_H) + 3\mathcal{M}\sqrt{\frac{\log\frac{2}{\alpha}}{2m}}. \quad \square
$$

B.5. Proof of Proposition 4.4

Proposition 4.4. (Discrepancy Estimation Bound). Let ℓ be a loss function bounded by $\mathcal{M} > 0$. $P, \widehat{P}, Q, \widehat{Q}$ are defined in the same way as in Proposition 4.3. Then, for any $\alpha > 0$, with probability at least $1 - 2\alpha$ over samples S of size m drawn according to P and samples $\mathcal{T}$ of size n drawn according to Q:

$$
disc_{adv}(P,Q) \leq disc_{adv}(\widehat{P},\widehat{Q}) + \widehat{\mathcal{R}}_S(\widetilde{L}_H) + \widehat{\mathcal{R}}_{\mathcal{T}}(\widetilde{L}_H) + 3\mathcal{M}\left(\sqrt{\frac{\log\frac{2}{\alpha}}{2m}} + \sqrt{\frac{\log\frac{2}{\alpha}}{2n}}\right). \tag{B.4}
$$

Proof of Proposition 4.4. By the Proposition Appendix B.2 we can write:

$$
disc_{adv}(P,Q) \leq disc_{adv}(P,\widehat{P}) + disc_{adv}(\widehat{P},\widehat{Q}) + disc_{adv}(\widehat{Q},Q).
$$

The results then follow by the application of Proposition 4.3 to $disc_{adv}(P,\widehat{P})$ and $disc_{adv}(Q,\widehat{Q})$ and union bound. $\square$

B.6. Proof of Theorem 4.5

Theorem 4.5. (Generalization Error Bound for ARUDA) Let ℓ be a loss function that is symmetric and obeys the triangle inequality. Then, for any hypothesis $h \in H$, the following inequality holds:

$$\epsilon_Q^{adv}(h, f_Q) \leq \epsilon_P^{adv}(h, f_P) + \lambda + disc_{adv}(P, Q), \tag{B.5}$$

where

$$\lambda = \epsilon_P^{adv}(h^*, f_P) + \epsilon_Q^{adv}(h^*, f_Q), \tag{B.6}$$

and

$$h^* = \underset{h \in H}{argmin}\ \epsilon_P^{adv}(h, f_P) + \epsilon_Q^{adv}(h, f_Q). \tag{B.7}$$

Before we provide the proof of Theorem 4.5, we introduce the following lemma to claim that the adversarial error ϵ_D^{adv} verifies the triangle inequality.

Lemma B.3. *Assume the loss function l is symmetric and obeys the triangle inequality. Then, for any distribution D and any functions f_1, f_2, f_3, we have:*

$$\begin{aligned} \epsilon_D^{adv}(f_1, f_3) &\leq \epsilon_D^{adv}(f_1, f_2) + \epsilon_D^{adv}(f_2, f_3) \\ \epsilon_D^{adv}(f_1, f_2) &\leq \epsilon_D^{adv}(f_1, f_3) + \epsilon_D^{adv}(f_2, f_3) \end{aligned} \tag{B.8}$$

Remark B.4. Note that the adversarial error is not symmetric.

Proof of Lemma Appendix B.3. For any x, we have:

$$\begin{aligned} &\sup_{a \in N(x)} \ell(f_1(a), f_3(x)) \\ &= \ell(f_1(a^*), f_3(x)) \quad (a^* = \underset{a \in N(x)}{argmax}\ \ell(f_1(a), f_3(x))) \\ &\overset{(i)}{\leq} \ell(f_1(a^*), f_2(x)) + \ell(f_2(x), f_3(x)) \\ &\leq \sup_{a \in N(x)} \ell(f_1(a), f_2(x)) + \sup_{a \in N(x)} \ell(f_2(a), f_3(x)), \end{aligned}$$

where (i) follows from the application of the triangle inequality of ℓ.

Then, by taking expectation in both sides we yields the first inequality in (B.8). The second inequality can be derived in a similar way. □

Now we prove the Theorem 4.5.

Proof of Theorem 4.5.

$$\begin{aligned} &\epsilon_Q^{adv}(h, f_Q) \\ &\overset{(i)}{\leq} \epsilon_Q^{adv}(h, h^*) + \epsilon_Q^{adv}(h^*, f_Q) \\ &\leq \epsilon_Q^{adv}(h^*, f_Q) + \epsilon_P^{adv}(h, h^*) + |\epsilon_Q^{adv}(h, h^*) - \epsilon_P^{adv}(h, h^*)| \\ &\overset{(ii)}{\leq} \epsilon_Q^{adv}(h^*, f_Q) + \epsilon_P^{adv}(h, h^*) + disc_{adv}(P, Q) \\ &\overset{(iii)}{\leq} \epsilon_Q^{adv}(h^*, f_Q) + \epsilon_P^{adv}(h, f_P) + \epsilon_P^{adv}(h^*, f_P) + disc_{adv}(P, Q) \\ &\overset{(iv)}{\leq} \epsilon_P^{adv}(h, f_P) + \lambda + disc_{adv}(P, Q), \end{aligned}$$

where (i) and (iii) are the applications of (B.8); (ii) is from the Proposition Appendix B.1; (iv) comes from (9). □

B.7. Proof of Corollary 4.7

Corollary 4.7. Let ℓ be a loss function that is symmetric and obeys the triangle inequality. Then, for any hypothesis $h \in H$ and $\alpha > 0$, with probability at least $1 - 3\alpha$ over samples S of size m drawn according to P and samples $\mathcal{T}$ of size n drawn according to Q, the following inequality holds:

$$\epsilon_Q^{adv}(h, f_Q) \le \hat{\epsilon}_P^{adv}(h, f_P) + \lambda + disc_{adv}(\hat{P}, \hat{Q}) + 2\hat{\mathcal{R}}_S(\widetilde{L}_H) + \hat{\mathcal{R}}_T(\widetilde{L}_H) + 3\mathcal{M}\left(2\sqrt{\frac{\log\frac{2}{\alpha}}{2m}} + \sqrt{\frac{\log\frac{2}{\alpha}}{2n}}\right). \tag{B.9}$$

Proof of Corollary 4.7. By Theorem 4.5, for any hypothesis $h \in H$, we have:

$$\epsilon_Q^{adv}(h, f_Q) \le \epsilon_P^{adv}(h, f_P) + \lambda + disc_{adv}(P, Q). \tag{B.10}$$

We first bound the term $\epsilon_P^{adv}(h, f_P)$. By Proposition Appendix B.1, we have:

$$\epsilon_P^{adv}(h, f_P) \le \hat{\epsilon}_P^{adv}(h, f_P) + disc_{adv}(P, \hat{P}). \tag{B.11}$$

Then, we bound the terms $disc_{adv}(P, \hat{P})$ and $disc_{adv}(P, Q)$ by Proposition 4.3 and Corollary 4.4 respectively. For any $\alpha > 0$, with probability at least $1 - \alpha$, the following inequality holds:

$$disc_{adv}(P, \hat{P}) \le \hat{\mathcal{R}}_S(\widetilde{L}_H) + 3\mathcal{M}\sqrt{\frac{\log\frac{2}{\alpha}}{2m}}. \tag{B.12}$$

For any $\alpha > 0$, with probability at least $1 - 2\alpha$, the following inequality holds:

$$disc_{adv}(P, Q) \le disc_{adv}(\hat{P}, \hat{Q}) + \hat{\mathcal{R}}_S(\widetilde{L}_H) + \hat{\mathcal{R}}_T(\widetilde{L}_H) + 3\mathcal{M}\left(\sqrt{\frac{\log\frac{2}{\alpha}}{2m}} + \sqrt{\frac{\log\frac{2}{\alpha}}{2n}}\right). \tag{B.13}$$

Combining (B.10), (B.11), (B.12), (B.13) and union bound, then, for any $\alpha > 0$, with probability at least $1 - 3\alpha$, the following inequality holds:

$$\epsilon_Q^{adv}(h, f_Q) \le \hat{\epsilon}_P^{adv}(h, f_P) + \lambda + disc_{adv}(\hat{P}, \hat{Q}) + 2\hat{\mathcal{R}}_S(\widetilde{L}_H) + \hat{\mathcal{R}}_T(\widetilde{L}_H) + 3\mathcal{M}(2\sqrt{\frac{\log\frac{2}{\alpha}}{2m}} + \sqrt{\frac{\log\frac{2}{\alpha}}{2n}}). \quad \square$$

B.8. Proof of Lemma 7.2

Lemma 7.2. Let $\mathbb{B}_d^p(\tau)$ be the ℓ_p-norm ball in $\mathbb{R}^d$ with radius τ. The δ-covering number of $\mathbb{B}_d^p(\tau)$ with respect to the ℓ_p-norm obeys the following bound:

$$C(\delta; \mathbb{B}_d^p(\tau), \|\cdot\|_p) \le \left(1 + \frac{2\tau}{\delta}\right)^d.$$

Proof of Lemma 7.2. Let $\{\theta^1, ..., \theta^{\mathcal{P}}\}$ be a maximal δ-packing of $\mathbb{B}_d^p(\tau)$ in the ℓ_p-norm. By maximality, this set must also be a δ-covering of $\mathbb{B}$ under the ℓ_p-norm. By the definition of δ-packing (see Appendix A), the balls $\{\theta^j + \frac{\delta}{2}\mathbb{B}_d^p(1),\ j = 1, ..., \mathcal{P}\}$ are all disjoint and contained within $\mathbb{B}_d^p(\tau) + \frac{\delta}{2}\mathbb{B}_d^p(1)$. Taking volumes, we conclude that $\sum_{j=1}^{\mathcal{P}} vol(\theta^j + \frac{\delta}{2}\mathbb{B}_d^p(1)) \le vol(\mathbb{B}_d^p(\tau) + \frac{\delta}{2}\mathbb{B}_d^p(1))$, and hence

$$\mathcal{P}\ vol(\frac{\delta}{2}\mathbb{B}_d^p(1)) \le vol((\tau + \frac{\delta}{2})\mathbb{B}_d^p(1)).$$

Finally, we have $vol(k\ \mathbb{B}_d^p(1)) = k^d\ vol(\mathbb{B}_d^p(1))$, then

$$\mathcal{P} \le \left(1 + \frac{2\tau}{\delta}\right)^d.$$

The conclusion follows since $C(\delta; \mathbb{B}_d^p(\tau), \|\cdot\|_p) \le \mathcal{P}$. $\square$

B.9. Proof of Theorem 7.6

Theorem 7.6 ($\hat{\mathcal{R}}_S(\widetilde{L}_H)$ *under* ℓ_r*-norm attack for linear models).* Let $H = \{x \to \langle w, x\rangle | \ \|w\|_p \le U\}$ be the bounded linear function class and $1/p + 1/p^* = 1$. We consider a ℓ_q-norm regression loss, $\ell(y_1, y_2) = |y_1 - y_2|^q$. Accordingly, we have:

$$\hat{\mathcal{R}}_S(\widetilde{L}_H) \le 2L_q\hat{\mathcal{R}}_S(H) + \frac{L_q U\varepsilon}{\sqrt{m}} max\left(d^{1-\frac{1}{r}-\frac{1}{p}}, 1\right), \tag{B.14}$$

where

$$L_q = q(\sup_x \sup_{\|a-x\|_r \le \varepsilon} |h(a) - h'(x)|)^{q-1} \tag{B.15}$$

and

$$\widehat{\mathcal{R}}_S(H) \leq \begin{cases} \frac{U}{m}\sqrt{2log(2d)}\|X^T\|_{2,\infty} & if\ p=1 \\ \frac{\sqrt{2}U}{m}\left[\frac{\Gamma(\frac{p^*+1}{2})}{\sqrt{\pi}}\right]^{\frac{1}{p^*}}\|X^T\|_{2,p^*} & if\ 1<p\leq 2 \\ \frac{U}{m}\|X^T\|_{2,p^*} & if\ p\geq 2 \end{cases}. \tag{B.16}$$

Before proceeding with the proof of Theorem 7.6, we introduce some useful lemmas.

Lemma B.5 *([3]). Let $p, r > 1$ and let $\frac{1}{r}+\frac{1}{r^*}=1$. Then it holds that:*

$$\mathop{\mathbb{E}}_{\sigma}\left[\sup_{\|w\|_p\leq U}\frac{2}{m}\sum_{i=1}^{m}\sigma_i\epsilon\|w\|_{r^*}\right] \geq \frac{U\epsilon\ max\left(d^{1-\frac{1}{r}-\frac{1}{p}},1\right)}{\sqrt{2m}} \tag{B.17}$$

and

$$\mathop{\mathbb{E}}_{\sigma}\left[\sup_{\|w\|_p\leq U}\frac{2}{m}\sum_{i=1}^{m}\sigma_i\epsilon\|w\|_{r^*}\right] \leq \frac{U\epsilon\ max\left(d^{1-\frac{1}{r}-\frac{1}{p}},1\right)}{\sqrt{m}}. \tag{B.18}$$

Lemma B.6 *([3]). Let $H=\{x\to\langle w,x\rangle|\ \|w\|_p\leq U\}$ be the bounded linear function class. Let $\frac{1}{p}+\frac{1}{p^*}=1$. Then, given samples $S=\{x_1,...,x_m\}$, we have:*

$$\widehat{\mathcal{R}}_S(H) \leq \begin{cases} \frac{U}{m}\sqrt{2log(2d)}\|X^T\|_{2,\infty} & if\ p=1 \\ \frac{\sqrt{2}U}{m}\left[\frac{\Gamma(\frac{p^*+1}{2})}{\sqrt{\pi}}\right]^{\frac{1}{p^*}}\|X^T\|_{2,p^*} & if\ 1<p\leq 2 \\ \frac{U}{m}\|X^T\|_{2,p^*} & if\ p\geq 2 \end{cases}. \tag{B.19}$$

Proof of Theorem 7.6. The proof mainly contains two parts. Firstly, by the linearity of hypothesis we remove the *sup* in $\widetilde{L}_H$. We convert the adversarial Rademacher Complexity $\widehat{\mathcal{R}}_S(\widetilde{L}_H)$ to the non-adversarial Rademacher Complexity $\widehat{\mathcal{R}}_S(H)$. Then we can use lemma Appendix B.6 to bound it.

By the definition of Rademacher Complexity, we have:

$$\begin{aligned}\widehat{\mathcal{R}}_S(\widetilde{L}_H) &= \mathop{\mathbb{E}}_{\sigma}\left[\sup_{h,h'\in H}\frac{2}{m}\sum_{i=1}^{m}\sigma_i\left(\sup_{\|a_i-x_i\|_r\leq\epsilon}\ell\left(h(a_i),h'(x_i)\right)\right)\right] \\ &= \mathop{\mathbb{E}}_{\sigma}\left[\sup_{h,h'\in H}\frac{2}{m}\sum_{i=1}^{m}\sigma_i\left(\sup_{\|a_i-x_i\|_r\leq\epsilon}\left|h(a_i)-h'(x_i)\right|^q\right)\right] \\ &\overset{(i)}{=} \mathop{\mathbb{E}}_{\sigma}\left[\sup_{h,h'\in H}\frac{2}{m}\sum_{i=1}^{m}\sigma_i\left(\sup_{\|a_i-x_i\|_r\leq\epsilon}\left|h(a_i)-h'(x_i)\right|\right)^q\right],\end{aligned} \tag{B.20}$$

where (i) is due to the monotonicity of the function $f(x)=x^q$ over $x>0$.

Since the input x is bounded, we assume the function $f(x)=\left(\sup_{\|a-x\|_r\leq\epsilon}\left|h(a)-h'(x)\right|\right)^q$ is L^q Lipschitz(L^q is defined in Theorem 7.6). Thus, by Talagrand's contraction Lemma [29], $\widehat{\mathcal{R}}_S(\widetilde{L}_H)$ is bounded by $L_q\widehat{\mathcal{R}}_S(\widetilde{L}_{H'})$, where

$$H'=\left\{x\to\sup_{\|a-x\|_r\leq\epsilon}\left|h(a)-h'(x)\right|\right\}.$$

Then, we have:

$$\begin{aligned}\widehat{\mathcal{R}}_S(\widetilde{L}_H) &\le L_q \mathop{\mathbb{E}}_{\sigma}\left[\sup_{h,h'\in H}\frac{2}{m}\sum_{i=1}^{m}\sigma_i \sup_{\|a_i-x_i\|_r\le\varepsilon}\left|h(a_i)-h'(x_i)\right|\right]\\ &= L_q \mathop{\mathbb{E}}_{\sigma}\left[\sup_{\substack{\|w\|_p\le U\\ \|w'\|_p\le U}}\frac{2}{m}\sum_{i=1}^{m}\sigma_i \sup_{\|a_i-x_i\|_r\le\varepsilon}\left|\langle w,a_i\rangle-\langle w',x_i\rangle\right|\right]\\ &= L_q \mathop{\mathbb{E}}_{\sigma}\left[\sup_{\substack{\|w\|_p\le U\\ \|w'\|_p\le U}}\frac{2}{m}\sum_{i=1}^{m}\sigma_i \sup_{\|\triangle_i\|_r\le\varepsilon}\left|\langle w,x_i\rangle-\langle w',x_i\rangle+\langle w,\triangle_i\rangle\right|\right].\end{aligned} \tag{B.21}$$

To further simplify this formula, we calculate the exact solution of the inner supremum over Δ_i, i.e.,

$$\sup_{\|\Delta_i\|_r\le\varepsilon}|\langle w,x_i\rangle-\langle w',x_i\rangle+\langle w,\Delta_i\rangle|.$$

Using Hölder's inequality, we obtain that:

$$-\varepsilon\|w\|_{r^*}\le\langle w,\Delta_i\rangle\le\varepsilon\|w\|_{r^*},$$

where $\frac{1}{r}+\frac{1}{r^*}=1$, since $\|\Delta_i\|_r\le\varepsilon$. Hence, when $\langle w,x_i\rangle-\langle w',x_i\rangle\le 0$, we can find an appropriate Δ_i such that $\langle w,\Delta_i\rangle=-\varepsilon\|w\|_{r^*}$, then $|\langle w,x_i\rangle-\langle w',x_i\rangle+\langle w,\Delta_i\rangle|$ reaches the supremum, and

$$\sup_{\|\Delta_i\|_r\le\varepsilon}|\langle w,x_i\rangle-\langle w',x_i\rangle+\langle w,\Delta_i\rangle|=\left|\langle w-w',x_i\rangle\right|+\varepsilon\|w\|_{r^*}.$$

Similarly, when $\langle w,x_i\rangle-\langle w',x_i\rangle\ge 0$, we can find an appropriate Δ_i such that $\langle w,\Delta_i\rangle=\varepsilon\|w\|_{r^*}$, then $|\langle w,x_i\rangle-\langle w',x_i\rangle+\langle w,\Delta_i\rangle|$ reaches the supremum, and

$$\sup_{\|\Delta_i\|_r\le\varepsilon}|\langle w,x_i\rangle-\langle w',x_i\rangle+\langle w,\Delta_i\rangle|=\left|\langle w-w',x_i\rangle\right|+\varepsilon\|w\|_{r^*}.$$

Then, we obtain that:

$$\begin{aligned}\widehat{\mathcal{R}}_S(\widetilde{L}_H) &\le L_q \mathop{\mathbb{E}}_{\sigma}\left[\sup_{\substack{\|w\|_p\le U\\ \|w'\|_p\le U}}\frac{2}{m}\sum_{i=1}^{m}\sigma_i \sup_{\|\Delta_i\|_r\le\varepsilon}|\langle w,x_i\rangle-\langle w',x_i\rangle+\langle w,\Delta_i\rangle|\right]\\ &= L_q \mathop{\mathbb{E}}_{\sigma}\left[\sup_{\substack{\|w\|_p\le U\\ \|w'\|_p\le U}}\frac{2}{m}\sum_{i=1}^{m}\sigma_i(\left|\langle w-w',x_i\rangle\right|+\varepsilon\|w\|_{r^*})\right].\end{aligned} \tag{B.22}$$

By the subadditivity of *sup*, we then have:

$$\begin{aligned}\widehat{\mathcal{R}}_S(\widetilde{L}_H) &\le L_q \mathop{\mathbb{E}}_{\sigma}\left[\sup_{\substack{\|w\|_p\le U\\ \|w'\|_p\le U}}\frac{2}{m}\sum_{i=1}^{m}\sigma_i\left|\langle w-w',x_i\rangle\right|+\sup_{\|w\|_p\le U}\frac{2}{m}\sum_{i=1}^{m}\sigma_i\varepsilon\|w\|_{r^*}\right]\\ &\le L_q \mathop{\mathbb{E}}_{\sigma}\left[\sup_{\substack{\|w\|_p\le U\\ \|w'\|_p\le U}}\frac{2}{m}\sum_{i=1}^{m}\sigma_i\left|\langle w-w',x_i\rangle\right|\right]+L_q \mathop{\mathbb{E}}_{\sigma}\left[\sup_{\|w\|_p\le U}\frac{2}{m}\sum_{i=1}^{m}\sigma_i\varepsilon\|w\|_{r^*}\right].\end{aligned} \tag{B.23}$$

For the first part in (B.23), we reuse Talagrand's contraction Lemma by assuming $f(x)=|x|$ is 1-Lipschitz. For the second component, we use lemma Appendix B.5 to directly bound it. Then we have:

$$\begin{aligned}\widehat{\mathcal{R}}_S(\widetilde{L}_H) &\le L_q \mathop{\mathbb{E}}_{\sigma}\left[\sup_{\substack{\|w\|_p\le U\\ \|w'\|_p\le U}} \frac{2}{m}\sum_{i=1}^{m}\sigma_i\langle w-w', x_i\rangle\right] + \frac{L_q U\varepsilon}{\sqrt{m}} max\left(d^{1-\frac{1}{r}-\frac{1}{p}}, 1\right)\\ &= L_q \mathop{\mathbb{E}}_{\sigma}\left[\sup_{\|w\|_p\le U} \frac{2}{m}\sum_{i=1}^{m}\sigma_i\langle w, x_i\rangle\right] + L_q \mathop{\mathbb{E}}_{\sigma}\left[\sup_{\|w'\|_p\le U} \frac{2}{m}\sum_{i=1}^{m}\sigma_i\langle w', x_i\rangle\right] + \frac{L_q U\varepsilon}{\sqrt{m}} max\left(d^{1-\frac{1}{r}-\frac{1}{p}}, 1\right)\\ &\le 2L_q \mathop{\mathbb{E}}_{\sigma}\left[\sup_{\|w\|_p\le U} \frac{2}{m}\sum_{i=1}^{m}\sigma_i\langle w, x_i\rangle\right] + \frac{L_q U\varepsilon}{\sqrt{m}} max\left(d^{1-\frac{1}{r}-\frac{1}{p}}, 1\right)\\ &\overset{(i)}{=} 2L_q\widehat{\mathcal{R}}_S(H) + \frac{L_q U\varepsilon}{\sqrt{m}} max\left(d^{1-\frac{1}{r}-\frac{1}{p}}, 1\right),\end{aligned} \tag{B.24}$$

where H is the linear hypothesis class we defined in Theorem 7.6. Combining with the Lemma Appendix B.6, we can get:

$$\widehat{\mathcal{R}}_S(\widetilde{L}_H) \le \begin{cases} \frac{U}{m}\sqrt{2log(2d)}\|X^T\|_{2,\infty} + \frac{L_q U\varepsilon}{\sqrt{m}} max\left(d^{1-\frac{1}{r}-\frac{1}{p}}, 1\right) & if\ p=1\\ \frac{\sqrt{2}U}{m}\left[\frac{\Gamma(\frac{p^*+1}{2})}{\sqrt{\pi}}\right]^{\frac{1}{p^*}}\|X^T\|_{2,p^*} + \frac{L_q U\varepsilon}{\sqrt{m}} max\left(d^{1-\frac{1}{r}-\frac{1}{p}}, 1\right) & if\ 1<p\le 2\\ \frac{U}{m}\|X^T\|_{2,p^*} + \frac{L_q U\varepsilon}{\sqrt{m}} max\left(d^{1-\frac{1}{r}-\frac{1}{p}}, 1\right) & if\ p\ge 2\end{cases}. \quad \square \tag{B.25}$$

B.10. Proof of Theorem 7.8

Theorem 7.8. Assume the loss function is ℓ_q-norm regression loss and is L_q-Lipschitz, where L_q is defined by (B.15). Given the hypothesis class H defined in equation (24), we have:

$$\widehat{\mathcal{R}}_S(\widetilde{L}_H) \le \frac{256}{\sqrt{m}} L_q L_g^{z-1}\sqrt{2z\sum_{l=1}^{z} h_l h_{l-1}}\prod_{l=1}^{z} U_l\left[max\left\{1, d^{\frac{1}{2}-\frac{1}{r}}\right\}(\|X\|_{r,\infty}+\varepsilon) + \|X\|_{2,\infty}\right]. \tag{B.26}$$

Before we provide the proof, we first introduce the following lemmas.

Lemma B.7. *If* $x_i^* \in N(x_i) = \left\{x_i' \middle| \|x_i - x_i'\|_r \le \varepsilon\right\}$, *then for* $\frac{1}{q^*}+\frac{1}{q}=1$, *we have:*

$$\|x_i^*\|_{q^*} \le max\left\{1, d^{1-\frac{1}{q}-\frac{1}{r}}\right\}\left(\|X\|_{r,\infty}+\varepsilon\right). \tag{B.27}$$

Proof of Lemma Appendix B.7. We divide the proof into two cases.

1. If $r \ge q^*$, then by Hölder's inequality with $\frac{1}{q^*} = \frac{1}{r}+\frac{1}{s}$, we have:

$$\|x_i^*\|_{q^*} \le sup\ \|\mathbf{1}\|_s \cdot \|x_i^*\|_r = \|\mathbf{1}\|_s \cdot \|x_i^*\|_r = d^{\frac{1}{s}}\|x_i^*\|_r = d^{1-\frac{1}{q}-\frac{1}{r}}\|x_i^*\|_r,$$

   where the equality holds when all the entries are equal.
2. If $r < q^*$, we have:

$$\|x_i^*\|_{q^*} \le \|x_i^*\|_r,$$

   where the equality holds when one of the entries of x_i^* equals to one and the others equal to zero.

Then we have:

$$\begin{aligned}\|x_i^*\|_{q^*} &\le max\left\{1, d^{1-\frac{1}{q}-\frac{1}{r}}\right\}\|x_i^*\|_r\\ &\le max\left\{1, d^{1-\frac{1}{q}-\frac{1}{r}}\right\}\left(\|x_i\|_r + \|x_i - x_i^*\|_r\right)\\ &\le max\left\{1, d^{1-\frac{1}{q}-\frac{1}{r}}\right\}\left(\|X\|_{r,\infty}+\varepsilon\right). \quad \square\end{aligned}$$

Lemma B.8. *Let $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^n$, then we have:*

$$\|A \cdot b\|_2 \le |||A|||_F \cdot \|b\|_2.$$

Proof of Lemma Appendix B.8. Let A_i be the rows of $A, i = 1, 2, \dots, m$, we have:

$$\|A \cdot b\|_2 = \sqrt{\sum_{i=1}^{m} \left(A_i b\right)^2} \overset{(i)}{\le} \sqrt{\sum_{i=1}^{m} \|A_i\|_2^2 \cdot \|b\|_2^2} = \sqrt{\sum_{i=1}^{m} \|A_i\|_2^2} \cdot \sqrt{\|b\|_2^2} = |||A|||_F \cdot \|b\|_2,$$

where (i) is from Hölder's inequality. □

Now, we turn to bound $\widehat{\mathcal{R}}_S(\widetilde{L}_H)$ with $H = \{x \to W_z g(...g(W_1 x)), \|W_l\|_F \le U_l, l = 1, ..., z\}$ and $\ell(y_1, y_2) = |y_1 - y_2|^q$.

Proof of Theorem 7.8. Firstly, we use Talagrand's contraction lemma to remove the power to q. By the definition of Rademacher complexity, we have:

$$\begin{aligned}
\widehat{\mathcal{R}}_S(\widetilde{L}_H) &= \mathop{\mathbb{E}}_{\sigma}\left[\sup_{h,h'\in H} \frac{2}{m}\sum_{i=1}^{m}\sigma_i\left(\sup_{\|a_i - x_i\|_r \le \epsilon} \ell\left(h(a_i), h'(x_i)\right)\right)\right] \\
&= \mathop{\mathbb{E}}_{\sigma}\left[\sup_{h,h'\in H} \frac{2}{m}\sum_{i=1}^{m}\sigma_i\left(\sup_{\|a_i - x_i\|_r \le \epsilon} \left|h(a_i) - h'(x_i)\right|^q\right)\right] \\
&= \mathop{\mathbb{E}}_{\sigma}\left[\sup_{h,h'\in H} \frac{2}{m}\sum_{i=1}^{m}\sigma_i\left(\sup_{\|a_i - x_i\|_r \le \epsilon} \left|h(a_i) - h'(x_i)\right|\right)^q\right] \\
&\overset{(i)}{\le} L_q \mathop{\mathbb{E}}_{\sigma}\left[\sup_{h,h'\in H} \frac{2}{m}\sum_{i=1}^{m}\sigma_i \sup_{\|a_i - x_i\|_r \le \epsilon} \left|h(a_i) - h'(x_i)\right|\right],
\end{aligned} \tag{B.28}$$

where (i) is an application of Talagrand's contraction lemma. L_q is a data-dependent constant defined in (B.15).

To simplify the following derivation, we transfer the $\frac{2}{m}$ to the left. Then we have:

$$\frac{m}{2}\widehat{\mathcal{R}}_S(\widetilde{L}_H) \le L_q \mathop{\mathbb{E}}_{\sigma}\left[\sup_{h,h'\in H}\sum_{i=1}^{m}\sigma_i \sup_{\|a_i - x_i\|_r \le \epsilon} \left|h(a_i) - h'(x_i)\right|\right]. \tag{B.29}$$

Due to the non-linearity of neural networks, we can not follow the linear model and directly remove the *sup* by Hölder's inequality. Our solution is based on calculating the covering number of the set $\mathbb{T}$,

$$\mathbb{T} = \left\{ \theta \,\middle|\, \theta_i = \sup_{\|a_i - x_i\|_r \le \epsilon} \left|h(a_i) - h'(x_i)\right|;\ h, h' \in H;\ i \in [m] \right\}. \tag{B.30}$$

Let

$$X_{h,h'} := X_\theta = \sum_{i=1}^{m}\sigma_i \sup_{\|a_i - x_i\|_r \le \epsilon} \left|h(a_i) - h'(x_i)\right|$$

and

$$X_{\tilde{h},\tilde{h}'} := X_{\tilde{\theta}} = \sum_{i=1}^{m}\sigma_i \sup_{\|a_i - x_i\|_r \le \epsilon} \left|\tilde{h}(a_i) - \tilde{h}'(x_i)\right|.$$

We view (h, h') and $(\tilde{h}, \tilde{h}')$ as pairs of parameters from $H \bigtriangleup H$ and yields a Rademacher process $\{X_{h,h'}, (h, h') \in H \bigtriangleup H\}$. Then we can convert (B.29) to

$$\frac{m}{2}\widehat{\mathcal{R}}_S(\widetilde{L}_H) \le L_q \mathop{\mathbb{E}}_{\sigma}\left[\sup_{h,h'\in H} X_{h,h'}\right]. \tag{B.31}$$

Since the Rademacher process is a sub-Gaussian process with respect to the Euclidean metric, **we then apply Dudley's entropy integral bound (Lemma 7.4) to upper bound (B.31)**, with $\rho_X = \|\cdot\|_2$ and $\delta \to 0^+$. We obtain:

$$\frac{m}{2}\widehat{\mathcal{R}}_S(\widetilde{L}_H) \le L_q \cdot 32 \cdot \int_{0^+}^{D} \sqrt{log\mathcal{C}(u, \mathbb{T}, \|\cdot\|_2)}du, \tag{B.32}$$

where

$$D = \sup_{\theta,\tilde{\theta}\in\mathbb{T}} \rho_X(\theta,\tilde{\theta}) = \sup_{\theta,\tilde{\theta}\in\mathbb{T}} \|\theta-\tilde{\theta}\|_2.$$

(I). We first calculating the bound of diameter D.

$$\begin{aligned} D &= \sup_{h,h',\tilde{h},\tilde{h}'\in H} \sqrt{\sum_{i=1}^{m}\left[\sup_{\|a_i-x_i\|_r\le\varepsilon}\left|h(a_i)-h'(x_i)\right| - \sup_{\|a_i-x_i\|_r\le\varepsilon}\left|\tilde{h}(a_i)-\tilde{h}'(x_i)\right|\right]^2} \\ &\le \sup_{h,h',\tilde{h},\tilde{h}'\in H} \sqrt{m}\sup_{i\in[m]}\left[\sup_{\|a_i-x_i\|_r\le\varepsilon}\left|h(a_i)-h'(x_i)\right| - \sup_{\|a_i-x_i\|_r\le\varepsilon}\left|\tilde{h}(a_i)-\tilde{h}'(x_i)\right|\right] \\ &\le \sqrt{m}\sup_{h,h',\tilde{h},\tilde{h}'\in H;i\in[m]}\left[\sup_{\|a_i-x_i\|_r\le\varepsilon}\left|h(a_i)-h'(x_i)\right| - \sup_{\|a_i-x_i\|_r\le\varepsilon}\left|\tilde{h}(a_i)-\tilde{h}'(x_i)\right|\right]. \end{aligned} \tag{B.33}$$

Given $h,h'\in H$, let $a_i^* = \underset{\|a_i-x_i\|_r\le\varepsilon}{argmax}\,|h(a_i)-h'(x_i)|$, and let a_i^l be the output of a_i^* pass through the first $l-1$ layers. Similarly, given $\tilde{h},\tilde{h}'\in H$, let $\tilde{a}_i^* = \underset{\|a_i-x_i\|_r\le\varepsilon}{argmax}\,|\tilde{h}(a_i)-\tilde{h}'(x_i)|$, and let $\tilde{a}_i^l$ be the output of $\tilde{a}_i^*$ pass through the first $l-1$ layers. Then we have:

$$\begin{aligned} &\sup_{\|a_i-x_i\|_r\le\varepsilon}\left|h(a_i)-h'(x_i)\right| - \sup_{\|a_i-x_i\|_r\le\varepsilon}\left|\tilde{h}(a_i)-\tilde{h}'(x_i)\right| \\ &\overset{(i)}{=} \left|h(a_i^*)-h'(x_i)\right| - \left|\tilde{h}(\tilde{a}_i^*)-\tilde{h}'(x_i)\right| \\ &\overset{(ii)}{\le} \left|h(a_i^*)-h'(x_i)-\tilde{h}(\tilde{a}_i^*)+\tilde{h}'(x_i)\right| \\ &\overset{(iii)}{\le} \left|h(a_i^*)\right| + \left|h'(x_i)\right| + \left|\tilde{h}(\tilde{a}_i^*)\right| + \left|\tilde{h}'(x_i)\right|, \end{aligned} \tag{B.34}$$

where (i) is from the definition of a_i^* and $\tilde{a}_i^*$; (ii) and (iii) are applications of the triangle inequality

$$||a|-|b|| \le |a\pm b| \le |a|+|b|.$$

For the perturbed terms $\left|h(a_i^*)\right|$ and $\left|\tilde{h}(\tilde{a}_i^*)\right|$, we have:

$$\begin{aligned} \left|h(a_i^*)\right| &= \|W_z g(W_{z-1}a_i^{z-1})\|_2 \\ &\overset{(i)}{\le} \|W_z\|_F\cdot\|g(W_{z-1}a_i^{z-1})\|_2 \\ &\overset{(ii)}{\le} \|W_z\|_F\cdot\|g(W_{z-1}a_i^{z-1})-g(0)\|_2 \\ &\overset{(iii)}{\le} L_g\|W_z\|_F\cdot\|W_{z-1}a_i^{z-1}\|_2 \\ &\overset{(iv)}{\le} L_g U_z\cdot\|W_{z-1}a_i^{z-1}\|_2 \\ &\le \dots \\ &\le L_g^{z-1}\prod_{l=2}^{z}U_l\|W_1 a_i^*\|_2 \\ &\le L_g^{z-1}\prod_{l=1}^{z}U_l\|a_i^*\|_2 \\ &\overset{(v)}{\le} L_g^{z-1}\prod_{l=1}^{z}U_l max\left\{1, d^{\frac{1}{2}-\frac{1}{r}}\right\}(\|X\|_{r,\infty}+\varepsilon), \end{aligned} \tag{B.35}$$

where (i) is from Lemma Appendix B.8; (ii) is from the property $g(0)=0$; (iii) is due to the L_g-Lipschitz property of g; (iv) is due to $\|W_l\|_F \le U_l$ for $l=1,\dots,z$; (v) is a result of Lemma Appendix B.7.

Similarly,

$$\begin{aligned} \left|\tilde{h}(\tilde{a}_i^*)\right| &= \|W_z g(W_{z-1}\tilde{a}_i^{z-1})\|_2 \\ &\le \|W_z\|_F\cdot\|g(W_{z-1}\tilde{a}_i^{z-1})\|_2 \\ &\le \|W_z\|_F\cdot\|g(W_{z-1}\tilde{a}_i^{z-1})-g(0)\|_2 \\ &\le L_g\|W_z\|_F\cdot\|W_{z-1}\tilde{a}_i^{z-1}\|_2 \\ &\le L_g U_z\cdot\|W_{z-1}\tilde{a}_i^{z-1}\|_2 \end{aligned}$$

$$
\begin{aligned}
&\leq \dots \\
&\leq L_g^{z-1} \prod_{l=2}^{z} U_l \|W_1 \tilde{a}_i^*\|_2 \\
&\leq L_g^{z-1} \prod_{l=1}^{z} U_l \|\tilde{a}_i^*\|_2 \\
&\leq L_g^{z-1} \prod_{l=1}^{z} U_l max\left\{1, d^{\frac{1}{2}-\frac{1}{r}}\right\} (\|X\|_{r,\infty} + \varepsilon).
\end{aligned} \tag{B.36}
$$

For the terms $\left|h'(x_i^*)\right|$, we have:

$$
\begin{aligned}
\left|h'(x_i)\right| &= \|W_z g(W_{z-1} x_i^{z-1})\|_2 \\
&\leq \dots \ (similar\ to\ the\ proof\ (B.35)) \\
&\leq L_g^{z-1} \prod_{l=1}^{z} U_l \|x_i^*\|_2 \\
&\leq L_g^{z-1} \prod_{l=1}^{z} U_l \|X\|_{2,\infty}.
\end{aligned} \tag{B.37}
$$

It is easy to verify that $\left|\tilde{h}'(\tilde{x}_i^*)\right|$ has the same upper bound.

$$
\left|\tilde{h}'(\tilde{x}_i^*)\right| \leq L_g^{z-1} \prod_{l=1}^{z} U_l \|X\|_{2,\infty}. \tag{B.38}
$$

Combining (B.35), (B.36), (B.37), (B.38) and (B.33), we obtain:

$$
D \triangleq 2\sqrt{m} L_g^{z-1} \prod_{l=1}^{z} U_l \left[max\left\{1, d^{\frac{1}{2}-\frac{1}{r}}\right\} (B_x^r + \varepsilon) + B_x^2\right], \tag{B.39}
$$

where $B_x^r = \|X\|_{r,\infty}$ and $B_x^2 = \|X\|_{2,\infty}$. Given the domain, B_x^r and B_x^2 are constants.

(II). Now, we turn to bound $logC(u, \mathbb{T}, \|\cdot\|_2)$

Main ideas: Recall the definition of covering number, $C(u, \mathbb{T}, \|\cdot\|_2)$ is the cardinality of the smallest u-cover of $\mathbb{T}$. So we can find an artificial u-cover of $\mathbb{T}$ to bound $C(u, \mathbb{T}, \|\cdot\|_2)$. Since θ is associated with (h, h') and (h, h') are associated with their matrices (W_l, W_l') for $l = 1, \dots, z$. We first construct sets C_l to be δ_l-covers of $\{\|W_l\|_F \leq U_l\}$ with respect to $\|\cdot\|_F$ for $l = 1, \dots, z$. Then, we show that by giving special values to δ_ls, we can obtain an u-cover of $\mathbb{T}$.

Let C_ls be δ_l-covers of $\{\|W_l\|_F \leq U_l\}$ for $l = 1, \dots, z$, which means

$$
\forall\ W_l\ s.t.\ \|W_l\|_F \leq U_l,\ \exists\ W_l^c \in C_l\ s.t.\ \|W_l - W_l^c\|_F \leq \delta_l. \tag{B.40}
$$

We can get the corresponding hypothesis class H^c and set $\mathbb{T}^c$ as follows.

$$
H^c = \left\{x \to W_z^c g(...g(W_1^c x)), W_l^c \in C_l, l = 1, ..., z\right\}, \tag{B.41}
$$

$$
\mathbb{T}^c = \left\{ \theta^c \ \middle| \theta_i^c = \sup_{\|a_i - x_i\|_r \leq \varepsilon} \left|h_c(a_i) - h_c'(x_i)\right|;\ h_c, h_c' \in H^c;\ i \in [m] \right\}. \tag{B.42}
$$

For any $\theta \in \mathbb{T}$, we now calculate the smallest distance to the set $\mathbb{T}^c$, namely

$$
\sup_{\theta \in \mathbb{T}} \inf_{\theta^c \in \mathbb{T}^c} \|\theta - \theta^c\|_2
$$

By the definition of $\mathbb{T}$ and $\mathbb{T}^c$ in (B.30) and (B.42), we have:

$$
\begin{aligned}
\|\theta - \theta^c\|_2 &= \sqrt{\sum_{i=1}^{m} \left(\sup_{\|a_i - x_i\|_r \leq \varepsilon} \left|h(a_i) - h'(x_i)\right| - \sup_{\|a_i - x_i\|_r \leq \varepsilon} \left|h_c(a_i) - h_c'(x_i)\right| \right)^2} \\
&\leq \sqrt{m} \sup_{i \in [m]} \left| \sup_{\|a_i - x_i\|_r \leq \varepsilon} \left|h(a_i) - h'(x_i)\right| - \sup_{\|a_i - x_i\|_r \leq \varepsilon} \left|h_c(a_i) - h_c'(x_i)\right| \right|.
\end{aligned} \tag{B.43}
$$

Owing to (B.40), given a pair of (h, h') in H, there is a corresponding pair of (h_c, h_c') in H_c, such that:

$$
\begin{aligned}
&h_c = W_z^c g(\dots g(W_1^c x)) \\
&h_c' = W_z^{c'} g(\dots g(W_1^{c'} x)) \\
&W_l^c \in C_l,\ \|W_l - W_l^c\|_F \le \delta_l,\ l = 1, \dots, z \\
&W_l^{c'} \in C_l,\ \|W_l' - W_l^{c'}\|_F \le \delta_l,\ l = 1, \dots, z
\end{aligned} \tag{B.44}
$$

For any (x_i, a_i), $i = 1, \dots, m$. We now bound the term

$$
\left| \sup_{\|a_i - x_i\|_r \le \epsilon} \left| h(a_i) - h'(x_i) \right| - \sup_{\|a_i - x_i\|_r \le \epsilon} \left| h_c(a_i) - h_c'(x_i) \right| \right|.
$$

Let

$$
a_i^* = \operatorname*{argmax}_{\|a_i - x_i\|_r \le \epsilon} |h(a_i) - h'(x_i)|
$$

and

$$
a_i^c = \operatorname*{argmax}_{\|a_i - x_i\|_r \le \epsilon} |h_c(a_i) - h_c'(x_i)|.
$$

Then we have:

$$
\begin{aligned}
&\left| \sup_{\|a_i - x_i\|_r \le \epsilon} \left| h(a_i) - h'(x_i) \right| - \sup_{\|a_i - x_i\|_r \le \epsilon} \left| h_c(a_i) - h_c'(x_i) \right| \right| \\
&= \left| |h(a_i^*) - h'(x_i)| - |h_c(a_i^c) - h_c'(x_i)| \right| \\
&\le \left| |h(s_i) - h'(x_i)| - |h_c(s_i) - h_c'(x_i)| \right|,
\end{aligned} \tag{B.45}
$$

where

$$
s_i = \begin{cases} a_i^*, & if\ |h(a_i^*) - h'(x_i)| \ge |h_c(a_i^c) - h_c'(x_i)| \\ a_i^c, & if\ |h(a_i^*) - h'(x_i)| < |h_c(a_i^c) - h_c'(x_i)| \end{cases}.
$$

To bound the term in (B.45), we introduce a new function $\gamma_{l_1,l_2}(\cdot)$ and $\gamma'_{l_1,l_2}(\cdot)$, namely

$$
\begin{aligned}
&\gamma_{l_1,l_2}(x) = W_{l_2} g(W_{l_2-1} g(\dots W_{l_1+1} g(W_{l_1}^c \dots g(W_1^c x)))), \\
&\gamma'_{l_1,l_2}(x) = W'_{l_2} g(W'_{l_2-1} g(\dots W'_{l_1+1} g(W_{l_1}^{c'} \dots g(W_1^{c'} x)))).
\end{aligned}
$$

The function $\gamma_{l_1,l_2}(\cdot)$ has two types of weight matrices. For the layers $1 \le l \le l_1$, the matrix is W_l; for the layers $l_1 + 1 \le l \le l_2$, the matrix is $W_l^c \in C_l$. The function $\gamma'_{l_1,l_2}(\cdot)$ is defined similarly. Then we have $h(x) = \gamma_{0,z}(x)$, $h'(x) = \gamma'_{0,z}(x)$, $h_c(x) = \gamma_{z,z}(x)$ and $h_c'(x) = \gamma'_{z,z}(x)$.

We can decompose

$$
\begin{aligned}
&\left| |h(s_i) - h'(x_i)| - |h_c(s_i) - h_c'(x_i)| \right| \\
&= \left| |\gamma_{0,z}(s_i) - \gamma'_{0,z}(x_i)| - |\gamma_{z,z}(s_i) - \gamma'_{z,z}(x_i)| \right| \\
&\overset{(i)}{\le} \left| \gamma_{0,z}(s_i) - \gamma_{z,z}(s_i) - \gamma'_{0,z}(x_i) + \gamma'_{z,z}(x_i) \right| \\
&\overset{(ii)}{\le} \left| \gamma_{0,z}(s_i) - \gamma_{z,z}(s_i) \right| + \left| \gamma'_{0,z}(x_i) - \gamma'_{z,z}(x_i) \right| \\
&\overset{(iii)}{\le} \left| \gamma_{0,z}(s_i) - \gamma_{1,z}(s_i) \right| + \dots + \left| \gamma_{z-1,z}(s_i) - \gamma_{z,z}(s_i) \right| \\
&\quad + \left| \gamma'_{0,z}(x_i) - \gamma'_{z,z}(x_i) \right| + \dots + \left| \gamma'_{z-1,z}(x_i) - \gamma'_{z,z}(x_i) \right|,
\end{aligned} \tag{B.46}
$$

where (i), (ii) and (iii) are applications of triangle inequality $\left| |a| - |b| \right| \le |a \pm b| \le |a| + |b|$.

To bound $\left| |h(s_i) - h'(x_i)| - |h_c(s_i) - h_c'(x_i)| \right|$, we first calculate the single term in (B.46), namely, for $l = 1, \dots, z$,

$$
\begin{aligned}
\left| \gamma_{l-1,z}(s_i) - \gamma_{l,z}(s_i) \right| &= \left| W_z g(\gamma_{l-1,z-1}(s_i)) - W_z g(\gamma_{l,z-1}(s_i)) \right| \\
&\overset{(i)}{\le} \|W_z\|_F \cdot \|g(\gamma_{l-1,z-1}(s_i)) - g(\gamma_{l,z-1}(s_i))\|_2 \\
&\overset{(ii)}{\le} L_g U_z \|\gamma_{l-1,z-1}(s_i) - \gamma_{l,z-1}(s_i)\|_2
\end{aligned}
$$

$$
\begin{aligned}
&= L_g U_z \|W_{z-1} g(\gamma_{l-1,z-2}(s_i)) - W_{z-1} g(\gamma_{l,z-2}(s_i))\|_2 \\
&\leq \dots \\
&\leq L_g^{z-l} \prod_{j=l+1}^{z} U_j \|W_l g(\gamma_{l-1,l-1}(s_i)) - W_l^c g(\gamma_{l-1,l-1}(s_i))\|_2 \\
&\leq L_g^{z-l} \prod_{j=l+1}^{z} U_j \|(W_l - W_l^c) g(\gamma_{l-1,l-1}(s_i))\|_2 \\
&\overset{(iii)}{\leq} L_g^{z-l} \prod_{j=l+1}^{z} U_j \|W_l - W_l^c\|_F \cdot \|g(\gamma_{l-1,l-1}(s_i))\|_2 \\
&\overset{(iv)}{\leq} L_g^{z-l} \prod_{j=l+1}^{z} U_j \delta_l \|g(\gamma_{l-1,l-1}(s_i))\|_2,
\end{aligned} \tag{B.47}
$$

where (i) and (iii) come from Lemma Appendix B.8; (ii) is from (24) and the L_g-Lipschitz property of function $g(\cdot)$; (iv) is from (B.44)

Then, we bound the term $\|g(\gamma_{l-1,l-1}(s_i))\|_2$:

$$
\begin{aligned}
\|g(\gamma_{l-1,l-1}(s_i))\|_2 &= \|g(\gamma_{l-1,l-1}(s_i)) - g(0)\|_2 \\
&\overset{(i)}{\leq} L_g \|\gamma_{l-1,l-1}(s_i)\|_2 \\
&= L_g \|W_{l-1}^c g(\gamma_{l-2,l-2}(s_i))\|_2 \\
&\overset{(ii)}{\leq} L_g \|W_{l-1}^c\|_F \cdot \|g(\gamma_{l-2,l-2}(s_i))\|_2 \\
&\overset{(iii)}{\leq} L_g U_{l-1} \cdot \|g(\gamma_{l-2,l-2}(s_i))\|_2 \\
&\leq \dots \\
&\leq L_g^{l-1} \prod_{j=1}^{l-1} U_j \cdot \|s_i\|_2 \\
&\overset{(iv)}{\leq} L_g^{l-1} \prod_{j=1}^{l-1} U_j \cdot max\left\{1, d^{\frac{1}{2}-\frac{1}{r}}\right\} (B_x^r + \varepsilon),
\end{aligned} \tag{B.48}
$$

where (i) is because the function $g(\cdot)$ is L_g-Lipschitz; (ii) is from Lemma Appendix B.8; (iii) is from (24); (iv) is an application of Lemma Appendix B.7.

Combining (B.47) and (B.48), we have:

$$
\left|\gamma_{l-1,z}(s_i) - \gamma_{l,z}(s_i)\right| \leq L_g^{z-1} \frac{\prod_{j=1}^{z} U_j}{U_l} \delta_l max\left\{1, d^{\frac{1}{2}-\frac{1}{r}}\right\} (B_x^r + \varepsilon). \tag{B.49}
$$

Similarly to the procedure of (B.47) and (B.48), we can deduce:

$$
\left|\gamma'_{l-1,z}(x_i) - \gamma'_{l,z}(x_i)\right| \leq L_g^{z-1} \frac{\prod_{j=1}^{z} U_j}{U_l} \delta_l B_x^2. \tag{B.50}
$$

Combining (B.45), (B.46), (B.49) and (B.50), we obtain:

$$
\begin{aligned}
&\left| \sup_{\|a_i - x_i\|_r \leq \varepsilon} \left|h(a_i) - h'(x_i)\right| - \sup_{\|a_i - x_i\|_r \leq \varepsilon} \left|h_c(a_i) - h'_c(x_i)\right| \right| \\
&\leq \sum_{l=1}^{z} L_g^{z-1} \frac{\prod_{j=1}^{z} U_j}{U_l} \delta_l \left[max\left\{1, d^{\frac{1}{2}-\frac{1}{r}}\right\} (B_x^r + \varepsilon) + B_x^2 \right] \\
&= \sum_{l=1}^{z} \frac{D\delta_l}{2U_l\sqrt{m}}.
\end{aligned} \tag{B.51}
$$

Combining (B.43) and (B.51), we have:

$$
\sup_{\theta \in \mathbb{T}} \inf_{\theta^c \in \mathbb{T}^c} \|\theta - \theta^c\|_2 \leq \sum_{l=1}^{z} \frac{D\delta_l}{2U_l}.
$$

Let

$$\delta_l = \frac{2U_l u}{zD},\ l = 1, \ldots, z, \tag{B.52}$$

we have

$$\sup_{\theta \in \mathbb{T}} \inf_{\theta^c \in \mathbb{T}^c} \|\theta - \theta^c\|_2 \leq \sum_{l=1}^{z} \frac{D\delta_l}{2U_l} \leq u.$$

In summary, by constructing a δ_l-cover of the set W_l, we obtain a u-cover of $\mathbb{T}$. We then upper bound the u-covering number $C(u, \mathbb{T}, \|\cdot\|_2)$.

$$\begin{aligned} C(u, \mathbb{T}, \|\cdot\|_2) \leq |\mathbb{T}^c| &\overset{(i)}{=} \left(\prod_{l=1}^{z} |C_l| \right)^2 \\ &\overset{(ii)}{\leq} \left(\prod_{l=1}^{z} \left(1 + \frac{2U_l}{\delta_l} \right)^{h_l h_{l-1}} \right)^2 \overset{(iii)}{=} \left(1 + \frac{zD}{u} \right)^{2\sum_{l=1}^{z} h_l h_{l-1}} \end{aligned} \tag{B.53}$$

where (i) is from the construction of $\mathbb{T}^c$; (ii) is due to Lemma 7.2; (iii) is from the (B.52). Then, we have:

$$log\ C(u, \mathbb{T}, \|\cdot\|_2) \leq 2 \sum_{l=1}^{z} h_l h_{l-1} log \left(1 + \frac{zD}{u} \right) \tag{B.54}$$

Then, we substitute (B.54) into (B.32), namely:

$$\begin{aligned} \frac{m}{2} \widehat{\mathcal{R}}_S(\widetilde{L}_H) &\leq L_q \cdot 32 \cdot \int_{0^+}^{D} \sqrt{log C(u, \mathbb{T}, \|\cdot\|_2)} du \\ &\leq 32 L_q \int_{0^+}^{D} \sqrt{2 \sum_{l=1}^{z} h_l h_{l-1} log \left(1 + \frac{zD}{u} \right)} du \\ &\leq 32 L_q \sqrt{2 \sum_{l=1}^{z} h_l h_{l-1}} \cdot \int_{0^+}^{D} \sqrt{\frac{zD}{u}} du \\ &\leq 64 L_q D \sqrt{2z \sum_{l=1}^{z} h_l h_{l-1}}. \end{aligned} \tag{B.55}$$

Recall the D we obtain in (B.39):

$$D \triangleq 2\sqrt{m} L_g^{z-1} \prod_{l=1}^{z} U_l \left[max \left\{ 1, d^{\frac{1}{2} - \frac{1}{r}} \right\} (B_x^r + \varepsilon) + B_x^2 \right].$$

Substituting this D into (B.55), we get:

$$\frac{m}{2} \widehat{\mathcal{R}}_S(\widetilde{L}_H) \leq 128 L_q L_g^{z-1} \prod_{l=1}^{z} U_l \sqrt{2mz \sum_{l=1}^{z} h_l h_{l-1}} \left[max \left\{ 1, d^{\frac{1}{2} - \frac{1}{r}} \right\} (B_x^r + \varepsilon) + B_x^2 \right]. \tag{B.56}$$

Finally, substituting the definitions of B_x^2 and B_x^r into the inequality,

$$\widehat{\mathcal{R}}_S(\widetilde{L}_H) \leq \frac{256}{\sqrt{m}} L_q L_g^{z-1} \prod_{l=1}^{z} U_l \sqrt{2z \sum_{l=1}^{z} h_l h_{l-1}} \left[max \left\{ 1, d^{\frac{1}{2} - \frac{1}{r}} \right\} (\|X\|_{r,\infty} + \varepsilon) + \|X\|_{2,\infty} \right]. \quad \square \tag{B.57}$$

B.11. Proposition 5.1

Proposition 5.1. Let P and Q be any distributions over $\mathcal{X}$. Assume the loss function ℓ is L_l-Lipschitz and the hypothesis $c \circ \psi$ is L_c-Lipschitz. We have

$$|disc_{adv}^{H_\psi}(P, Q) - disc_{adv}^{sur}(P, Q)| \leq 2\varepsilon L_l L_c.$$

Before we provide the proof, we first introduce the following lemma.

Lemma B.9. *Given any function $a, b : \Theta \to \mathbb{R}$, we have*

$$\sup_{\theta\in\Theta}|a(\theta)| - \sup_{\theta\in\Theta}|b(\theta)| \le \sup_{\theta\in\Theta}|a(\theta)-b(\theta)|$$

Proof of Lemma Appendix B.9. Based on the triangle inequality:

$$|a(\theta)| \le |a(\theta)-b(\theta)| + |b(\theta)|,$$

we have

$$\sup_{\theta\in\Theta}|a(\theta)| \le \sup_{\theta\in\Theta}(|a(\theta)-b(\theta)| + |b(\theta)|) \le \sup_{\theta\in\Theta}|a(\theta)-b(\theta)| + \sup_{\theta\in\Theta}|b(\theta)|$$

then we get

$$\sup_{\theta\in\Theta}|a(\theta)| - \sup_{\theta\in\Theta}|b(\theta)| \le \sup_{\theta\in\Theta}|a(\theta)-b(\theta)|. \quad \square$$

Now we can bound the difference between AD and the approximated AD as below.

Proof of Proposition 5.1. Recall the definition of AD and the approximated AD:

$$disc_{adv}^{H_\psi}(P,Q) = \sup_{c',c''\in\mathfrak{C}}\left|\mathop{\mathbb{E}}_{x\sim P}\left[\sup_{a\in N(x)}\ell\left(c'\circ\psi(a), c''\circ\psi(x)\right)\right] - \mathop{\mathbb{E}}_{x\sim Q}\left[\sup_{a\in N(x)}\ell\left(c'\circ\psi(a), c''\circ\psi(x)\right)\right]\right|,$$

and

$$disc_{adv}^{apx}(P,Q) = \sup_{c',c''\in\mathfrak{C}}\left|\mathop{\mathbb{E}}_{x\sim P}\left[\sup_{a\in N(x)}\ell\left(c'\circ\psi(a), c''\circ\psi(a)\right)\right] - \mathop{\mathbb{E}}_{x\sim Q}\left[\sup_{a\in N(x)}\ell\left(c'\circ\psi(a), c''\circ\psi(a)\right)\right]\right|.$$

Then, we turn to bound the difference between these two terms:

$$\begin{aligned}
&|disc_{adv}^{H_\psi}(P,Q) - disc_{adv}^{sur}(P,Q)| \\
&= \left|\sup_{c',c''\in\mathfrak{C}}\left|\mathop{\mathbb{E}}_{x\sim P}\left[\sup_{a\in N(x)}\ell\left(c'\circ\psi(a), c''\circ\psi(x)\right)\right] - \mathop{\mathbb{E}}_{x\sim Q}\left[\sup_{a\in N(x)}\ell\left(c'\circ\psi(a), c''\circ\psi(x)\right)\right]\right|\right. \\
&\quad \left. - \sup_{c',c''\in\mathfrak{C}}\left|\mathop{\mathbb{E}}_{x\sim P}\left[\sup_{a\in N(x)}\ell\left(c'\circ\psi(a), c''\circ\psi(a)\right)\right] - \mathop{\mathbb{E}}_{x\sim Q}\left[\sup_{a\in N(x)}\ell\left(c'\circ\psi(a), c''\circ\psi(a)\right)\right]\right|\right| \\
&\overset{(i)}{\le} \sup_{c',c''\in\mathfrak{C}}\left|\mathop{\mathbb{E}}_{x\sim P}\left[\sup_{a\in N(x)}\ell\left(c'\circ\psi(a), c''\circ\psi(x)\right)\right] - \mathop{\mathbb{E}}_{x\sim P}\left[\sup_{a\in N(x)}\ell\left(c'\circ\psi(a), c''\circ\psi(a)\right)\right]\right. \\
&\qquad \left. + \mathop{\mathbb{E}}_{x\sim Q}\left[\sup_{a\in N(x)}\ell\left(c'\circ\psi(a), c''\circ\psi(a)\right)\right] - \mathop{\mathbb{E}}_{x\sim Q}\left[\sup_{a\in N(x)}\ell\left(c'\circ\psi(a), c''\circ\psi(x)\right)\right]\right| \\
&\le \sup_{c',c''\in\mathfrak{C}}\left|\mathop{\mathbb{E}}_{x\sim P}\sup_{a\in N(x)}\left[\ell\left(c'\circ\psi(a), c''\circ\psi(x)\right) - \ell\left(c'\circ\psi(a), c''\circ\psi(a)\right)\right]\right. \\
&\qquad \left. + \mathop{\mathbb{E}}_{x\sim Q}\sup_{a\in N(x)}\left[\ell\left(c'\circ\psi(a), c''\circ\psi(a)\right) - \ell\left(c'\circ\psi(a), c''\circ\psi(x)\right)\right]\right| \\
&\le \sup_{c''\in\mathfrak{C}}\left|\mathop{\mathbb{E}}_{x\sim P}\sup_{a\in N(x)}|c''\circ\psi(x) - c''\circ\psi(a)|L_l + \mathop{\mathbb{E}}_{x\sim Q}\sup_{a\in N(x)}|c''\circ\psi(a) - c''\circ\psi(x)|L_l\right| \\
&\le 2\varepsilon L_l L_c,
\end{aligned}$$

where (i) is the application of Lemma Appendix B.9, L_l is the Lipschitz constant of $\ell(\cdot,\cdot)$ and L_c is the Lipschitz constant of $c\circ\psi(\cdot)$. $\square$

Appendix C. Additional experimental results

In this section, we present some additional results. The results of UDA methods implemented by ResNet-18 on Office-31 are shown in Table C.11. Table C.12 contains detailed results of the ablation study.

From Table C.11, we can see that our proposed method achieves the best adversarial accuracy on almost all transfer tasks of Office-31. For example, compared with the runner-up baseline, our method improves the adversarial accuracy by 9.7% on task **A** → **D** under PGD-20 attack with ℓ_∞-norm bounded perturbation of radius $\varepsilon = 3/255$. Moreover, similar to the results on ResNet-50, our

Table C.11
Adversarial accuracy (%) of various methods on Office-31. All methods are based on ResNet-18 [22]. The adversarial examples are generated by PGD-20 attack [35] with ℓ_∞-norm bounded perturbation of radius $\varepsilon = 3/255$. Avg. denotes the average accuracy over all tasks. The benchmark methods are trained with adversarial examples.

Method	A→D	A→W	D→W	D→A	W→A	W→D	Avg.
DIRT-T^{adv}	10.6	10.9	33.5	12.6	15.1	35.1	19.6
MCD^{adv}	9.8	12.1	35.7	2.3	6.5	51.9	19.7
SSAT	5.0	2.4	59.0	7.8	18.1	66.7	26.5
MDD^{adv}	24.9	33.1	79.6	14.9	26.5	88.6	44.6
BSP^{adv}	24.7	31.1	81.9	18.2	27.3	91.2	45.7
DANN^{adv}	25.7	32.3	87.0	13.7	27.3	95.2	46.9
RST	28.9	31.8	83.9	28.4	31.0	87.8	48.6
MCC^{adv}	33.3	36.6	84.3	21.9	28.5	**95.6**	50.0
Ours	**43.0**	**44.4**	**89.4**	**31.2**	**37.7**	94.4	**56.7**

Table C.12
Ablation study (%) of various methods on Office-31. All methods are based on ResNet-18 [22]. The adversarial examples are generated by PGD-20 attack [35] with ℓ_∞-norm bounded perturbation of radius $\varepsilon = 3/255$. Avg. denotes the average accuracy over all tasks.

Source	Discrepancy	A→D	A→W	D→W	D→A	W→A	W→D	Avg.
clean	–	0.0	0.0	0.0	0.0	0.0	0.0	0.0
clean	$H \bigtriangleup H$	0.0	0.0	0.0	0.0	0.0	0.0	0.0
adversarial	–	15.3	17.7	66.2	11.2	17.2	76.9	34.1
adversarial	$H \bigtriangleup H$	24.5	32.3	88.1	18.3	24.8	**95.6**	47.3
adversarial	AD	**43.0**	**44.4**	**89.4**	**31.2**	**37.7**	94.4	**56.7**

method based on ResNet-18 clearly boosts the adversarial robustness on the relatively difficult tasks **D** → **A** and **W** → **A** where the source domain is very small.

Data availability

Data will be made available on request.

References

[1] J. Alayrac, J. Uesato, P. Huang, A. Fawzi, R. Stanforth, P. Kohli, Are labels required for improving adversarial robustness?, in: NeurIPS, 2019, pp. 12192–12202.
[2] M. Awais, F. Zhou, H. Xu, L. Hong, P. Luo, S. Bae, Z. Li, Adversarial robustness for unsupervised domain adaptation, in: ICCV, IEEE, 2021, pp. 8548–8557.
[3] P. Awasthi, N. Frank, M. Mohri, Adversarial learning guarantees for linear hypotheses and neural networks, in: ICML, 2020, pp. 431–441.
[4] P.L. Bartlett, S. Mendelson, Rademacher and gaussian complexities: risk bounds and structural results, J. Mach. Learn. Res. 3 (2002) 463–482.
[5] B.R. Bartoldson, J. Diffenderfer, K. Parasyris, B. Kailkhura, Adversarial robustness limits via scaling-law and human-alignment studies, in: ICML, 2024.
[6] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, J.W. Vaughan, A theory of learning from different domains, Mach. Learn. 79 (2010) 151–175.
[7] S. Ben-David, J. Blitzer, K. Crammer, F. Pereira, Analysis of representations for domain adaptation, in: NeurIPS, 2006, pp. 137–144.
[8] Q. Cai, C. Liu, D. Song, Curriculum adversarial training, in: IJCAI, 2018, pp. 3740–3747.
[9] N. Carlini, D.A. Wagner, Towards evaluating the robustness of neural networks, in: SP, 2017, pp. 39–57.
[10] Y. Carmon, A. Raghunathan, L. Schmidt, J.C. Duchi, P. Liang, Unlabeled data improves adversarial robustness, in: NeurIPS, 2019, pp. 11190–11201.
[11] D. Chen, H. Hu, Q. Wang, Y. Li, C. Wang, C. Shen, Q. Li, CARTL: cooperative adversarially-robust transfer learning, in: ICML, 2021, pp. 1640–1650.
[12] X. Chen, S. Wang, M. Long, J. Wang, Transferability vs. discriminability: batch spectral penalization for adversarial domain adaptation, in: ICML, 2019, pp. 1081–1090.
[13] Y. Chen, W. Liu, A theory of transfer-based black-box attacks: explanation and implications, in: NeurIPS, 2023, pp. 13887–13907.
[14] F. Croce, M. Hein, Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, in: ICML, 2020, pp. 2206–2216.
[15] D. Cullina, A.N. Bhagoji, P. Mittal, Pac-learning in the presence of adversaries, in: NeurIPS, 2018, pp. 228–239.
[16] M. Dinu, M. Holzleitner, M. Beck, H.D. Nguyen, A. Huber, H. Eghbal-zadeh, B.A. Moser, S.V. Pereverzyev, S. Hochreiter, W. Zellinger, Addressing parameter choice issues in unsupervised domain adaptation by aggregation, in: ICLR, 2023.
[17] A. Farahani, S. Voghoei, K. Rasheed, H.R. Arabnia, A brief review of domain adaptation, preprint, arXiv:2010.03978, 2020.
[18] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V.S. Lempitsky, Domain-adversarial training of neural networks, J. Mach. Learn. Res. 17 (2016) 59:1–59:35.
[19] Q. Gao, X. Wang, Theoretical investigation of generalization bounds for adversarial learning of deep neural networks, J. Stat. Theory Pract. (2021).
[20] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: ICLR, 2015.
[21] A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf, A.J. Smola, A kernel method for the two-sample-problem, in: NeurIPS, 2006, pp. 513–520.
[22] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016, pp. 770–778.
[23] J. Huang, D. Guan, A. Xiao, S. Lu, RDA: robust domain adaptation via fourier adversarial attacking, in: ICCV, 2021, pp. 8968–8979.
[24] R. Huang, B. Xu, D. Schuurmans, C. Szepesvári, Learning with a Strong Adversary, 2015.
[25] Y. Jin, X. Wang, M. Long, J. Wang, Minimum class confusion for versatile domain adaptation, in: ECCV, 2020, pp. 464–480.
[26] G. Kang, L. Jiang, Y. Yang, A.G. Hauptmann, Contrastive adaptation network for unsupervised domain adaptation, in: CVPR, 2019, pp. 4893–4902.
[27] J. Khim, P. Loh, Adversarial risk bounds for binary classification via function transformation, preprint, arXiv:1810.09519, 2018.
[28] A. Kurakin, I.J. Goodfellow, S. Bengio, Adversarial examples in the physical world, in: ICLR, 2017.

[29] M. Ledoux, M. Talagrand, Probability in Banach Spaces: Isoperimetry and Processes, vol. 23, 1991.
[30] D. Li, Y. Yang, Y. Song, T.M. Hospedales, Deeper, broader and artier domain generalization, in: ICCV, 2017, pp. 5543–5551.
[31] S. Lo, V.M. Patel, Exploring adversarially robust training for unsupervised domain adaptation, in: ACCV, 2022, pp. 561–577.
[32] M. Long, Z. Cao, J. Wang, M.I. Jordan, Conditional adversarial domain adaptation, in: NeurIPS, 2018, pp. 1647–1657.
[33] X. Ma, Z. Wang, W. Liu, On the tradeoff between robustness and fairness, in: NeurIPS, 2022, pp. 26230–26241.
[34] L. van der Maaten, G. Hinton, Visualizing data using t-sne, J. Mach. Learn. Res. (2008).
[35] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in: ICLR, 2018.
[36] Y. Mansour, M. Mohri, A. Rostamizadeh, Domain adaptation: learning bounds and algorithms, in: COLT, 2009.
[37] S. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Deepfool: a simple and accurate method to fool deep neural networks, in: CVPR, 2016, pp. 2574–2582.
[38] W. Mustafa, Y. Lei, M. Kloft, On the generalization analysis of adversarial learning, in: ICML, 2022, pp. 16174–16196.
[39] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E.Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: an imperative style, high-performance deep learning library, in: NeurIPS, 2019, pp. 8024–8035.
[40] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, K. Saenko, Visda: the visual domain adaptation challenge, preprint, arXiv:1710.06924, 2017.
[41] Z. Qiu, Y. Zhang, H. Lin, S. Niu, Y. Liu, Q. Du, M. Tan, Source-free domain adaptation via avatar prototype generation and adaptation, in: IJCAI, 2021, pp. 2921–2927.
[42] A. Raghunathan, J. Steinhardt, P. Liang, Certified defenses against adversarial examples, in: ICLR, 2018.
[43] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M.S. Bernstein, A.C. Berg, L. Fei-Fei, Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (2015) 211–252.
[44] K. Saenko, B. Kulis, M. Fritz, T. Darrell, Adapting visual category models to new domains, in: ECCV, 2010, pp. 213–226.
[45] K. Saito, D. Kim, P. Teterwak, S. Sclaroff, T. Darrell, K. Saenko, Tune it the right way: unsupervised validation of domain adaptation via soft neighborhood density, in: ICCV, 2021, pp. 9164–9173.
[46] K. Saito, K. Watanabe, Y. Ushiku, T. Harada, Maximum classifier discrepancy for unsupervised domain adaptation, in: CVPR, 2018, pp. 3723–3732.
[47] H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, A. Madry, Do adversarially robust imagenet models transfer better?, in: NeurIPS, 2020.
[48] S. Sankaranarayanan, Y. Balaji, C.D. Castillo, R. Chellappa, Generate to adapt: aligning domains using generative adversarial networks, in: CVPR, 2018, pp. 8503–8512.
[49] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, A. Madry, Adversarially robust generalization requires more data, in: NeurIPS, 2018, pp. 5019–5031.
[50] A. Shafahi, P. Saadatpanah, C. Zhu, A. Ghiasi, C. Studer, D.W. Jacobs, T. Goldstein, Adversarially robust transfer learning, in: ICLR, 2020.
[51] U. Shaham, Y. Yamada, S. Negahban, Understanding adversarial training: increasing local stability of neural nets through robust optimization, preprint, arXiv: 1511.05432, 2015.
[52] L. Shi, W. Liu, Adversarial self-training improves robustness and generalization for gradual domain adaptation, in: NeurIPS, 2023, pp. 37321–37333.
[53] L. Shi, W. Liu, A closer look at curriculum adversarial training: from an online perspective, in: AAAI, 2024, pp. 14973–14981.
[54] H. Shimodaira, Improving predictive inference under covariate shift by weighting the log-likelihood function, J. Stat. Plan. Inference 90 (2000) 227–244.
[55] R. Shu, H.H. Bui, H. Narui, S. Ermon, A DIRT-T approach to unsupervised domain adaptation, in: ICLR, 2018.
[56] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I.J. Goodfellow, R. Fergus, Intriguing properties of neural networks, in: ICLR, 2014.
[57] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, in: CVPR, 2017, pp. 2962–2971.
[58] H. Venkateswara, J. Eusebio, S. Chakraborty, S. Panchanathan, Deep hashing network for unsupervised domain adaptation, in: CVPR, 2017, pp. 5385–5394.
[59] M.J. Wainwright, High-Dimensional Statistics: A Non-asymptotic Viewpoint, vol. 48, 2019.
[60] Q. Wu, H. Liu, Unsupervised domain adaptation for semantic segmentation using depth distribution, in: NeurIPS, 2022.
[61] J. Xiao, Y. Fan, R. Sun, Z. Luo, Adversarial rademacher complexity of deep neural networks, preprint, arXiv:2211.14966, 2022.
[62] J. Xiao, Y. Fan, R. Sun, Z.Q. Luo, Adversarial rademacher complexity of deep neural networks, preprint, arXiv:2211.14966, 2022.
[63] J. Xu, W. Liu, On robust multiclass learnability, in: NeurIPS, 2022, pp. 32412–32423.
[64] J. Yang, R. Xu, R. Li, X. Qi, X. Shen, G. Li, L. Lin, An adversarial perturbation oriented domain adaptation approach for semantic segmentation, in: AAAI, 2020, pp. 12613–12620.
[65] D. Yin, K. Ramchandran, P.L. Bartlett, Rademacher complexity for adversarially robust generalization, in: ICML, 2019, pp. 7085–7094.
[66] K. You, X. Wang, M. Long, M.I. Jordan, Towards accurate model selection in deep unsupervised domain adaptation, in: ICML, 2019, pp. 7124–7133.
[67] R. Zhai, T. Cai, D. He, C. Dan, K. He, J.E. Hopcroft, L. Wang, Adversarially robust generalization just requires more unlabeled data, preprint, arXiv:1906.00555, 2019.
[68] H. Zhang, Y. Yu, J. Jiao, E.P. Xing, L.E. Ghaoui, M.I. Jordan, Theoretically principled trade-off between robustness and accuracy, in: ICML, 2019, pp. 7472–7482.
[69] Y. Zhang, T. Liu, M. Long, M.I. Jordan, Bridging theory and algorithm for domain adaptation, in: ICML, 2019, pp. 7404–7413.
[70] Z. Zhou, W. Liu, Sample complexity for distributionally robust learning under chi-square divergence, J. Mach. Learn. Res. 24 (2023) 1–27.
[71] X. Zou, W. Liu, Generalization bounds for adversarial contrastive learning, J. Mach. Learn. Res. 24 (2023) 1–54.
[72] X. Zou, W. Liu, On the adversarial robustness of out-of-distribution generalization models, in: NeurIPS, 2023, pp. 68908–68938.