Taxonomy-Structured Domain Adaptation

Tianyi Liu [*1] Zihao Xu [*1] Hao He [2] Guang-Yuan Hao [3] Guang-He Lee Hao Wang [1]

Abstract

Domain adaptation aims to mitigate distribution shifts among different domains. However, traditional formulations are mostly limited to categorical domains, greatly simplifying nuanced domain relationships in the real world. In this work, we tackle a generalization with taxonomy-structured domains, which formalizes domains with nested, hierarchical similarity structures such as animal species and product catalogs. We build on the classic adversarial framework and introduce a novel *taxonomist*, which competes with the adversarial discriminator to preserve the taxonomy information. The equilibrium recovers the classic adversarial domain adaptation's solution if given a non-informative domain taxonomy (e.g., a flat taxonomy where all leaf nodes connect to the root node) while yielding non-trivial results with other taxonomies. Empirically, our method achieves state-of-the-art performance on both synthetic and real-world datasets with successful adaptation. Code is available at https://github.com/Wang-ML-Lab/TSDA.

1. Introduction

Learning generalizable models is a central goal in machine learning. The majority of the literature has been devoted to the standard i.i.d. setting where training data and testing data are assumed to have the same distribution. However, many real world problems inherently exhibit distributional shifts. For example, to make commercial impacts, a company must be able to reach brand-new groups of users (Ding et al., 2022), whose data are not likely to be the same as the data used to fit production models. As a rapidly emerging subfield, transfer learning aims to tackle this problem (Pan & Yang, 2010; Zhuang et al., 2020).

A series of work has attempted to accomplish transfer learning via domain adaptation (Pan & Yang, 2009; Pan et al., 2010; Long et al., 2018; Saito et al., 2018; Sankaranarayanan et al., 2018; Zhang et al., 2019; Peng et al., 2019; Chen et al., 2019; Dai et al., 2019; Nguyen-Meidine et al., 2021), i.e., learning representations whose distribution is well aligned across different domains. This is typically done through learning against an adversary who tries to distinguish different domains. While the approach is widely studied both theoretically and empirically (Redko et al., 2020; Ben-David et al., 2010; Zhao et al., 2018; Zhang et al., 2019; Zhao et al., 2019), the representation of domains is almost entirely limited to simple categories. This is problematic since categorical variables do not admit any meaningful measurements of similarity or distance. Indeed, an ideal transfer learning approach should be able to control the transferability depending on the similarity across domains. For instance, if we treat dog breeds as domains, an ideal video segmentation system for bassets should behave more similarly for beagles than for pomeranians. While existing adversarial approach mitigates the shifts among domains, the solution can still be inadequate since it ignores the potential structure among domains.

To this end, we extend domain adaptation to taxonomy-structured domains (see Fig. 2 and Fig. 3 for some example taxonomies). Taxonomies abound in our culture for categorizing items, ranging from biological studies to library classification systems. Mathematically, "taxonomy" is a *nested* hierarchical representation (Rangapuram et al., 2023). Each *node* in the taxonomy specifies a level of *invariance* that holds for the domains within the node, which can be further broken down to a lower level of invariance in its child nodes. Finally, the lowest level of invariance is simply a single domain. We emphasize that a taxonomy of domains is different from a tree of domains, as a non-leaf node in a taxonomy typically involves multiple domains. The specification of invariance among domains thus becomes nested, which can be naturally translated to the induced similarities. Indeed, the similarity between two domains can be easily gauged through their common ancestors, which are also nested.

In this work, we propose Taxonomy-Structured Domain Adaptation (TSDA), a plug-in extension of adversarial do-

[*] Equal contribution [1] Rutgers University [2] Massachusetts Institute of Technology [3] The Chinese University of Hong Kong. Correspondence to: Tianyi Liu <tl579@scarletmail.rutgers.edu>, Zihao Xu <zihao.xu@rutgers.edu>, Hao Wang <hw488@cs.rutgers.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

main adaption by introducing a novel *taxonomist*, who guides the representation to exhibit the given domain taxonomy. This enables representation learning with a flexible balance between domain similarity and domain invariance. Despite the obvious contradictory nature, it is evident that either extreme fails to capture some important inductive biases in learning. Indeed, absolute invariance to the domain prevents the model from leveraging similarities among domains to improve statistical efficiency, while purely relying on domain similarities can easily pick up spurious correlations among different domains. The flexibility allows us to achieve a suitable trade off according to the predictive task. We summarize our contributions are as follows:

- We identify the problem of adaptation across taxonomy-structured domains and develop taxonomy-structured domain adaptation (TSDA) as the first general DA method to address this problem.
- Our theoretical analysis shows that a natural extension of typical DA methods fails to take advantage of domain similarity reflected in a domain taxonomy and degenerates to uniform alignment.
- We further prove that TSDA retains typical DA methods' capability of uniform alignment when the domain taxonomy is non-informative, and balances domain similarity and domain invariance for other taxonomies.
- Empirical results show that our TSDA improves upon state-of-the-art DA methods on both synthetic and real-world datasets.

2. Related Work

Adversarial Domain Adaptation. There is a rich literature on domain adaptation (Pan & Yang, 2009; Pan et al., 2010; Long et al., 2018; Saito et al., 2018; Sankaranarayanan et al., 2018; Zhang et al., 2019; Peng et al., 2019; Chen et al., 2019; Dai et al., 2019; Nguyen-Meidine et al., 2021; Zou et al., 2018; Kumar et al., 2020; Prabhu et al., 2021; Maria Carlucci et al., 2017; Mancini et al., 2019; Tasar et al., 2020; Jin et al., 2022). To adapt a model across domains, existing methods typically align the encoding distributions of source and target domains, either by direct matching (Pan et al., 2010; Tzeng et al., 2014; Sun & Saenko, 2016; Peng et al., 2019; Nguyen-Meidine et al., 2021; Wang et al., 2020a) or adversarial training (Ganin et al., 2016; Zhao et al., 2017; Tzeng et al., 2017; Zhang et al., 2019; Kuroki et al., 2019; Chen et al., 2019; Dai et al., 2019). The adversarial domain adaptation framework becomes increasingly popular recently due to its solid theoretical foundation (Goodfellow et al., 2014; Ben-David et al., 2010; Zhao et al., 2018; Redko et al., 2020; Zhang et al., 2019; Zhao et al., 2019), efficient end-to-end implementation, and promising performance. Generally, these methods train an encoder to produce domain-invariant representations by trying to fool a discriminator that is trained to distinguish different domains; essentially they aim to perfectly align data from different domains in the encoding space, i.e., uniform alignment. Such uniform alignment can sometimes harms domain adaptation performance because it completely remove useful domain similarity information (e.g., information captured by a domain taxonomy) from the encoding. In contrast, our TSDA relaxes uniform alignment by introducing a taxonomist as the fourth player to recover the domain taxonomy from encodings (as shown in Fig. 1), thereby significantly improving adaptation performance.

Domain Adaptation Related to Taxonomies. Loosely related to our method are works related to both domain adaptation and taxonomies. For instance, Gong et al. (2022) [1] focuses on the *label* taxonomies and considers domain adaptation on semantic segmentation with one source domain and one target domain; it assumes the source domain and the target domain have different label taxonomies and aims to alleviate label shift (Wu et al., 2019) between these two domains. In contrast, TSDA considers a *completely different* setting. Specifically, Gong et al. (2022) adapts between *two domains* with different label distributions, where taxonomies are used to describe *labels relations*. Our TSDA adapts across *multiple domains* (e.g., with each species as a domain) according to a domain taxonomy (e.g., an animal taxonomy), where taxonomies are used to describe *domain relations*. Gong et al. (2022) is therefore *not applicable* to our setting. Also related to our work is graph-relational domain adaptation (GRDA) (Xu et al., 2022), which adapts across domains connected by a graph with each domain as a node, continuously indexed domain adaptation (CIDA) (Wang et al., 2020a), which adapts across continuously indexed domains, and variational domain indexing (VDI) (Xu et al., 2023), which infers domain indices in CIDA using a hierarchical Bayesian deep learning model (Wang et al., 2015; Wang & Yeung, 2016; Wang, 2017; Huang et al., 2019; Wang et al., 2019; Wang & Yeung, 2020; Ding et al., 2022). While a domain taxonomy can be reduced to a graph or continuous indices, it loses important hierarchical information and therefore often leads to suboptimal performance; this is verified by our empirical results in Sec. 5.

3. Method

3.1. Problem Setting and Notation

In domain adaptation, an input $\mathbf{x} \in \mathcal{X}$ (e.g., a bird image) is used in conjunction with an additional domain specification $u \in \mathcal{U} = \{1, 2, \ldots, N\}$ (e.g., the species) to predict a label $y \in \mathcal{Y}$ (e.g., the wing color). Here we consider an unsu-

[1] This paper has another version (Gong et al., 2021). We cited the official published version.

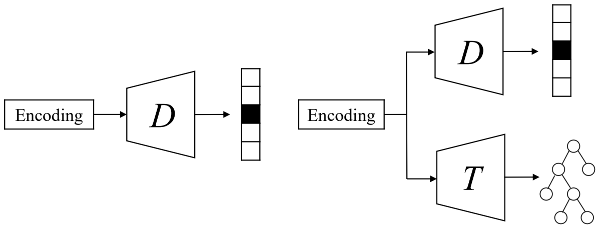

Figure 1. Difference between typical DA methods and TSDA. **Left:** In traditional DA methods, the discriminator classifies the domain index given an encoding. **Right:** In TSDA, the discriminator classifies the domain index while the taxonomist reconstructs the domain taxonomy given encodings of data from different domains.

pervised domain adaption setting with N domains, where labeled data $\{(\mathbf{x}_l^s, u_l^s, y_l^s)\}_{l=1}^n \subseteq \mathcal{X} \times \mathcal{U}_s \times \mathcal{Y}$ are only available in the *source* domains $\mathcal{U}_s \subseteq \mathcal{U}$, and the goal is to make predictions for the unlabeled data $\{(\mathbf{x}_l^t, u_l^t)\}_{l=1}^m \subseteq \mathcal{X} \times \mathcal{U}_t$ in the *target* domains $\mathcal{U}_t \subseteq \mathcal{U}$. Note that we do not require $\mathcal{U}_s \cap \mathcal{U}_t = \emptyset$. The goal is to predict the target-domain labels $\{(y_l^t)\}_{l=1}^m$ given source-domain labeled data and target-domain unlabeled data.

In this work, we assume that an additional taxonomy $\mathcal{T}$ is given; $\mathcal{T}$ specifies a hierarchical similarity structure over the domains $\mathcal{U}$. Formally, a taxonomy $\mathcal{T}$ can be represented as a (directed) tree:

- The root node is $\mathcal{U}$.
- In each layer, a parent $\mathcal{U}_p \subseteq \mathcal{U}$ is split into disjoint child nodes $\mathcal{U}_1^p, ..., \mathcal{U}_{n_p}^p$, where $\mathcal{U}_1^p \cup ... \cup \mathcal{U}_{n_p}^p = \mathcal{U}_p$.
- Each leaf node only contains one domain $\{u\}$.

We emphasize that a taxonomy of domains is different from a tree of domains, as a non-leaf node in a taxonomy does not (necessarily) correspond to a single domain.

3.2. Taxonomy-Structured Domain Adaptation

Overview. To balance representation learning between domain similarity and domain invariance, we develop a game-theoretic formulation with four players: 1) an encoder E that aims to produce the desired representation, 2) a discriminator D that prevents the encoder from picking up domain dependencies, 3) a taxonomist T that encourages the representation to preserve similarity information, and 4) a predictor F that produces predictions according to the encoder and therefore guides the representation to strike a balance for the prediction task of interest.

Before further elaboration on the game, note that one key challenge in the framework is to model the taxonomy–the key object representing similarity among domains. Indeed, taxonomy is a combinatorial object that seems naturally incompatible with the continuously distributed representations modeled by deep networks. To bridge the gap, we transform the taxonomy $\mathcal{T}$ to a *distance matrix* $\mathbf{A}$, where $\mathbf{A}_{ij}$ records the shortest distance between two domains i and j on the taxonomy $\mathcal{T}$. Below, we formally define the game.

Encoder. The encoder E leverages all the available information to build a representation $\mathbf{e}_l$. It takes as input the data $\mathbf{x}_l$, domain index u_l, and the distance matrix $\mathbf{A}$. To facilitate learning, we first leverage an intermediate domain embedding $\mathbf{z}_{u_l} = g(u_l, \mathbf{A})$ to capture the immediate dependency between u_l and $\mathbf{A}$:

$$\mathbf{e}_l = E(\mathbf{x}_l, u_l, \mathbf{A}) = f(\mathbf{x}_l, g(u_l, \mathbf{A})) = f(\mathbf{x}_l, \mathbf{z}_{u_l}),$$

where the encoder $E(\cdot)$ is defined by composition of $g(\cdot)$ and $f(\cdot)$. In principle, given sufficiently powerful f, every embeddings $\mathbf{z}_{u_l}$ works equally well as long as a bijection to the set of domains $\mathcal{U}$ can be formed. Here we use a simple pretraining procedure to obtain the domain embedding based on a reconstruction loss with respect to the domain distance matrix $\mathbf{A}$ (see more details in the Appendix).

Discriminator. The discriminator D aims to identify the domain from the encoding $\mathbf{e}_l$. This is realized as a function $D : \mathcal{Z} \to \mathcal{U}$ that minimizes a domain identification (classification) loss l_d:

$$L_d(D, E) \triangleq \mathbb{E}[l_d(D(E(\mathbf{x}_l, u_l, \mathbf{A}), u_l)],$$

where the expectation $\mathbb{E}$ is taken over the data distribution $p(\mathbf{x}, u)$; l_d is typically realized by cross-entropy.

Taxonomist. Similar to the discriminator D, the taxonomist T aims to recover taxonomy information within the encoding $\mathbf{e}_l$. Unlike the vanilla categorical domain index that can be easily compared element-wisely, the distance matrix $\mathbf{A}$ implied by the taxonomy is pairwise in nature. We therefore specify the taxonomy with paired inputs $T : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ and optimizes it with respect to the distance matrix $\mathbf{A}$:

$$L_t(T, E) \triangleq \mathbb{E}[l_t(T(E(\mathbf{x}_1, u_1, \mathbf{A}), E(\mathbf{x}_2, u_2, \mathbf{A})), \mathbf{A}_{u_1,u_2})],$$

where the expectation $\mathbb{E}$ is taken over a pair of i.i.d. samples $(\mathbf{x}_1, u_1), (\mathbf{x}_2, u_2)$ from the joint data distribution $p(\mathbf{x}, u)$. Here l_t denotes a regression loss (e.g., ℓ_2 distance).

Predictor. The predictor $F : \mathcal{Z} \to \mathcal{Y}$ simply takes encoding $\mathbf{e}_l$ as input and outputs a label. We therefore optimize it with respect to the labels:

$$L_f(F, E) \triangleq \mathbb{E}^s[l_f(F(E(\mathbf{x}_l, u_l, \mathbf{A}), y_l)],$$

where the expectation $\mathbb{E}^s$ is taken over only the data in the source-domain data distribution $p^s(\mathbf{x}, y, u)$, since labels are not available in the target domains. Here l_f could be specified according to the prediction task at hand (e.g., cross-entropy loss for classification or ℓ_2 distance for regression).

The Full Game. With the aforementioned players, we are now ready to specify the full game.

$$\min_{E,F,T} \max_{D} L_f(E, F) - \lambda_d L_d(D, E) + \lambda_t L_t(T, E), \quad (1)$$

where E, F, and T play cooperatively against the discriminator D. The opposite optimization direction between the discriminator D and the taxonomist T is due to the inherent difference in their inducing properties: the discriminator D aims to find remaining domain information in the encoding, thus in an adversarial position with respect to the encoder; in contrast, the encoder and the taxonomist have the same goal of keeping taxonomy information within the encoding. The hyper-parameters $\lambda_d, \lambda_t \geq 0$ are introduced to enable a flexible trade-off, such that the encoder E and the predictor F can be learned with proper regularization.

Discussion. Traditional adversarial domain adaptation enforces the encoder E to fool the discriminator D, so that it aligns all domains uniformly. In our model, due to the addition of taxonomist T, the encoder E has to retain a certain amount of domain information in order to recover the taxonomy $\mathbf{A}$, such that the alignment is no longer uniform. As a result, the discriminator must compete with the taxonomist during the optimization process to reach an optimal balance, thus adapting successfully across domains in the taxonomy. Detailed analysis of the competition would be done in Sec. 4.

4. Theory

In this section, we will prove the intuition mentioned in Sec. 3. With the addition of the taxonomist T, the discriminator cannot enforce perfect alignment on encoding space. An interesting corollary is that TSDA can recover DANN with a non-informative taxonomy, highlighting the flexibility of our model. Furthermore, we will also discuss a straightforward extension of DANN with weighted pairwise discriminators. We prove that such DANN only produces uniform alignment, and therefore cannot incorporate the taxonomy information during adaptation.

4.1. Analysis of the Taxonomist

In this section, we formally show that the proposed formulation indeed allows the model to achieve a balance between the two contrasting goals: removing domain information and preserving domain structure. Here we focus on the analysis of the encoder which is the direct subject of the regularization. Following prior literature, we say that domain information is *fully removed* or the domains are *uniformly aligned* if $\mathbf{e} \perp\!\!\!\perp u$, which means the encoding distributions of every domain are aligned as defined below.

Definition 4.1 (Uniform Alignment). *A domain adaptation model achieves uniform alignment if its encoder* $\mathbf{e} = E(\mathbf{x})$ *satisfies* $p(\mathbf{e}|u) = p(\mathbf{e}), \forall u.$

On the other hand, the taxonomy information is *fully retained* if $L_t(T, E) = 0$, i.e., the domain taxonomy can be perfectly reconstructed from the encodings.

We begin our analysis by showing that the two goals are contradictory except for the scenario where the taxonomy does not contain any extra information beyond the domain index u; We say that such a domain taxonomy is *non-informative*, with the formal definition below.

Definition 4.2 (Non-Informative Domain Taxonomy). *A domain taxonomy is* non-informative *if and only if* $\mathbf{A}_{ij} = a, \forall i \neq j$, *for some constant* $a \in \mathbb{Z}_{>0}$, *where* $\mathbf{A}$ *is the domain taxonomy's associated domain distance matrix.*

In words, if the distance of every pair of domains is the same, the taxonomy structure cannot provide meaningful comparisons of the similarity between different domains. A domain taxonomy is non-informative if all the domains have the same parent. Now we present the main result.

Theorem 4.1 (Incompatibility). *If* $\mathbf{e} \perp\!\!\!\perp u$ *and* ℓ_2 *distance is used in* $L_t(\cdot)$,

$$L_t(T, E) = 0 \implies \mathbf{A}_{ij} = a, \forall i \neq j \in \mathcal{U}$$

for some $a \in \mathbb{Z}_{>0}$.

We make a few remarks here. First, the theorem essentially states that if the domain taxonomy is not non-informative, preserving taxonomy information and achieving uniform alignment is *incompatible*; that is, the taxonomy information cannot be fully preserved if uniform alignment is achieved (i.e., the domain information is fully removed). Second, the other direction

$$\mathbf{A}_{ij} = a, \forall i \neq j \in \mathcal{U} \implies \min_T L_t(T, E) = 0 \qquad (2)$$

holds trivially without requiring $\mathbf{e} \perp\!\!\!\perp u$ so long as T is not less powerful than a constant function. Note that the theorem is stronger than Eq. (2) since no optimization is involved. Albeit the intuitive nature of the statement, the proof is not trivial (see the Appendix for details).

We also show that TSDA can recover DANN as a special case in Corollary 4.1 (proof in the Appendix); therefore TSDA is methodologically more general than DANN.

Corollary 4.1 (TSDA Generalizes DANN). *Omitting the predictor, if the taxonomy is non-informative, then the optimum of TSDA is achieved if and only if the embedding distributions of all the domains are the same, i.e.* $p(\mathbf{e}|u = 1) = \cdots = p(\mathbf{e}|u = N) = p(\mathbf{e}), \forall \mathbf{e}$.

4.2. Effective Hyperparameters

As discussed in Sec. 3.2, λ_d and λ_t of Eq. (1) balance 1) the encoder-discriminator sub-game, which tries to achieve uniform alignment by removing domain-specific information in the representation $\mathbf{e}$, and 2) the encoder-taxonomist sub-game, which tries to preserve the part of domain-specific information captured by the domain taxonomy. We show in Theorem 4.2 that their ratio λ_t/λ_d needs to be large enough to make the balancing effect to happen.

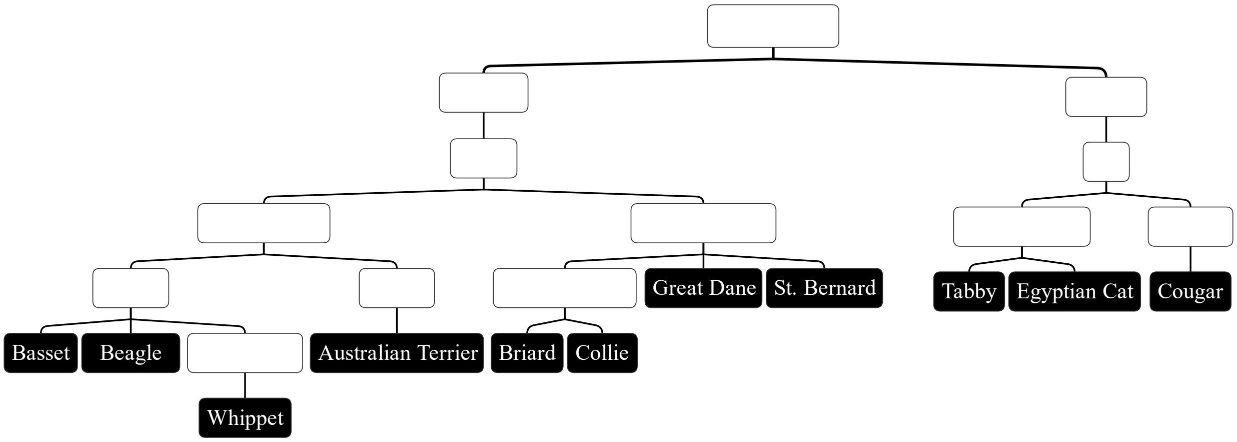

Figure 2. Domain taxonomy of *ImageNet-Attribute-DT* with 11 domains shown as leaf nodes. Note that leaf nodes are marked with black base and white text. Non-leaf nodes and their connection to leaf nodes are obtained from WordNet (Miller, 1995).

Theorem 4.2 (**Uniform Alignment, λ_t, and λ_d**)**.** *If $\lambda_t > \lambda_d$ and the domain taxonomy is not non-informative,*

$\min_{E,T} \max_D -\lambda_d L_d(D,E) + \lambda_t L_t(T,E)$ will not yield uniform alignment (Definition 4.1).

Theorem 4.2 implies that we need $\lambda_t > \lambda_d$ to prevent TSDA from converging to a trivial solution where the taxonomist is ignored and uniform alignment is achieved. In such a trivial solution, no information on the domain taxonomy is preserved, and TSDA degenerates to DANN-like methods. Guided by Theorem 4.2, λ_t and λ_d in our experiments are chosen such that $\lambda_t > \lambda_d$ (more details in Sec. 5).

4.3. An Alternative Method and Its Analysis

Since TSDA involves an extra taxonomist, one might wonder whether the same effect can be achieved by simply adjusting existing methods without algorithmic innovation. Here we show that this can be done easily by showing an impossibility result for a natural extension of DANN for domain taxonomy. Note that DANN is equivalent to TSDA without the taxonomist (i.e., $\lambda_t = 0$).

To model the pairwise distance induced by a domain taxonomy, a straightforward way is to utilize a weighting function w_{ij}, which captures the distance for each pair of domains. For example, the weight could be inversely proportional to the domain distance $w_{ij} \propto 1/\mathbf{A}_{ij}$. We can then adapt DANN by utilizing a distinct discriminator D_{ij} for each pair of domains (i,j):

$$\min_{E} \max_{D_{ij}} \mathbb{E}[\sum_{j\neq i} w_{ij} \log D_{ij}(E(\mathbf{x}))], \qquad (3)$$

where the expectation is over the data distribution $p(\mathbf{x}, u)$. For clarify, here we omit the impact of the predictor to simplify the analysis. Note that for every encoder E, each inner-maximization problem for D_{ij} reduces to a standard adversarial domain adaptation problem with two domains. We therefore can easily invoke an existing result (Ganin et al., 2016; Xu et al., 2022) to find the optimal solution.

Lemma 4.1 (**Optimal Discriminator**)**.** *For every E, the optimal D_{ij} of Eq. (3) satisfies*

$$D_{ij}(\mathbf{e}) = \frac{p(\mathbf{e}|u=i)}{p(\mathbf{e}|u=i) + p(\mathbf{e}|u=j)}, \forall \mathbf{e} \in \mathcal{Z}.$$

That says, for each pair (i,j) of domains, the optimal discriminator simply uses the appearing ratio of $\mathbf{x}$ under the two domains as the output. We can then use this result to approach the equilibrium of the minimax game (Eq. (3)).

Theorem 4.3 (**Optimal Encoder**)**.** *The min-max game in Eq. (3) has a tight lower bound:*

$$\max_{D_{ij}} \mathbb{E}[\sum_{j\neq i} w_{ij} \log D_{ij}(E(\mathbf{x}))] \geq -\frac{\log 2}{N} \sum_{i \neq j} w_{ij},$$

where N denotes the number of domains. Furthermore, the equality, i.e., the optimum, is achieved when

$$p(\mathbf{e}|u=i) = p(\mathbf{e}|u=j), \textit{ for any } i, j,$$

or equivalently, $p(\mathbf{e}|u=i) = p(\mathbf{e})$.

Theorem 4.3 states that regardless of the value for w_{ij}, encoder will always produce uniform alignment. This indicates such a modified DANN failed to incorporate taxonomy information into the domain adaptation process. The proof is available in the Appendix.

5. Experiments

In this section, we evaluate TSDA and existing methods on both synthetic and real-world datasets.

5.1. Datasets

DT-14 is a synthetic binary classification dataset with 14 domains, each consisting of 100 positive and negative la-

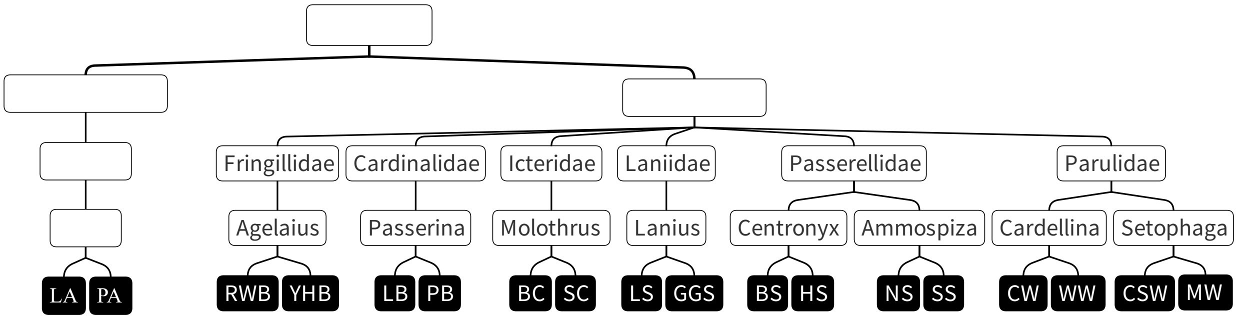

Figure 3. Domain taxonomy of *CUB-DT* with 18 domains shown as leaf nodes. Note that leaf nodes are marked with black base and white text. For clarity we abbreviate domain names in the figure: Least Auklet (**LA**), Parakeet Auklet (**PA**), Red Winged Blackbird (**RWB**), Yellow Headed Blackbird (**YHB**), Lazuli Bunting (**LB**), Painted Bunting (**PB**), Bronzed Cowbird (**BC**), Shiny Cowbird (**SC**), Loggerhead Shrike (**LS**), Great Grey Shirke (**GGS**), Baird Sparrow (**BS**), Henslow's Sparrow (**HS**), Nelson's Sparrow (**NS**), Seaside Sparrow (**SS**), Canada Warbler (**CW**), Wilson's Warbler (**WW**), Chestnut-sided Warbler (**CSW**), Myrtle Warbler (**MW**).

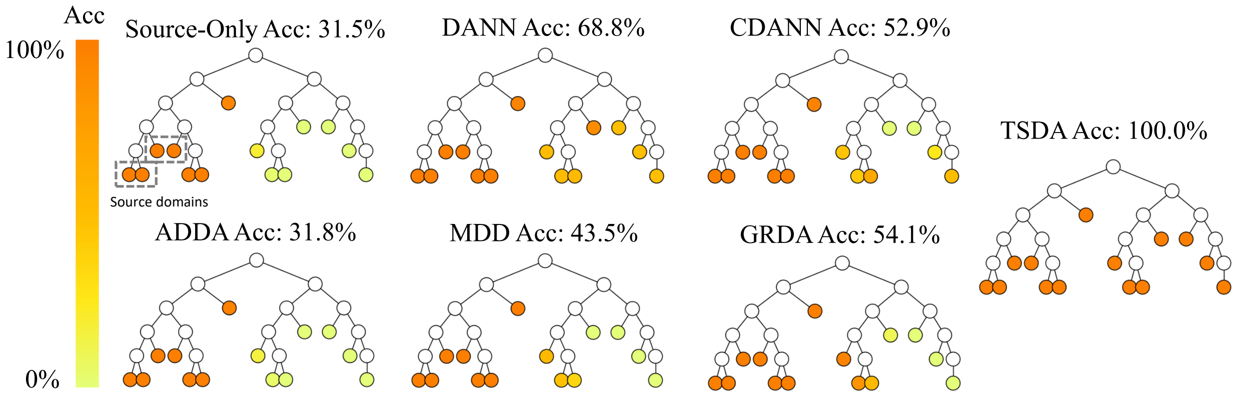

Figure 4. Detailed results on *DT-14* with 14 domains. We use the 4 domains in the dashed box as source domains. The spectrum from 'orange' to 'yellow' indicates accuracy from 100% to 0% (best viewed in color).

beled data points. To simulate real-world scenarios, we argue that a synthetic dataset should have an informative domain taxonomy (as defined in Sec. 4.1) which reflects the similarity among domains. We first randomly generate $2^5 = 32$ unit vectors $[a_i, b_i]$, $(a_i \in \mathbb{R}, b_i \in \mathbb{R}^+)$ and denote their angles as $\theta_i = \arcsin(\frac{b_i}{a_i})$. We sort all the unit vectors by their angles, and pair consecutive unit vectors according to this order (e.g., $[a_0, b_0]$ with $[a_1, b_1]$, $[a_2, b_2]$ with $[a_3, b_3]$). We then assign each pair a "parent" unit vector $[a'_i, b'_i] = \frac{1}{2}[a_{2i} + a_{2i+1}, b_{2i} + b_{2i+1}]$. This leads to a new group of unit vectors, and we can repeat the previous steps until we reach the "root" node. This produces a 6-level unit vector tree (a perfect binary tree) with 32 leaf nodes. We then randomly prune the tree to generate the final domain taxonomy with 14 leaf nodes (Fig. 4), each associated with a unit vector. We randomly generate positive data $(\mathbf{x}, i, 1)$ and negative data $(\mathbf{x}, i, 0)$ from two different 2-dimensional Gaussian distributions, $\mathcal{N}(\boldsymbol{\mu}_{i,1}, \mathbf{I})$ and $\mathcal{N}(\boldsymbol{\mu}_{i,0}, \mathbf{I})$ where $\boldsymbol{\mu}_{i,1} = [\frac{\omega_i}{\pi}a_i, \frac{\omega_i}{\pi}b_i]$ and $\boldsymbol{\mu}_{i,0} = [-\frac{\omega_i}{\pi}a_i, -\frac{\omega_i}{\pi}b_i]$. See Fig. 4 for the generated domain taxonomy.

DT-40 is constructed with the same procedure as *DT-14* except that it is pruned from a 7-level perfect binary tree, with 40 domains as leaf nodes after pruning and $b_i \in \mathbb{R}$. We select 6 domains as source domains, with others as target domains.

ImageNet-Attribute-DT (Ouyang et al., 2015) builds on the animal images from ImageNet, with additional attribute labels (e.g., whether the skin color is black or not). Here we focus on a binary classification task for the attribute "brown", because this attribute is available for the largest number of image categories, i.e., 11 categories. We use these 11 image categories as 11 domains, with "Great Dane", "St. Bernard", "Tabby", "Egyptian Cat" and "Cougar" as source domains, and the others as target domains. Each domain contains 25 images, and the domain taxonomy, shown in Fig. 2, is constructed by the hierarchies of image categories from WordNet (Miller, 1995).

CUB-DT (He & Peng, 2019) contains 11,788 images of 200 bird categories. Every image is annotated with 312 binary attributes (e.g., birds' body parts, shapes, colors). To ensure label balance, we choose to focus on the classification task of predicting "whether the upper part of a bird is black". We construct the domain taxonomy with 18 domains based on the database of the National Center for Biotechnology Information (NCBI) (Wheeler et al., 2007), with "LA", "PA",

Table 1. Accuracy for each of the 6 target domains on the *ImageNet-Attribute-DT* dataset (domain taxonomy in Fig. 2) as well as the average accuracy for different methods. Note that there is only one single DA model per column. We mark the best result with **bold face**.

Target Domain	Source-Only	DANN	CDANN	ADDA	MDD	GRDA	TSDA
Basset	84.0	84.0	72.0	88.0	88.0	84.0	**92.0**
Beagle	68.0	64.0	68.0	44.0	68.0	**76.0**	**76.0**
Whippet	68.0	64.0	68.0	68.0	**76.0**	72.0	**76.0**
Australian Terrier	80.0	80.0	72.0	84.0	**84.0**	**84.0**	**84.0**
Briad	**80.0**	**80.0**	**80.0**	**80.0**	72.0	68.0	72.0
Collie	84.0	80.0	**88.0**	84.0	84.0	84.0	84.0
Average	77.3	75.3	74.7	74.7	78.7	78.0	**80.7**

Table 2. Accuracy for each of the 9 target domains on the *CUB-DT* dataset (domain taxonomy in Fig. 3) as well as the average accuracy for different methods. Note that there is only one single DA model per column. We mark the best result with **bold face**.

Target Domain	Source-Only	DANN	CDANN	ADDA	MDD	GRDA	TSDA
Great Grey Shrike	**95.0**	78.3	73.3	58.3	80.0	23.3	**95.0**
Baird Sparrow	**71.7**	28.3	35.0	50.0	40.0	63.3	53.3
Henslow's Sparrow	65.0	48.3	50.0	63.3	58.3	**75.0**	61.7
Nelson's Sparrow	80.0	73.3	70.0	86.7	86.7	80.0	**100.0**
Seaside Sparrow	70.0	93.3	75.0	**96.7**	95.0	93.3	95.0
Canada Warbler	76.7	70.0	70.0	75.0	80.0	60.0	**88.3**
Wilson's Warbler	76.7	73.3	63.3	66.7	78.3	41.7	**85.0**
Chestnut-sided Warbler	81.7	71.7	70.0	86.7	86.7	76.7	**93.3**
Myrtle Warbler	66.7	65.0	56.7	66.7	66.7	68.3	**70.0**
Average	75.9	66.9	62.6	72.2	74.6	64.6	**82.4**

Table 3. Accuracy (%) on *DT-14* and *DT-40*.

Method	SO	DANN	CDANN	ADDA	MDD	GRDA	TSDA
DT-14	31.5	68.8	52.9	31.8	43.5	54.1	**100.0**
DT-40	43.1	55.4	43.4	43.1	42.9	44.0	**82.6**

"RWB", "YHB", "LB", "PB", "BC", "SC", "LS" as sources domains and the others as target domains. These 18 domains contain $1{,}035$ images in total. Fig. 3 shows the constructed domain taxonomy.

5.2. Baselines and Implementation

We compare TSDA with various state-of-the-art adversarial domain adaptation models, including Domain Adversarial Neural Networks (**DANN**)(Ganin et al., 2016), Adversarial Discriminative Domain Adaptation (**ADDA**)(Tzeng et al., 2017), Conditional Domain Adaptation Neural Networks (**CDANN**)(Zhao et al., 2017), Margin Disparity Discrepancy (**MDD**)(Zhang et al., 2019) and Graph-Relational Domain Adaptation (**GRDA**)(Xu et al., 2022). We also include results when one trains the model in the source domains is directly test it in the target domains (**Source-Only** or **SO** in short). Since each domain in *ImageNet-Attribute-DT* and *CUB-DT* represents a real-world category or species, we report both individual accuracy in each target domain and average accuracy over all target domains. All models above are implemented in PyTorch. The balancing hyperparameters λ_d and λ_t range from 0.1 to 1 (see the Appendix for more details on training). For a fair comparison, the encoder for all the baselines takes as input the data $\mathbf{x}$, the domain index u, and the node embedding $\mathbf{z}$ (see the Appendix for more implementation details).

5.3. Results

***DT-14* and *DT-40*.** Table 3 shows the accuracy of all the baselines and TSDA on the synthetic datasets. In *DT-14*, Source-Only "overfits" source domains and does not generalize to target domains, achieving an accuracy of 31.5%, significantly lower than random guess (50%). Both DANN and CDANN only slightly outperform random guesses, with DANN as the best baseline. Interestingly, other baselines, ADDA, MDD, and GRDA even underperform random guesses. This is possible because these baselines either ignore the domain taxonomy and blindly align different domains (DANN, CDANN, ADDA, and MDD) or fail to faithfully capture the information in the domain taxonomy during adaptation (GRDA). In contrast, TSDA successfully performs domain adaptation by aligning data from different domains according to the domain taxonomy, thereby achieving significantly higher accuracy. Similarly in *DT-40*, DANN is the best baseline and our TSDA significantly out-

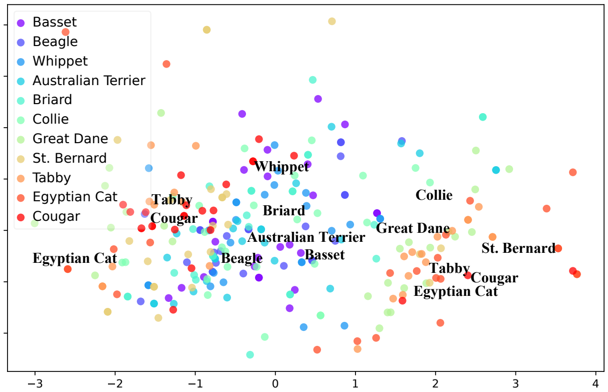

Figure 5. TSDA's learned encoding $\mathbf{e}$ for 11 domains on *ImageNet-Attribute-DT*. Domains related to "dogs", e.g., "Basset" and "Beagle", contain encodings in the middle, while domains related to "cats", e.g., "Tabby", contain encodings on both sides; this is consistent with the domain taxonomy in Fig. 2. Each domain contains 2 clusters because of the binary classification task.

performs all baselines. Note that the accuracy in *DT-40* is generally lower than that in *DT-14* since *DT-40* has more domains and a more complex domain taxonomy.

Fig. 4 shows the detailed accuracy of TSDA and all baselines in each domain of *DT-14*. The spectrum from "orange" to "yellow" on the left indicates accuracy from 100% to 0%. For all baselines, it is clear that target domains that are closer to source domains tend to have higher accuracy. This is expected because adjacent domains have similar decision boundaries, and therefore traditional adversarial DA methods are able to achieve reasonable accuracy in nearby domains by blindly enforcing uniform alignment. However, their performance is substantially worse for target domains farther away from source domains. In contrast, TSDA achieves promising results in all domains.

ImageNet-Attribute-DT. Table 1 shows the accuracy for each of the 6 target domains on *ImageNet-Attribute-DT* as well as the average accuracy for different methods. Compared to Source-Only, DANN, CDANN, and ADDA achieve a negative performance boost in terms of average accuracy, demonstrating the difficulty of performing DA across taxonomy-structured domains. Both MDD and GRDA slightly outperform Source-Only in terms of average accuracy; however, MDD's and GRDA's accuracy falls under 70% in domain "Beagle" and domain "Briad", respectively. In contrast, our TSDA manages to outperform all baselines in terms of average accuracy and achieve accuracy higher than 70% in every individual target domain.

Fig. 5 plots TSDA's learned encodings $\mathbf{e}$ on all 11 domains of *ImageNet-Attribute-DT*. Interestingly, the learned encodings' positions are consistent with the domain taxonomy in Fig. 2. For example, domains related to "dogs", e.g., "Basset" and "Beagle", contain encodings in the middle of Fig. 5, while domains related to "cats", e.g., "Egyptian Cat", contain encodings on both sides of Fig. 5. Moreover, domain "Basset" and domain "Beagle" share the same parent in the domain taxonomy, and encodings from these two domains are close in Fig. 5. Note that each domain contains 2 clusters because of the binary classification task.

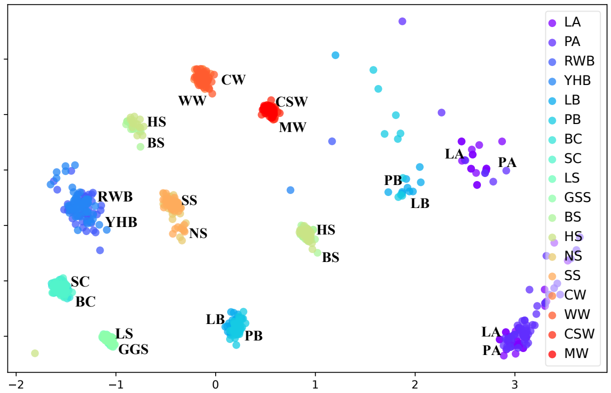

Figure 6. TSDA's learned encoding $\mathbf{e}$ for 18 domains on *CUB-DT*. Domain "Least Auklet" (LA) and domain "Parakeet Auklet" (PA) are very different from all others in the domain taxonomy (Fig. 3), correspondingly their encodings are on the far right side, with most of them in the bottom-right corner. Note that each domain has 2 clusters associated with the binary classification labels.

CUB-DT. Table 2 shows the accuracy for each of the 9 target domains on *CUB-DT* as well as the average accuracy for different methods. Compared to Source-Only, existing DA methods including DANN, CDANN, ADDA, MDD, and GRDA fail to achieve performance boost in terms of average accuracy. In terms of individual target domain accuracy, ADDA and GRDA are able to improve upon Source-Only in domain "Seaside Sparrow" and domain "Henslow's Sparrow", respectively. In contrast, our TSDA significantly improves upon all baselines in terms of average accuracy and achieves the highest accuracy in 6 out of the 9 target domains by taking full advantage of the domain taxonomy during domain adaptation.

Fig. 6 plots TSDA's learned encodings $\mathbf{e}$ on all 18 domains of *CUB-DT*. We can see that the positions of the learned encodings are consistent with the domain taxonomy in Fig. 3. For example, domain "Least Auklet" (LA) and domain "Parakeet Auklet" (PA) share the same parent in the domain taxonomy, and therefore their encodings are close in the figure. Moreover, these two domains are very different from all other domains in the domain taxonomy (Fig. 3); correspondingly their encodings are on the far right side of the figure, with most of them in the bottom-right corner.

6. Conclusion and Future Work

We propose to characterize domain similarity using a *domain taxonomy*, identify the problem of adaptation across taxonomy-structured domains, and develop taxonomy-structured domain adaptation (TSDA) as the first general DA method to address this problem. We provide theoretical analysis showing that our TSDA retains typical DA methods' capability of uniform alignment when the domain taxonomy is non-informative, and balances domain similarity and domain invariance for other domain taxonomies. As a limitation, our TSDA still assumes the availability of a taxonomy structure to describe relationship among different domains. Therefore from the methodological perspective, it would be interesting future work to explore jointly inferring the domain taxonomy and performing domain adaptation through either conditional or causal approaches (Wang et al., 2020b) while also accounting for the uncertainty of the inferred taxonomy (Mi et al., 2022). From the empirical (application) perspective, it would also be interesting future work to explore taxonomy-structured multi-domain data from other modalities including images (Mao et al., 2021), speech signals (Huang et al., 2020), time series (Yang et al., 2022), wireless signals (Zhao et al., 2020), etc.

Acknowledgement

We would like to thank Bernie Wang and the reviewers/AC for the constructive comments to improve the paper. ZX and HW are partially supported by NSF Grant IIS-2127918 and an Amazon Faculty Research Award.

References

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

Chen, Z., Zhuang, J., Liang, X., and Lin, L. Blending-target domain adaptation by adversarial meta-adaptation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2248–2257, 2019.

Dai, S., Sohn, K., Tsai, Y.-H., Carin, L., and Chandraker, M. Adaptation across extreme variations using unlabeled domain bridges. *arXiv preprint arXiv:1906.02238*, 2019.

Das, D. and Lee, C. G. Graph matching and pseudo-label guided deep unsupervised domain adaptation. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27*, pp. 342–352. Springer, 2018.

Ding, H., Ma, Y., Deoras, A., Wang, Y., and Wang, H. Zero-shot recommender systems. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022.

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

Gong, R., Danelljan, M., Dai, D., Wang, W., Paudel, D. P., Chhatkuli, A., Yu, F., and Van Gool, L. Tada: Taxonomy adaptive domain adaptation. *Withdrawn from ICLR 2022*, 2021.

Gong, R., Danelljan, M., Dai, D., Paudel, D. P., Chhatkuli, A., Yu, F., and Gool, L. V. Tacs: Taxonomy adaptive cross-domain semantic segmentation. In *ECCV*, 2022.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *NIPS*, pp. 2672–2680, 2014.

Grandvalet, Y. and Bengio, Y. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004.

He, X. and Peng, Y. Fine-grained visual-textual representation learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(2):520–531, 2019.

Huang, H., Wang, H., and Mak, B. Recurrent poisson process unit for speech recognition. In *AAAI*, volume 33, pp. 6538–6545, 2019.

Huang, H., Xue, F., Wang, H., and Wang, Y. Deep graph random process for relational-thinking-based speech recognition. In *ICML*, 2020.

Jin, X., Park, Y., Maddix, D., Wang, H., and Wang, Y. Domain adaptation for time series forecasting via attention sharing. In *International Conference on Machine Learning*, pp. 10280–10297. PMLR, 2022.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Kumar, A., Ma, T., and Liang, P. Understanding self-training for gradual domain adaptation. In *International Conference on Machine Learning*, pp. 5468–5479. PMLR, 2020.

Kuroki, S., Charoenphakdee, N., Bao, H., Honda, J., Sato, I., and Sugiyama, M. Unsupervised domain adaptation based on source-guided discrepancy. In *AAAI*, pp. 4122–4129, 2019.

Long, M., Cao, Z., Wang, J., and Jordan, M. I. Conditional adversarial domain adaptation. In *NIPS*, pp. 1647–1657, 2018.

Mancini, M., Bulo, S. R., Caputo, B., and Ricci, E. Adagraph: Unifying predictive and continuous domain adaptation through graphs. In *CVPR*, pp. 6568–6577, 2019.

Mao, C., Chiquier, M., Wang, H., Yang, J., and Vondrick, C. Adversarial attacks are reversible with natural supervision. In *ICCV*, 2021.

Maria Carlucci, F., Porzi, L., Caputo, B., Ricci, E., and Rota Bulo, S. Autodial: Automatic domain alignment layers. In *Proceedings of the IEEE international conference on computer vision*, pp. 5067–5075, 2017.

Mi, L., Wang, H., Tian, Y., and Shavit, N. Training-free uncertainty estimation for neural networks. In *AAAI*, 2022.

Miller, G. A. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

Nguyen-Meidine, L. T., Belal, A., Kiran, M., Dolz, J., Blais-Morin, L.-A., and Granger, E. Unsupervised multi-target domain adaptation through knowledge distillation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1339–1347, 2021.

Ouyang, W., Li, H., Zeng, X., and Wang, X. Learning deep representation with large-scale attributes. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1895–1903, 2015.

Pan, S. J. and Yang, Q. A survey on transfer learning. *TKDE*, 22(10):1345–1359, 2009.

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

Pan, S. J., Tsang, I. W., Kwok, J. T., and Yang, Q. Domain adaptation via transfer component analysis. *TNN*, 22(2): 199–210, 2010.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1406–1415, 2019.

Peng, X., Li, Y., and Saenko, K. Domain2vec: Domain embedding for unsupervised domain adaptation. *arXiv preprint arXiv:2007.09257*, 2020.

Pilancı, M. and Vural, E. Domain adaptation on graphs by learning aligned graph bases. *IEEE Transactions on Knowledge and Data Engineering*, 34(2):587–600, 2020.

Prabhu, V., Khare, S., Kartik, D., and Hoffman, J. Sentry: Selective entropy optimization via committee consistency for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8558–8567, 2021.

Rangapuram, S. S., Kapoor, S., Nirwan, R. S., Mercado, P., Januschowski, T., Wang, Y., and Bohlke-Schneider, M. Coherent probabilistic forecasting of temporal hierarchies. In *International Conference on Artificial Intelligence and Statistics*, pp. 9362–9376. PMLR, 2023.

Redko, I., Morvant, E., Habrard, A., Sebban, M., and Bennani, Y. A survey on domain adaptation theory: learning bounds and theoretical guarantees. *arXiv preprint arXiv:2004.11829*, 2020.

Saito, K., Watanabe, K., Ushiku, Y., and Harada, T. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, pp. 3723–3732, 2018.

Sankaranarayanan, S., Balaji, Y., Castillo, C. D., and Chellappa, R. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR*, pp. 8503–8512, 2018.

Sun, B. and Saenko, K. Deep CORAL: correlation alignment for deep domain adaptation. In *ICCV workshop on Transferring and Adapting Source Knowledge in Computer Vision (TASK-CV)*, pp. 443–450, 2016.

Tasar, O., Tarabalka, Y., Giros, A., Alliez, P., and Clerc, S. Standardgan: Multi-source domain adaptation for semantic segmentation of very high resolution satellite images by data standardization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 192–193, 2020.

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

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. Adversarial discriminative domain adaptation. In *CVPR*, pp. 7167–7176, 2017.

Wang, H. *Bayesian Deep Learning for Integrated Intelligence: Bridging the Gap between Perception and Inference*. PhD thesis, Hong Kong University of Science and Technology, 2017.

Wang, H. and Yeung, D.-Y. Towards bayesian deep learning: A framework and some existing methods. *TDKE*, 28(12): 3395–3408, 2016.

Wang, H. and Yeung, D.-Y. A survey on bayesian deep learning. *CSUR*, 53(5):1–37, 2020.

Wang, H., Wang, N., and Yeung, D. Collaborative deep learning for recommender systems. In *KDD*, pp. 1235–1244, 2015.

Wang, H., Mao, C., He, H., Zhao, M., Jaakkola, T. S., and Katabi, D. Bidirectional inference networks: A class of deep bayesian networks for health profiling. In *AAAI*, volume 33, pp. 766–773, 2019.

Wang, H., He, H., and Katabi, D. Continuously indexed domain adaptation. In *ICML*, 2020a.

Wang, Y., Menkovski, V., Wang, H., Du, X., and Pechenizkiy, M. Causal discovery from incomplete data: A deep learning approach. 2020b.

Wheeler, D. L., Barrett, T., Benson, D. A., Bryant, S. H., Canese, K., Chetvernin, V., Church, D. M., DiCuccio, M., Edgar, R., Federhen, S., et al. Database resources of the national center for biotechnology information. *Nucleic acids research*, 36(suppl_1):D13–D21, 2007.

Wu, Y., Winston, E., Kaushik, D., and Lipton, Z. Domain adaptation with asymmetrically-relaxed distribution alignment. In *International conference on machine learning*, pp. 6872–6881. PMLR, 2019.

Xu, Z., He, H., Lee, G.-H., Wang, Y., and Wang, H. Graph-relational domain adaptation. In *ICLR*, 2022.

Xu, Z., Hao, G.-Y., He, H., and Wang, H. Domain-indexing variational bayes: Interpretable domain index for domain adaptation. In *ICLR*, 2023.

Yang, B. and Yuen, P. C. Cross-domain visual representations via unsupervised graph alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5613–5620, 2019.

Yang, Y., Yuan, Y., Zhang, G., Wang, H., Chen, Y.-C., Liu, Y., Tarolli, C., Crepeau, D., Bukartyk, J., Junna, M., Videnovic, A., Ellis, T., Lipford, M., Dorsey, R., and Katabi, D. Artificial intelligence-enabled detection and assessment of parkinson's disease using nocturnal breathing signals. *Nature medicine*, 1(1):1–1, 2022.

Zhang, Y., Liu, T., Long, M., and Jordan, M. I. Bridging theory and algorithm for domain adaptation. *arXiv preprint arXiv:1904.05801*, 2019.

Zhao, H., Zhang, S., Wu, G., Moura, J. M. F., Costeira, J. P., and Gordon, G. J. Adversarial multiple source domain adaptation. In *NIPS*, pp. 8568–8579, 2018.

Zhao, H., des Combes, R. T., Zhang, K., and Gordon, G. J. On learning invariant representations for domain adaptation. In *ICML*, pp. 7523–7532, 2019.

Zhao, M., Yue, S., Katabi, D., Jaakkola, T. S., and Bianchi, M. T. Learning sleep stages from radio signals: A conditional adversarial architecture. In *ICML*, pp. 4100–4109, 2017.

Zhao, M., Hoti, K., Wang, H., Raghu, A., and Katabi, D. Assessment of medication self-administration using artificial intelligence. *Nature medicine*, 2020.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

Zou, Y., Yu, Z., Kumar, B. V., and Wang, J. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

A. Proof

Theorem 4.1 (Incompatibility). *If* $\mathbf{e} \perp\!\!\!\perp u$ *and* ℓ_2 *distance is used in* $L_t(\cdot)$*,*

$$L_t(T,E) = 0 \implies \mathbf{A}_{ij} = a, \forall i \neq j \in \mathcal{U}$$

for some $a \in \mathbb{Z}_{>0}$*.*

Proof. We first expand the loss as

$$\begin{aligned} L_t(T,E) =& \frac{1}{2} \iint \sum_{i \neq j} (\mathbf{A}_{ij} - T(\mathbf{e},\mathbf{e}'))^2 p(\mathbf{e},i) p(\mathbf{e}',j) \; d\mathbf{e} \, d\mathbf{e}' \\ =& \frac{1}{2N^2} \iint \sum_{i \neq j} (\mathbf{A}_{ij} - T(\mathbf{e},\mathbf{e}'))^2 p(\mathbf{e}) p(\mathbf{e}') \; d\mathbf{e} \, d\mathbf{e}' \\ =& \frac{1}{2N^2} \sum_{i \neq j} \mathbf{A}_{ij}^2 - 2\mathbf{A}_{ij} \mathbb{E}[T(\mathbf{e},\mathbf{e}')] + \mathbb{E}[T(\mathbf{e},\mathbf{e}')^2], \end{aligned}$$

where the second equality is due to $p(\mathbf{e},i) = p(\mathbf{e})p(i) = p(\mathbf{e})/N$. We also have the equality

$$\mathbb{E}[T(\mathbf{e},\mathbf{e}')^2] = \mathbb{E}[T(\mathbf{e},\mathbf{e}')]^2 + \text{Var}[T(\mathbf{e}_i,\mathbf{e}_j)].$$

Combining the two equalities, we get

$$L_t(T,E) = \alpha \text{Var}[T(\mathbf{e},\mathbf{e}')] + \beta \sum_{i \neq j} [\mathbf{A}_{ij} - \mathbb{E}[T(\mathbf{e},\mathbf{e}')]]^2 \geq 0,$$

where $\alpha = (N)(N-1)/(2N^2)$ and $\beta = 1/(2N^2)$. Finally,

$$L_t(T,E) = 0 \implies \mathbf{A}_{ij} = \mathbb{E}[T(\mathbf{e},\mathbf{e}')], \forall i \neq j,$$

completing the proof. □

Corollary 4.1 (TSDA Generalizes DANN). *Omitting the predictor, if the taxonomy is non-informative, then the optimum of TSDA is achieved if and only if the embedding distributions of all the domains are the same, i.e.* $p(\mathbf{e}|u=1) = \cdots = p(\mathbf{e}|u=N) = p(\mathbf{e}), \forall \mathbf{e}$*.*

Proof. Based on Theorem 4.1, we know that given the taxonomy is non-informative, we always have $T(\mathbf{e},\mathbf{e}') = a = A_{ij}$, such that $L_t(T,E) = 0$ for any $\mathbf{e},\mathbf{e}'$, (at this time, $\text{Var}[T(\mathbf{e},\mathbf{e}')] = 0$, $\sum_{i \neq j} [\mathbf{A}_{ij} - \mathbb{E}[T(\mathbf{e},\mathbf{e}')]]^2 = 0$). Thus, to ensure that TSDA is optimal, we only need to ensure that the discriminator achieves its optimum, because $\forall \mathbf{e}$, the taxonomist will always be optimal. The discriminator achieves its optimum if and only $p(\mathbf{e}|u=1) = \cdots = p(\mathbf{e}|u=N) = p(\mathbf{e}), \forall \mathbf{e}$, completing our proof.

□

Theorem 4.2 (Uniform Alignment, λ_t, and λ_d). *If* $\lambda_t > \lambda_d$ *and the domain taxonomy is not non-informative,* $\min_{E,T} \max_D -\lambda_d L_d(D,E) + \lambda_t L_t(T,E)$ *will not yield uniform alignment .*

Proof. Define $A_e \sim p(A_e) = \frac{1}{N^2-N} \sum_{i \neq j} p(A_e|i,j)$ where $p(A_e|i,j) = \delta(A_e - \mathbf{A}_{i,j})$. We model $q(A_e|\mathbf{e},\mathbf{e}') = \mathcal{N}(T(\mathbf{e},\mathbf{e}'),1)$, which is trainable. Let $L_t(T,E) = -\mathbb{E}_{p(A_e,\mathbf{e},\mathbf{e}')}[\log q(A_e|\mathbf{e},\mathbf{e}')]$, which is another form of $L_t(T,E)$ in the main paper.

$$\begin{aligned} & \min_E \min_T \max_D -\lambda_d L_d(D,E) + \lambda_t [L_t(T,E)] \\ =& \min_E \lambda_d [I(u;\mathbf{e}) - H[u]) - \lambda_t [I(A_e;\mathbf{e},\mathbf{e}') - H(A_e)] \\ =& \lambda_d \{[H(A_e) - H(u)] + \min_E [I(u;\mathbf{e}) - \lambda I(A_e;\mathbf{e},\mathbf{e}')]\} \end{aligned}$$

where $\lambda = \frac{\lambda_t}{\lambda_d}$.

The domain taxonomy is not non-informative, and thus $H(A_e) > 0$. Define $I^{\#}$ as $I^{\#}(A_e; \mathbf{e}, \mathbf{e}') = H(A_e) = H(\mathbf{e}, \mathbf{e}')$ holds. When $I^{\#}(A_e; \mathbf{e}, \mathbf{e}') = H(A_e) = H(\mathbf{e}, \mathbf{e}')$, we have

$$\begin{aligned}
&\min_E I(u; \mathbf{e}) - \lambda I(A_e; \mathbf{e}, \mathbf{e}') \\
&\quad \leq I^{\#}(u; \mathbf{e}) - \lambda I^{\#}(A_e; \mathbf{e}, \mathbf{e}') \\
&\quad \leq I^{\#}(u; \mathbf{e}, \mathbf{e}') - \lambda H(A_e) \\
&\quad \leq H(\mathbf{e}, \mathbf{e}') - \lambda H(A_e) \\
&\quad = H(A_e) - \lambda H(A_e)
\end{aligned}$$

When $H(A_e) - \lambda H(A_e) < 0$, i.e., $\lambda = \frac{\lambda_t}{\lambda_d} > 1$, $\min_E I(u; \mathbf{e}) - \lambda I(A_e; \mathbf{e}, \mathbf{e}') < 0$. Then, we have $I(u; \mathbf{e}) > 0$ since, if $I(u; \mathbf{e}) = 0$, $I(u; \mathbf{e}) - \lambda I(A_e; \mathbf{e}, \mathbf{e}') = 0$ must hold. $I(u; \mathbf{e}) > 0$ indicates $\min_{E,T} \max_D -\lambda_d L_d(D, E) + \lambda_t L_t(T, E)$ will not lead to uniform alignment.

□

Next, we will prove the lemma and the theorem in Sec 4.3. For reference, we restate the minimax game of the alternative method as follows:

$$\min_E \max_{D_{ij}} \mathbb{E}[\sum_{j \neq i} w_{ij} \log D_{ij}(E(\mathbf{x}))]. \tag{4}$$

Lemma 4.1 (Optimal Discriminator). *For every E, the optimal D_{ij} of Eq. (3) satisfies*

$$D_{ij}(\mathbf{e}) = \frac{p(\mathbf{e}|u = i)}{p(\mathbf{e}|u = i) + p(\mathbf{e}|u = j)}, \forall \mathbf{e} \in \mathcal{Z}.$$

Proof. For the fixed E, Eq. (3) could be written as:

$$\begin{aligned}
\mathbb{E}[\sum_{j \neq i} w_{ij} \log D_{ij}(E(\mathbf{x}))] &= \mathbb{E}_{\mathbf{x}, i \sim p(\mathbf{x}, u)}[\sum_{j \neq i} w_{ij} \log D_{ij}(E(\mathbf{x}))] && (5)\\
&= \mathbb{E}_{\mathbf{e}, i \sim p(\mathbf{e}, u)}[\sum_{j \neq i} w_{ij} \log D_{ij}(\mathbf{e})] && (6)\\
&= \frac{1}{2} \int \sum_{i \neq j} w_{ij} (p(\mathbf{e}, i) \log D_{ij}(\mathbf{e}) + p(\mathbf{e}, j) \log(1 - D_{ij}(\mathbf{e}))) \, d\mathbf{e} && (7)\\
&= \frac{1}{2} \int \sum_{i \neq j} w_{ij} (p(\mathbf{e}|u = i) p(u = i) \log D_{ij}(\mathbf{e}) + p(\mathbf{e}|u = j) p(u = j) \log(1 - D_{ij}(\mathbf{e}))) \, d\mathbf{e} && (8)\\
&= \frac{1}{2} \sum_{i \neq j} w_{ij} \int (p(\mathbf{e}|u = i) p(u = i) \log D_{ij}(\mathbf{e}) + p(\mathbf{e}|j) p(u = j) \log(1 - D_{ij}(\mathbf{e}))) \, d\mathbf{e} && (9)\\
&= \frac{1}{2N} \sum_{i \neq j} w_{ij} \int (p(\mathbf{e}|u = i) \log D_{ij}(\mathbf{e}) + p(\mathbf{e}|u = j) \log(1 - D_{ij}(\mathbf{e}))) \, d\mathbf{e}, && (10)
\end{aligned}$$

where Eq. (10) holds because we assume each domain has the same amount of data, and thus for any domain identity i, we have $p(u = i) = \frac{1}{N}$.

For function $f(\alpha) = a \log \alpha + b \log(1 - \alpha)$ with $a, b \in \mathbb{R}^+$, we have $\operatorname{argmax}_\alpha f(\alpha) = \frac{a}{a+b}$. Therefore, to maximize the value function (Eq. (3)), we have the optimal $D_{ij}(e)$ as $\frac{p(\mathbf{e}|u=i)}{p(\mathbf{e}|u=i)+p(\mathbf{e}|u=j)}$ for any domain pair (i, j).

□

Theorem 4.3 (Optimal Encoder). *The min-max game in Eq. (3) has a tight lower bound:*

$$\max_{D_{ij}} \mathbb{E}[\sum_{j \neq i} w_{ij} \log D_{ij}(E(\mathbf{x}))] \geq \frac{\log 2}{N} \sum_{i \neq j} w_{ij},$$

where N denotes the number of domains. Furthermore, the equality, i.e., the optimum, is achieved when

$$p(\mathbf{e}|u=i) = p(\mathbf{e}|u=j), \textit{ for any } i, j,$$

or equivalently, $p(\mathbf{e}|u=i) = p(\mathbf{e})$.

Proof. Given the optimal discriminators $\{D^*_{ij}\}_{i\neq j}$ based on Lemma 4.1, the value for (3) w.r.t. the encoder E is:

$$\begin{aligned}
\max_{D_{ij}} \mathbb{E}[\sum_{j\neq i} w_{ij}\log D_{ij}(E(\mathbf{x}))] &= \sum_{i\neq j}\frac{w_{ij}}{2}\big(p(u=i)\mathbb{E}_{\mathbf{e}|u=i}[\log D^*_{ij}(\mathbf{e})] + p(u=j)\mathbb{E}_{\mathbf{e}|u=j}[\log(1-D^*_{ij}(\mathbf{e}))]\big) && (11)\\
&= \sum_{i\neq j}\frac{w_{ij}}{2N}\big(\mathbb{E}_{\mathbf{e}|u=i}[\log D^*_{ij}(\mathbf{e})] + \mathbb{E}_{\mathbf{e}|u=j}[\log(1-D^*_{ij}(\mathbf{e}))] && (12)\\
&= \sum_{i\neq j}\frac{w_{ij}}{2N}\left(\mathbb{E}_{\mathbf{e}|u=i}\log\frac{p(\mathbf{e}|u=i)}{p(\mathbf{e}|u=i)+p(\mathbf{e}|u=j)} + \mathbb{E}_{\mathbf{e}|u=j}\log\frac{p(\mathbf{e}|u=j)}{p(\mathbf{e}|u=i)+p(\mathbf{e}|u=j)}\right) && (13)\\
&= \frac{1}{2N}\sum_{i\neq j} w_{ij}[-2\log 2 + KL(p(\mathbf{e}|u=i)\|\frac{p(\mathbf{e}|u=i)+p(\mathbf{e}|u=j)}{2}) && (14)\\
&\quad + KL(p(\mathbf{e}|u=j)\|\frac{p(\mathbf{e}|u=i)+p(\mathbf{e}|u=j)}{2})] \geq \frac{-\log 2}{N}\sum_{i\neq j} w_{ij}, && (15)
\end{aligned}$$

where Eq. (12) is due to $p(u=i)=\frac{1}{N}$ for any i. The equality in Eq. (15) holds if and only if $p(\mathbf{e}|u=1) = p(\mathbf{e}|u=2) = \cdots = p(\mathbf{e}|u=N)$. (This can be easily verified using the property $\int p(\mathbf{e}|u=i)d\mathbf{e} = 1$.) Therefore the optimal encoder E^* is achieved if and only if $p(\mathbf{e}|u=1) = p(\mathbf{e}|u=2) = \cdots = p(\mathbf{e}|u=N)$, i.e., $\mathbf{e}\perp\!\!\!\perp u$. □

B. Baselines and Implementation

B.1. Model Architecture

For fair comparison, all baselines and TSDA use the same encoder and predictor. The encoder has the following components:

- A **raw data encoder** embeds the data $\mathbf{x}_l$ into intermediate embeddings $\mathbf{h}_l$.
- A **taxonomy encoder** embeds the domain distance matrix $\mathbf{A}$ and the domain index u_l to the domain embeddings $\mathbf{z}_{u_l}$ with 1 fully connected (FC) layer. We use a taxonomy embedding loss L_g to pretrain the taxonomy encoder:

  $$L_g = \mathbb{E}_{u_1,u_2\sim p(u)}[l_g(\|\mathbf{z}_{u_1}^\top\mathbf{z}_{u_2}\|_2, \mathbf{A}_{u_1,u_2})],$$

  where u_1, u_2 are two independent domain identities sampled from $p(u)$, and l_g denotes a regression loss (e.g.,ℓ_2 distance).
- A **joint encoder** then takes as input both $\mathbf{h}_l$ and $\mathbf{z}_l$ and produces the final embeddings $\mathbf{e}_l$ with 2 FC layers.

For toy datasets *DT-14* and *DT-40*, we use 3 FC layers as the raw data encoder, while for the real dataset *ImageNet-Attribute-DT* and *CUB-DT*, we use PyTorch's default pretrained Resnet-18 as the raw data encoder.

All the predictors of baselines and TSDA contain 3 FC layers, and all the discriminators have 6 FC layers. For GRDA, we treat every pair of domains that share a common grandparent node as connected, construct GRDA's domain graph, and feed it into its discriminator to recover the graph.

For the structure of the taxonomist, we first use a 6-FC-layer neural network $T^{'}$ to produce a 2-dimensional taxonomy representation $\mathbf{t}_l$ of the data embedding $\mathbf{e}_l$, and then calculate the ℓ_2 distance of a pair of taxonomy representations $\mathbf{t}_l$. This can be written as:

$$T(E(\mathbf{x}_1,u_1,\mathbf{A}), E(\mathbf{x}_2,u_2,\mathbf{A})) = \|T^{'}(E(\mathbf{x}_1,u_1,\mathbf{A})) - T^{'}(E(\mathbf{x}_2,u_2,\mathbf{A}))\|_2 = \|\mathbf{t}_1-\mathbf{t}_2\|_2$$

B.2. Other Parameters

We have λ_d, λ_t and λ_e as the weights that balance the discriminator loss, the taxonomist loss, and the predictor loss. Note that here, λ_e is not necessary theoretically, and we include it only for convenience of hyperparameter tuning. Our loss function could be written as $\lambda_e L_f(E,F) - \lambda_d L_d(D,E) + \lambda_t L_t(T,E)$, where λ_d and λ_e range from 0.1 to 1 and λ_t ranges from 0.1 to 10. During the hyperparameter tuning, we always ensure that $\lambda_t > \lambda_d$ (Theorem 4.2). We use Adam optimizer (Kingma & Ba, 2015) for all models with learning rates from 1×10^{-4} to 1×10^{-6}. The input data and the domain distance matrix $\mathbf{A}$ are normalized according to its mean and variance. All experiments are run on NVDIA GeForce RTX 2080 Ti GPUs.

Table 4 shows the experiment results of various λ_d, λ_t combination on DT-14. TSDA achieves robust performance when $\lambda_t > \lambda_d$ ($> 90\%$ accuracy), while suffers from performance loss when $\lambda_t = \lambda_d$ (84.1% accuracy). This is in line with the aforementioned Theorem 4.2's conclusion.

Table 4. Sensitivity of hyper-parameters λ_d and λ_t on DT-14. λ_e is fixed at 0.5. The accuracy of TSDA remains stable as long as $\lambda_t > \lambda_d$.

	$\lambda_t = 1$	$\lambda_t = 2$	$\lambda_t = 4$
$\lambda_d = 0.25$	100.0	100.0	100.0
$\lambda_d = 0.5$	97.9	99.9	96.5
$\lambda_d = 1$	84.1	90.6	91.7

B.3. Training Procedure

We implement the minimax game by alternately training modules of TSDA until convergence in the following two steps:

1. We fix the encoder E, the taxonomist T and the predictor F and optimize the discriminator D. With encoding generated from E, we use the disciminator loss $L_d(D,E)$ to train the discriminator.
2. We fix the discriminator D and minimize $\lambda_e L_f(E,F) - \lambda_d L_d(D,E) + \lambda_t L_t(T,E)$ to train the encoder E, the taxonomist T and, the predictor F.

We summarize the training procedure formally in Algorithm 1.

Algorithm 1 TSDA Training

```
Specify the distance matrix $\mathbf{A}$
Initialize the encoder $E$, the taxonomist $T$, the predictor $F$ and the discriminator $D$ networks.
Initialize the $D$ optimizer and the $ETF$ optimizer for $E$, $T$, $F$.
for each epoch do
    for each mini-batch of data $\mathbf{x}_l$, domain index $u_l$ with size $m$ do
        Calculate the gradient for $D$: $\nabla_D = \nabla_{\theta_D} \frac{1}{m} L_d(D,E)(\mathbf{x}_l, u_l, \mathbf{A})$.
        Update the discriminator weights with $D$ optimizer using $\nabla_D$.
        Calculate the gradient for $E$, $T$, $F$: $\nabla_{E,T,F} = \nabla_{\theta_{E,T,F}} \frac{1}{m}[\lambda_e L_f(E,F) - \lambda_d L_d(D,E) + \lambda_t L_t(T,E)](\mathbf{x}_l, u_l, \mathbf{A})$.
        Update the weights of the encoder, the taxonomist and the predictor with $ETF$ optimizer using $\nabla_{E,T,F}$.
    end for
end for
```

C. Inference Procedure

The inference procedure of TSDA is formally presented in Algorithm 2. We only need the encoder E and the predictor F during inference to perform prediction.

D. Ablation Study

We perform an ablation study to demonstrate the effectiveness of the two key components, the discriminator and the taxonomist, in TSDA. The results in Table 5 show that without either component, we can observe significant performance

Algorithm 2 TSDA Inference

```
Load the encoder $E$, the predictor $F$.
Load the testing data $X_{test}$.
for each example $x_l$ in $X_{test}$ do
    Predict the output $\hat{y}_l = F(E(x_l))$.
    Store the prediction $\hat{y}_l$.
end for
Evaluate the performance of the model using desired metric(s) on the predicted ($\hat{y}_l$) and true labels ($y_l$).
```

drops in most tasks. TSDA without the discriminator is no longer an adversarial domain adaptation framework, which shows that simply aligning similar domains together does not perform well. TSDA without the taxonomist is equivalent to the baseline DANN, and its results reveal that without necessary taxonomy information, performance suffers. The ablation study illustrates that all components contribute to the full model TSDA.

Table 5. Accuracy of TSDA compared with TSDA without discriminator or taxonomist on all four datasets as well as the average for all tasks. Note that TSDA without Taxonomist is in fact DANN. We mark the best average accuracy with **bold face**

Target	DT-14	DT-40	ImageNet-Attribute-DT	CUB-DT	Average
TSDA *w/o Discriminator*	34.5	42.7	62.0	80.6	55.0
TSDA *w/o Taxonomist*	68.8	55.4	75.3	66.9	66.6
TSDA	**100.0**	**82.6**	**80.7**	**82.4**	**86.4**

E. Larger Figures

In this section, we provide larger versions of figures for TSDA's learned encoding visualization in the main paper. Results are shown in Fig. 7 and Fig. 8.

F. Additional Discussions

Adversarial Training versus Non-Adversarial Methods for Domain Alignment. The number of new adversarial training methods in recent literature has gradually reduced. However, to the best of our knowledge, adversarial training is still the state of the art for domain alignment. While it is true that non-adversarial approaches have been proposed and shown to be effective, such as maximum mean discrepancy (MMD) and entropy minimization (Grandvalet & Bengio, 2004), adversarial training still achieves state-of-the-art performance in many cases (Wang et al., 2020a; Xu et al., 2022); for example, Xu et al. (2022) (one of our baseline) has shown that its adversarial DA method could outperform state-of-the-art non-adversarial methods such as Grandvalet & Bengio (2004). Additionally, since our TSDA's taxonomist is cooperative rather than adversarial, it can potentially be incorporated into non-adversarial methods to improve their performance in our new taxonomy-structured DA setting too, which would be interesting future work.

Not All Datasets Have Domain Taxonomies as Additional Information. While taxonomy is not readily available in every dataset, we believe that such structure naturally formalize many nested, hierarchical domains in the real world; typical examples include product taxonomies in e-commerce and disease taxonomies in healthcare. We therefore see our work as a pilot study to demonstrate the benefit of leveraging domain taxonomies during domain adaptation; with this, we hope to encourage the community to pay more attention to such taxonomy structure both during methodology development and during data collection.

On the other hand, we might also conduct automatic taxonomy induction in an unsupervised way. For example, Peng et al. (2020) and Xu et al. (2023) show that it is possible to learn domain embeddings or domain indices that capture the relations between different domains in an unsupervised manner. Therefore, we can use Peng et al. (2020) and Xu et al. (2023) to first infer the domain embeddings or indices, construct a domain taxonomy according to these domain embeddings or indices, and then apply our TSDA. This would be interesting future work.

Comparsion with Das & Lee (2018); Yang & Yuen (2019); Pilancı & Vural (2020). All these works use graph matching as a domain discrepancy metric. Das & Lee (2018) and Yang & Yuen (2019) propose to align the "source graph" and "target

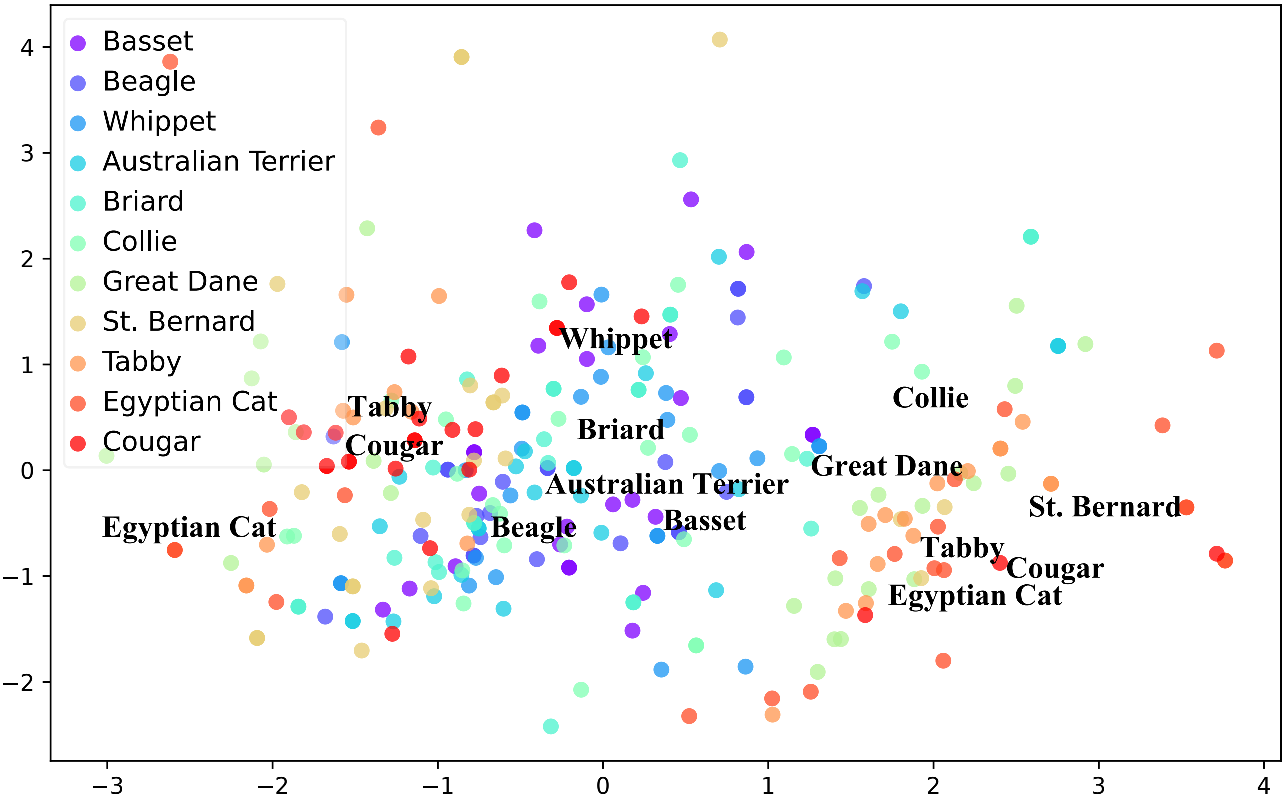

Figure 7. TSDA's learned encoding $\mathbf{e}$ for 11 domains on *ImageNet-Attribute-DT*.

graph" constructed from source and target domains, respectively, where *each data point is a node in the graph*. On the other hand, Pilancı & Vural (2020) aims to perform domain adaptation in a setting where *each data point is a graph*. In contrast, our TSDA focuses on the setting where each domain is a node in the domain taxonomy. Additionally, Das & Lee (2018); Yang & Yuen (2019); Pilancı & Vural (2020) focus on the setting with a single source domain and a single target domain, while our TSDA focuses on multiple source domains and target domains. Therefore, Das & Lee (2018); Yang & Yuen (2019); Pilancı & Vural (2020) are *not applicable* in our setting.

Extension to Source-Free Setting. Our method can be naturally extended to the source-free setting. Specifically, we could apply our approach only to target domains, and use the pretrained model to regularize the training process. We can keep the classifier C fixed and train the encoder E, taxonomist T, and discriminator D given the target domain taxonomy. Meanwhile, we encourage the model to produce similar predictions as the pretrained model.

Formally, denote the input as x, the pretrained encoder as E', and the encoder after adaptation as E. In this case, we can use $\text{DIST}(C(E'(x)), C(E(x)))$ as an additional regularization term to regularize the training process. Here "DIST" refers to the distance between two predictions, which could be cross-entropy (for classification tasks) or L2 (for regression tasks). The final objective function will then become the sum of $\text{DIST}(C(E'(x)), C(E(x)))$ and Eq. (1).

Table 6. Accuracy on DT-14 with different numbers of samples.

Sample Number	DANN	TSDA
50%	61.6	99.2
75%	65.6	99.9
100%	68.8	100.0

Sample Complexity. From an additional experiment on DT-14 (Table 6), we can see that as the number of samples in the source domain decreases (e.g., only 50% of the original samples), the baseline method (DANN) suffers from a performance drop. In contrast, TSDA remains stable and outperforms the baseline by a significant margin.

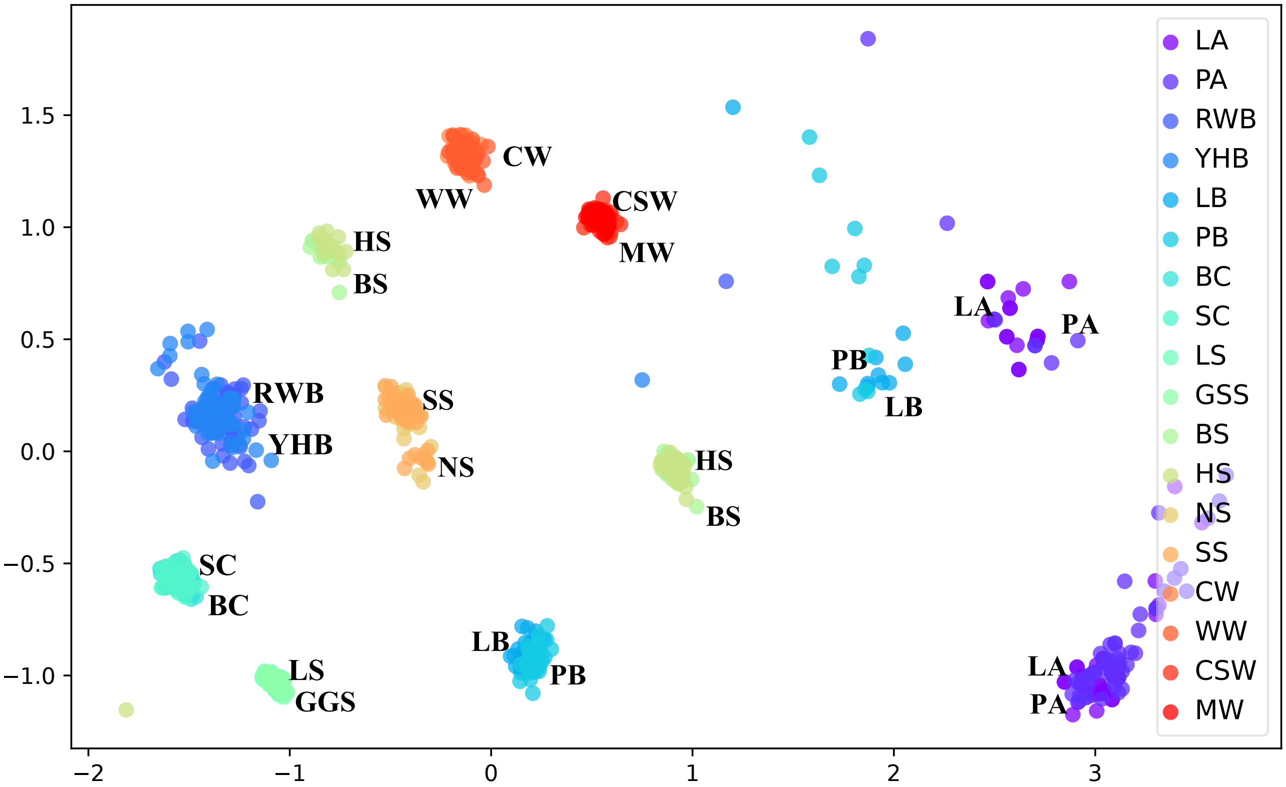

Figure 8. TSDA's learned encoding $\mathbf{e}$ for 18 domains on *CUB-DT*.

Limitation of Our Method. As mentioned in Corollary 4.1, our method will degenerate into DANN (Ganin et al., 2016) when the taxonomy is non-informative, i.e., when the distance between every pair of domains is identical (e.g., a flat taxonomy). In this case, our method essentially reduces to the standard DANN model. Another limitation is that the domain taxonomy should provide a suitable inductive bias to the learning task (the domains are similar when they are closer in the taxonomy). Taxonomies without such inductive bias yield no benefit, or even do harm to the domain adaptation performance.