
Domain Adaptation with Factorizable Joint Shift

Hao He¹ Yuzhe Yang¹ Hao Wang²

Abstract

Existing domain adaptation (DA) usually assumes the domain shift comes from either the covariates or the labels. However, in real-world applications, samples selected from different domains could have biases in both the covariates and the labels. In this paper, we propose a new assumption, Factorizable Joint Shift (FJS), to handle the co-existence of sampling bias in covariates and labels. Although allowing for the shift from both sides, FJS assumes the independence of the bias between the two factors. We provide theoretical and empirical understandings about when FJS degenerates to prior assumptions and when it is necessary. We further propose Joint Importance Aligning (JIA), a discriminative learning objective to obtain joint importance estimators for both supervised and unsupervised domain adaptation. Our method can be seamlessly incorporated with existing domain adaptation algorithms for better importance estimation and weighting on the training data. Experiments on a synthetic dataset demonstrate the advantage of our method.

1. Introduction

The core problem in domain adaptation originates from the *shift* between the training distribution (source domain) p and the test distribution (target domain) q . This equivalently indicates that the samples we obtain from one domain are biased comparing to that from the other domain. Depending on different beliefs about the cause of the sampling bias, there emerges two common domain adaptation assumptions: (1) *Covariate shift (CS)* assumes the bias is purely on the covariate while $p(y|x)$ and $q(y|x)$ equals. (2) *Label shift (LS)* assumes the bias is purely on the label while $p(x|y)$ and $q(x|y)$ equals. As observed by (Schölkopf et al., 2012), covariate shift corresponds to causal learning and label shift corresponds to anti-causal learning.

¹Massachusetts Institute of Technology, CSAIL ²Department of Computer Science, Rutgers University. Correspondence to: Hao He <haohe@mit.edu>.

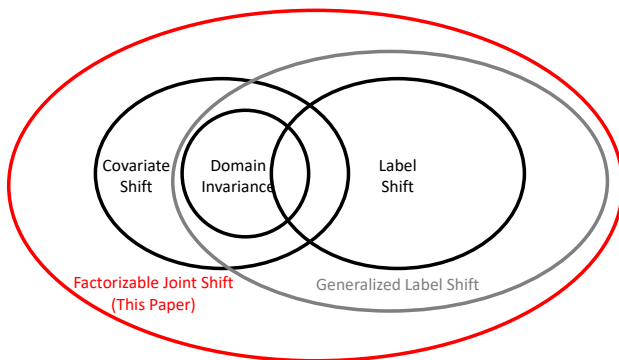


Figure 1. The Venn Diagram of domain adaptation assumptions.

Another widely adopted assumption is *domain invariance (DI)*, which assumes the discrepancy in two domains can be eliminated by feature transformations. One typical example is transferring from MNIST to colorful-MNIST, where a transformation discarding the color information will force the feature distribution invariant to the domains. Such assumption motivates many domain adaptation methods based on adversarial training (Ganin et al., 2016; Wang et al., 2020; Long et al., 2017). A recent work (Combes et al., 2020) proposed a new assumption called *generalized label shift (GLS)*, which aims to generalize domain adversarial techniques to handle mismatched label distributions.

In this paper, we focus on a more relaxed setting, called *Factorizable Joint Shift (FJS)*, where we do not assume any domain invariance and allow the bias to exist on both the data and the label. We only assume the biases coming from the data and the label are statistically independent. Through the lens of joint importance, we unify the traditional assumptions (see Figure 1) and identify that the uncertainty of the learning task affects the choice of assumptions.

We further propose a general strategy *Joint Importance Aligning (JIA)* to handle FJS. Essentially, JIA is a discriminative learning paradigm that learns the two factors of joint importance corresponding to the data and the label. We also identify, under the setting of unsupervised domain adaptation, there is a lack of information to discover the factorization of joint importance. Motivated by the smoothness of the importance, we propose to add a regularization to facilitate the clustering structure of the estimated importance. We conduct experiments on a synthetic dataset where our method outperforms all the baselines.

Table 1. Formal definition and importance factorization of traditional domain adaptation assumptions.

Assumptions	Mathematical Definitions	Joint Importance Factorizations
Covariate Shift	$\mathcal{D}_S(X) \neq \mathcal{D}_T(X), \mathcal{D}_S(Y X) = \mathcal{D}_T(Y X)$	$\mathcal{U}(x) = \frac{\mathcal{D}_T(x)}{\mathcal{D}_S(x)}, \mathcal{V}(y) = 1$
Label Shift	$\mathcal{D}_S(Y) \neq \mathcal{D}_T(Y), \mathcal{D}_S(X Y) = \mathcal{D}_T(X Y)$	$\mathcal{U}(x) = 1, \mathcal{V}(y) = \frac{\mathcal{D}_T(y)}{\mathcal{D}_S(y)}$
Domain Invariance	$\exists Z = g(X), \text{ s.t. } \mathcal{D}(Y g(X)) = \mathcal{D}(Y X), \mathcal{D}_S(Z, Y) = \mathcal{D}_T(Z, Y)$	$\mathcal{U}(x) = \frac{\mathcal{D}_T(x)}{\mathcal{D}_S(x)}, \mathcal{V}(y) = 1$
Generalized Label Shift	$\exists Z = g(X), \text{ s.t. } \mathcal{D}(Y g(X)) = \mathcal{D}(Y X), \mathcal{D}_S(Z Y) = \mathcal{D}_T(Z Y)$	$\mathcal{U}(x) = \frac{\mathcal{D}_T(X=x Z=g(x))}{\mathcal{D}_S(X=x Z=g(x))}, \mathcal{V}(y) = \frac{\mathcal{D}_T(y)}{\mathcal{D}_S(y)}$

2. Assumption: Factorizable Joint Shift

Notation. We use \mathcal{X} and \mathcal{Y} to denote the data space and the label space; x and y denote values in the two spaces; X and Y are random variables taking value in the two spaces. \mathcal{D}_S and \mathcal{D}_T denote the sample distribution in the source and the target domains. When the statement holds in both domains, we use \mathcal{D} to denote the distribution, ignoring the domain subscript. We use $\mathcal{D}(V_1)$, $\mathcal{D}(V_1|V_2)$, and $\mathcal{D}(V_1, V_2)$ to denote the marginal, conditional, and joint distribution of certain variables in the domain, respectively. We also use $\mathcal{D}(x)$ as an abbreviation of $\mathcal{D}(X = x)$ which is the probability (density) of a variable taking a certain value. Some assumptions involve the feature transformation of the data. By default, we denote g as the feature encoder and $Z \triangleq g(X)$ as the induced random variable of the feature.

Definition 2.1 (Joint Importance Weight). Joint importance weight \mathcal{W} is the ratio between joint distribution in the target domain T and that in the source domain S , i.e.,

$$\mathcal{W}(x, y) \triangleq \frac{\mathcal{D}_T(X = x, Y = y)}{\mathcal{D}_S(X = x, Y = y)}, \quad \forall x, y \in \mathcal{X} \times \mathcal{Y}.$$

Definition 2.2 (Factorizable Joint Shift (FJS)). Two domains \mathcal{D}_S and \mathcal{D}_T satisfy the *Factorizable Joint Shift* if their joint importance weight $\mathcal{W}(x, y)$ can be factorized, i.e., $\mathcal{W}(x, y) = \mathcal{U}(x)\mathcal{V}(y)$, where \mathcal{U} and \mathcal{V} are the data and the label importance factors, respectively.

We first mathematically define Factorizable Joint Shift (FJS) in Definition 2.2. Note that to avoid the discussion on invalid importance weights, throughout this paper, we assume the source domain has a larger support than the target domain, i.e., $\mathcal{D}_T(x, y) > 0 \Rightarrow \mathcal{D}_S(x, y) > 0$. As illustrated in Figure 1, FJS is a *relaxed* domain adaptation assumption and contains all four previously mentioned assumptions as subsets. In Table 1, we provide mathematical definitions of each assumption as well as its importance factorization. The factorizations for covariate shift and label shift are obvious by their definitions. Domain invariance exhibits the exact factorization as covariate shift since it is a subset of covariate shift. Yet, the factorization of generalized label shift requires more efforts. Due to the space limit, the proof is deferred to Appendix A.

In the rest of the section, we delve deeper into FJS. We study the scope of FJS from two perspectives: 1) Are there

any principles about when FJS is required? 2) What are the cases that can only be handled by FJS while beyond the scope of previous assumptions in Table 1?

2.1. When FJS is not required: an Analysis

When we study the scope of FJS, especially comparing it with GLS, which is a prior assumption most similar to FJS, we observe that the key difference between FJS and GLS is also the key difference between CS and DI. Specifically, FJS and CS generally assume there exists shift in the data marginal distributions, while GLS and DI further constrain such data distribution shift by assuming the shift can be eliminated via a proper feature encoder. However, in the current stage of research on domain adaptation, domain adversarial training techniques tailored for DI are widely adopted, resulting in the ambiguity on the boundary between DI and CS for practical problems. Thinking along this line, we observe an interesting yet common phenomenon that, for most of the learning tasks used in domain adaptation literature, DI and CS make no differences. We further identify the reason behind is that the chosen tasks all target on learning *deterministic* functions. For example, consider the most widely used computer vision tasks such as digit or object classification. Given an image, the corresponding output is quite certain. As stated in Theorem 2.1 and Theorem 2.2, we theoretically prove that such determinacy between data and label degenerates CS and FJS, reducing them to the same as DI and GLS. Proofs can be found in Appendix A.

Theorem 2.1 (Determinacy + Matched Label + CS \Rightarrow DI). If there exists a deterministic mapping from data to label, and the two domains share the same label distribution, then the source and target domain satisfying covariate shift assumption implies they satisfy domain invariance.

Theorem 2.2 (Determinacy + FJS \Rightarrow GLS). If there exists a deterministic mapping from data to label, then the source and target domain satisfying factorizable joint shift assumption implies they satisfy generalized label shift.

Our theoretical findings are meaningful in the two fronts: On the one hand, we prove that using domain invariance and generalized label shift assumptions are quite “safe” when the task of interest is deterministic; On the other hand, we question the validity of assuming the existence of domain invariant representations when dealing with tasks involving clear uncertainty, such as age estimation from images.

2.2. When you need FJS: a Motivating Example

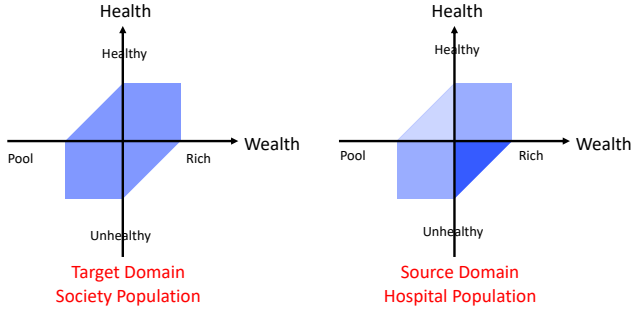


Figure 2. A toy example that satisfies FJS. The plots show the joint distributions of wealth and health. A deeper color indicates a larger probability density. **Left:** the joint distribution on the whole society which assumed to be uniform in the hexagon. **Right:** the joint distribution observed in a hospital which exhibits biases.

In this section, we construct a toy example which is within the assumption of FJS while outside of all assumptions in Table 1. The task is to predict one’s health condition y from her wealth x . With normalization, we abstract one’s health and wealth as real values from $[-1, 1]$. As shown in Figure 2, we assume our target domain is the whole society population with a uniform distribution over a hexagon on the wealth-health 2D plane. Further, the source domain is the population that observed in a hospital. Note that the source domain distribution is no longer uniform due to the bias from both data and label. Intuitively, we create biases for two reasons: (1) Along the “health” axis, populations from hospitals intrinsically induce bias towards unhealthy course. (2) Along the “wealth” axis, sampling from the hospital population also causes an implicit shift on the wealth distribution, where a preference is imposed on the rich direction. As a result, the source domain has a skewed distribution as shown in the right side of Figure 2. In this toy example, the joint importance has the following factorization, $\frac{\mathcal{D}_T(x,y)}{\mathcal{D}_S(x,y)} = U(x)V(y)$, where $U(x) = u_{\text{rich}} \cdot \mathbf{1}[x \geq 0] + u_{\text{poor}} \cdot \mathbf{1}[x < 0]$ and $V(y) = v_{\text{healthy}} \cdot \mathbf{1}[y \geq 0] + v_{\text{unhealthy}} \cdot \mathbf{1}[y < 0]$. It clearly satisfies FJS and does not satisfy CS and LS. In the Appendix, we prove this toy example is also beyond GLS.

3. Method: Joint Importance Aligning

We propose Joint Importance Aligning (JIA), which performs deep domain adaptation with factorizable domain shift. JIA employs two deep model, $U(x; \theta_u)$ and $V(y; \theta_v)$, to learn the data and label importance factor simultaneously. Once the importance is learned, we can apply any existing domain adaptation algorithms with learned weights on the source domain data during training.

In this paper, we extend the well-known Domain Adversarial Neural Networks (DANN) (Ganin et al., 2016) with JIA. We refer the resulting pipeline to as Joint Importance

Aligning Domain Adaptation (JIADA). It has three players: encoder E , predictor F and discriminator D . With learnt importance factors U, V , the three players perform the following optimization to learn to align the feature and make the prediction:

$$\begin{aligned} \max_E \min_D \quad & \mathbb{E}_{x,y \sim \mathcal{D}_S} U(x)V(y) \log(D(E(x))) \\ & + \mathbb{E}_{x \sim \mathcal{D}_T} \log(1 - D(E(x))), \\ \min_{F,E} \quad & \mathbb{E}_{x,y \sim \mathcal{D}_S} U(x)V(y) \mathcal{L}_{\text{pred}}(y, F(E(x))), \end{aligned}$$

where $\mathcal{L}_{\text{pred}}$ is the prediction loss (e.g., cross-entropy for classification or L_2 for regression). As indicated by the red color, $U(x)$ and $V(y)$ are the newly added terms comparing to the original DANN.

To estimate the importance factor, we propose two objectives for the supervised and unsupervised DA settings:

$$\begin{aligned} \mathcal{L}_{\text{sup}}(U, V) = & \mathbb{E}_{x,y \sim \mathcal{D}_S} \log(1 + U(x)V(y)) \\ & + \mathbb{E}_{x,y \sim \mathcal{D}_T} \log(1 + \frac{1}{U(x)V(y)}), \\ \mathcal{L}_{\text{unsup}}(U, V) = & \mathbb{E}_{x \sim \mathcal{D}_S} \log(1 + U(x)\tilde{V}(x)) \\ & + \mathbb{E}_{x \sim \mathcal{D}_T} \log(1 + \frac{1}{U(x)\tilde{V}(x)}). \end{aligned}$$

Note that in unsupervised objective, we define $\tilde{V}(x) \triangleq \mathbb{E}_{y \sim \mathcal{D}_S(y|x)} V(y)$ which is the label importance factor’s effect on the data.

Proposition 3.1. The importance factors U, V achieve the minima of the objective \mathcal{L}_{sup} if and only if their product equals the true joint importance. Formally, $U, V \in \arg \min_{U', V'} \mathcal{L}_{\text{sup}}(U', V')$ is equivalent to $U(x)V(y) \equiv U(x)V(y)$.

Proposition 3.2. The importance factors U, V achieve the minima of the objective $\mathcal{L}_{\text{unsup}}$ if and only if they matches the source data marginal distributions to the target domain. Formally, $U, V \in \arg \min_{U', V'} \mathcal{L}_{\text{sup}}(U', V')$ is equivalent to, for any $x, \int_y \mathcal{D}_S(x, y)U(x)V(y)dy = \mathcal{D}_T(x)$.

In Proposition 3.1 and 3.2, we theoretically show the correctness of these objectives. However there is still one difficulty remains in the unsupervised setting. Matching the data distribution of source and target domains does not guarantee to find the true importance factorization. This is because learning could collapse to a trivial solution, $U(x) = \frac{\mathcal{D}_T^X(x)}{\mathcal{D}_S^X(x)}$ and $V(y) = 1$, which ignores the label importance factor. To avoid the data importance factor overfitting at each point, we force it to have limited number of different values. Specifically, we ask the data importance factor to divide the data space into K sub-domains and assign a constant importance weight for each sub-domain. Mathematically, to achieve this, we parameterize the data importance factor by the product of a K -way classifier $C(x; \theta_c)$ and a score vector $s \in \mathbb{R}^K$, i.e., $U(x) = \sum_{k=1}^K s_k C_k(x)$ where C outputs a softmax result and C_k is the probability of data x being in the K^{th} sub-domain.

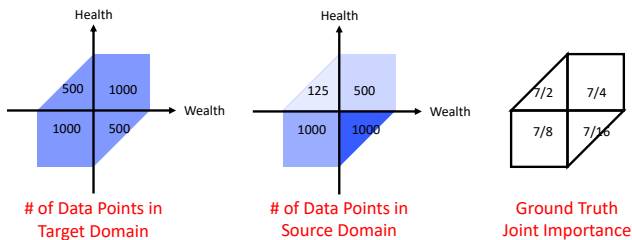


Figure 3. Statistics of the synthetic toy dataset.

Table 2. Quantitative results for unsupervised domain adaptation on the toy dataset. Best performance is shown in bold.

Assumptions	Methods	NLL (\downarrow)
	Target Only	0.600 \pm 0.002
No Shift	Source Only	0.765 \pm 0.021
Covariate Shift	SSBC (Zadrozny, 2004)	0.771 \pm 0.016
Label Shift	BBSC (Lipton et al., 2018)	0.773 \pm 0.044
Domain Invariance	DANN (Ganin et al., 2016)	0.772 \pm 0.021
Generalized Label Shift	IWDAN (Combes et al., 2020)	0.772 \pm 0.030
Factorizable Joint Importance	JIADA (Ours)	0.626 \pm 0.011

4. Experiments

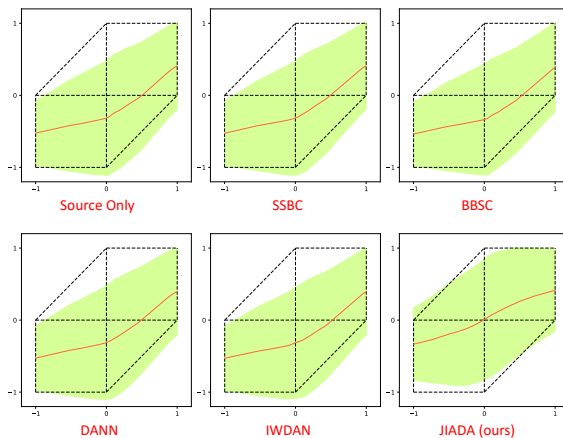
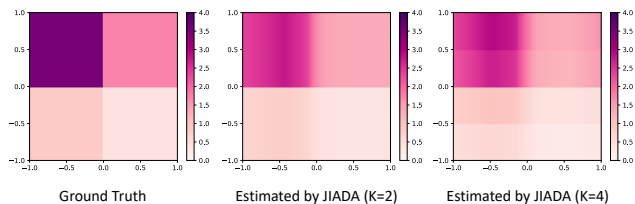
Dataset. We construct the dataset following our toy example in Section 2.2. As shown in Figure 3, we sample 3000 data points uniformly from the hexagon and treat them as target domain data. For the source domain data, there are 125, 500, 1000, and 1000 points generated in the four quadrants, respectively. The joint importance is different in each quadrant while being factorizable.

Task and Evaluation Metric. Our task is to predict one’s health y using her wealth x . For each input x , we ask the model to predict the mean $\mu(x)$ and variance $\sigma(x)$ of the label. To evaluate the model, we employ the target domain negative log-likelihood (NLL) as the criterion (lower the NLL, better the model learns $\mathcal{D}_T(Y|X)$):

$$\text{NLL}(\mu, \sigma) = \mathbb{E}_{x, y \sim \mathcal{D}_T} \left[\frac{(y - \mu(x))^2}{2\sigma(x)^2} + \log(\sqrt{2\pi}\sigma(x)) \right]. \quad (1)$$

Baselines. We use the following baselines in the unsupervised DA setting: *Sample Select Bias Correction (SSBC)* (Zadrozny, 2004), *Black Box Shift Correction (BBSC)* (Lipton et al., 2018), *Domain Adversarial Neural Networks (DANN)* (Ganin et al., 2016), and *Importance Weighting Domain Adversarial Networks (IWDAN)* (Combes et al., 2020). As listed in Table 2, each of them corresponds to one aforementioned DA assumption. We also include *Source Only* and *Target Only*, i.e., models that simply trained in the source or target domain. More details are in Appendix B.

Quantitative Results. We first show quantitative results in Table 2. As the table reports, compared to all baseline methods, JIADA significantly improves the results, achieving the best NLL. We also note that other methods do not show advantages when compared to *Source Only* model. This sheds light on the limitations of current DA assumptions when uncertainty comes into the picture. Finally, the performance of JIADA is also comparable to *Target Only* model, highlighting the effectiveness of the joint importance


 Figure 4. Qualitative results for unsupervised domain adaptation on the toy dataset. Red curves represent the predicted mean $\mu(x)$, green regions show the area deviated from the mean by $\pm\sqrt{3}\sigma(x)$.

 Figure 5. Visualization of the joint importance. We show ground truth as well as estimations by JIADA with different K .

aligning.

Qualitative Results. Furthermore, we show qualitative results in Figure 4, where we plot the estimated mean and variance for each method. We note that a random variable uniformly distributed in $[a, b]$ has the first and second moments as $\frac{a+b}{2}$ and $\frac{(b-a)^2}{12}$. If we learn this variable via Gaussian parameterization, the optimal Gaussian parameters should be $\mu = \frac{a+b}{2}$ and $\sigma = \frac{b-a}{2\sqrt{3}}$. Thus comparing $[a, b]$ with $[\mu - \sqrt{3}\sigma, \mu + \sqrt{3}\sigma]$ reflects how well the Gaussian aligns with that variable. The figure shows JIADA achieves the best estimates among all methods, where the predictions are mostly aligned with the ground truth.

Estimated Importance. We visualize the joint importance and our method’s estimation results in Figure 5. We observe that JIADA faithfully infers the underlying joint importance with small errors, and is robust to different hyperparameters. Interestingly, JIADA automatically identifies the group structure and treats $x \geq 0$ and $x < 0$ differently.

Closing Remarks. We have introduced a relaxed domain adaptation assumption through a joint importance perspective. The proposed FJS and JIADA are more generic and practical, extending the scope of current DA methods. We believe the next step is to test it in larger datasets and real-world tasks across different applications.

References

- Combes, R. T. d., Zhao, H., Wang, Y.-X., and Gordon, G. Domain adaptation with conditional distribution matching and generalized label shift. *arXiv preprint arXiv:2003.04475*, 2020.
- Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.
- Lipton, Z., Wang, Y.-X., and Smola, A. Detecting and correcting for label shift with black box predictors. In *International conference on machine learning*, pp. 3122–3130. PMLR, 2018.
- Long, M., Cao, Z., Wang, J., and Jordan, M. I. Conditional adversarial domain adaptation. *arXiv preprint arXiv:1705.10667*, 2017.
- Schölkopf, B., Janzing, D., Peters, J., Sgouritsa, E., Zhang, K., and Mooij, J. On causal and anticausal learning. *arXiv preprint arXiv:1206.6471*, 2012.
- Wang, H., He, H., and Katabi, D. Continuously indexed domain adaptation. In *International Conference on Machine Learning*, pp. 9898–9907. PMLR, 2020.
- Zadrozny, B. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 114, 2004.

A. Proofs

Proof of Domain Invariance \Rightarrow Covariate Shift. By domain invariance, we have $\mathcal{D}(Y|X = x) = \mathcal{D}(Y|Z = g(x))$ holds in both domains. We also know the joint distribution of feature and label is the same in the two domains, i.e., $\mathcal{D}_S(Y, Z) = \mathcal{D}_T(Y, Z)$ which gives the equality of condition distribution $\mathcal{D}_S(Y|Z) = \mathcal{D}_T(Y|Z)$. As a result, $\mathcal{D}_S(Y|X = x) = \mathcal{D}_S(Y|Z = g(x)) = \mathcal{D}_T(Y|Z = g(x)) = \mathcal{D}_T(Y|X = x)$ which means the covariate shift holds. \square

Proof of Importance Factorization of GLS. Under the GLS assumption, the feature $Z = g(X)$ has properties: $\mathcal{D}_S(Y|X) = \mathcal{D}_S(Y|Z)$, $\mathcal{D}_T(Y|X) = \mathcal{D}_T(Y|Z)$ and $\mathcal{D}_S(Z|Y) = \mathcal{D}_T(Z|Y)$. Hence,

$$\begin{aligned} \frac{\mathcal{D}_T(X, Y)}{\mathcal{D}_S(X, Y)} &= \frac{\mathcal{D}_T(X)\mathcal{D}_T(Y|X)}{\mathcal{D}_S(X)\mathcal{D}_S(Y|X)} = \frac{\mathcal{D}_T(X)\mathcal{D}_T(Y|Z)}{\mathcal{D}_S(X)\mathcal{D}_S(Y|Z)} \\ &= \frac{\mathcal{D}_T(X)\mathcal{D}_T(Y, Z)/\mathcal{D}_T(Z)}{\mathcal{D}_S(X)\mathcal{D}_S(Y, Z)/\mathcal{D}_S(Z)} = \frac{\mathcal{D}_T(X|Z)\mathcal{D}_T(Y)}{\mathcal{D}_S(X|Z)\mathcal{D}_S(Y)} \end{aligned}$$

Note that in the last equality, we use $\frac{\mathcal{D}(X=x)}{\mathcal{D}(Z=g(x))} = \frac{\mathcal{D}(X=x, Z=g(x))}{\mathcal{D}(Z=g(x))} = \mathcal{D}(X = x|Z = g(x))$. \square

Proof of Theorem 2.1. We construct a feature transform to demonstrate that DI assumption is satisfied. Let f denote the deterministic mapping from data to label. Now consider f as the feature transform. Apparently, function f preserve all the label information, i.e. $\mathcal{D}(Y = y|X = x) = \mathcal{D}(Y = y|f(X) = f(x))$. Second, the joint distribution of feature and label is the same in the two domains since $\mathcal{D}_S(Z = f(x), Y = y) = \mathcal{D}_S(Y = y) \cdot \mathbf{1}[y = f(x)] = \mathcal{D}_T(Y = y) \cdot \mathbf{1}[y = f(x)] = \mathcal{D}_T(Z = f(x), Y = y)$. The equality holds since we have the condition that label distributions are matched, i.e., $\mathcal{D}_S(Y = y) = \mathcal{D}_T(Y = y)$. \square

Proof of Theorem 2.2. We construct a feature transform to demonstrate that GLS assumption is satisfied. Let f denote the deterministic mapping from data to label. Now consider f as the feature transform. Apparently, function f preserve all the label information, i.e. $\mathcal{D}_S(Y = y|X = x) = \mathcal{D}_S(Y = y|f(X) = f(x))$. Second, the feature distribution conditioned on the label is the same in the two domains. $\mathcal{D}_S(f(X)|Y) = \mathcal{D}_S(Y|Y) = \mathcal{D}_T(Y|Y) = \mathcal{D}_T(f(X)|Y)$. \square

Proof of Toy Example Not Satisfying GLS. Our toy example has a property that for any two data $x_1 \neq x_2$, $\mathcal{D}(Y|X = x_1) \neq \mathcal{D}(Y|X = x_2)$. Thus a feature transformation g preserving all label information has to mapping x_1, x_2 to two different features $g(x_1)$ and $g(x_2)$. It means feature transformation is invertible, i.e. $x = g^{-1}(g(x))$. Thus $\mathcal{D}_S(X|Y) \neq \mathcal{D}_T(X|Y)$ implies $\mathcal{D}_S(g(x)|Y) \neq \mathcal{D}_T(g(x)|Y)$. Hence, it does not satisfy GLS. \square

Lemma A.1. Let \mathbb{P} and \mathbb{Q} are distributions over space \mathcal{X} which satisfy \mathbb{P} has a larger support than \mathbb{Q} . Consider the following optimization, $w^*(x) = \arg \min_w \mathbb{E}_{x \sim \mathbb{P}} \log(1 + w(x)) + \mathbb{E}_{x \sim \mathbb{Q}} \log(1 + \frac{1}{w(x)})$. We claim the optimal solution is the ratio between \mathbb{Q} and \mathbb{P} , i.e. $w(x) = \frac{\mathbb{Q}(x)}{\mathbb{P}(x)}$. The optimal value equals $2(\log 2 - \text{JSD}(\mathbb{P}||\mathbb{Q}))$ where JSD is Jensen–Shannon divergence, $\text{JSD}(\mathbb{P}||\mathbb{Q}) = H(\frac{\mathbb{P}+\mathbb{Q}}{2}) - \frac{1}{2}(H(\mathbb{P}) + H(\mathbb{Q}))$.

Proof. Considering function $f(w) = p \log(1 + w) + q \log(1 + w^{-1})$, we have $f'(w) = \frac{p}{w(1+w)}(w - \frac{q}{p})$. It indicates $f(w)$ reach's its optimal at $w = \frac{q}{p}$. Thus, the original optimization, $\int_x (\mathbb{P}(x) \log(1 + w(x)) + \mathbb{Q}(x) \log(1 + \frac{1}{w(x)})) dx$ achieves optima at $w(x) = \frac{\mathbb{Q}(x)}{\mathbb{P}(x)}$. \square

Proof of Proposition 3.1. By the Lemma A.1, we know \mathcal{L}_{sup} achieves minima when U, V satisfy $U(x)V(y) =$

$\frac{\mathcal{D}_T(x,y)}{\mathcal{D}_S(x,y)}$. With the FJS assumption, we have $\frac{\mathcal{D}_T(x,y)}{\mathcal{D}_S(x,y)} = \mathcal{U}(x)\mathcal{V}(y)$. \square

Proof of Proposition 3.2. By the Lemma A.1, we know $\mathcal{L}_{\text{unsup}}$ achieves minima when U, \tilde{V} satisfy $U(x)\tilde{V}(x) = \frac{\mathcal{D}_T(x)}{\mathcal{D}_S(x)}$ which is equivalent to $U(x) \int_y \mathcal{D}_S(y|x)V(y)dy = \frac{\mathcal{D}_T(x)}{\mathcal{D}_S(x)}$. \square

B. More Methodology Details

In this section, we detail the baseline methods introduced in Section 4, where one representative method is introduced for each traditional domain adaptation assumption.

- *Source-Only* method simply trains the model in the source domain and directly apply it to the target domain. The method assumes there exists no shift between source and target domains.
- *Sample Select Bias Correction (SSBC)* (Zadrozny, 2004) employs a domain classifier to predict the odds for a certain data point x of being from two domains, i.e., $\mathcal{D}_S(x)/\mathcal{D}_T(x)$. It then uses the reciprocal odds to re-weight the source domain data during training. Intuitively, SSBC follows the covariate shift assumption.
- *Black Box Shift Correction (BBSC)* (Lipton et al., 2018) first trains a predictor in the source domain. Then it evaluates the predictor’s confusion matrix as well as its predicted label distribution in the target domain. The label shift importance $\mathcal{D}_T(y)/\mathcal{D}_S(y)$ can be estimated based on the results. Finally, the estimated importance weights are leveraged to re-weight the training data from the source domain. We note that BBSC follows the label shift assumption.
- *Domain Adversarial Neural Networks (DANN)* (Ganin et al., 2016) adopts adversarial training to align the feature distributions between source and target domains. A predictor is trained based on such domain invariant features. The underlying assumption of DANN is the domain invariance.
- *Importance Weighting Domain Adversarial Networks (IW-DAN)* (Combes et al., 2020) goes behind the Generalized Label Shift assumption, where importance weights are estimated across each category between source and target domains. The estimated weights are further combined with adversarial training, and finally formulate an importance weighted domain adversarial training objective.