A Theory of Label Propagation for Subpopulation Shift

Tianle Cai[1 2], Ruiqi Gao[1 2], Jason D. Lee[1], Qi Lei[1]

[1]Princeton University
[2]Zhongguancun Haihua Institute for Frontier Information Technology

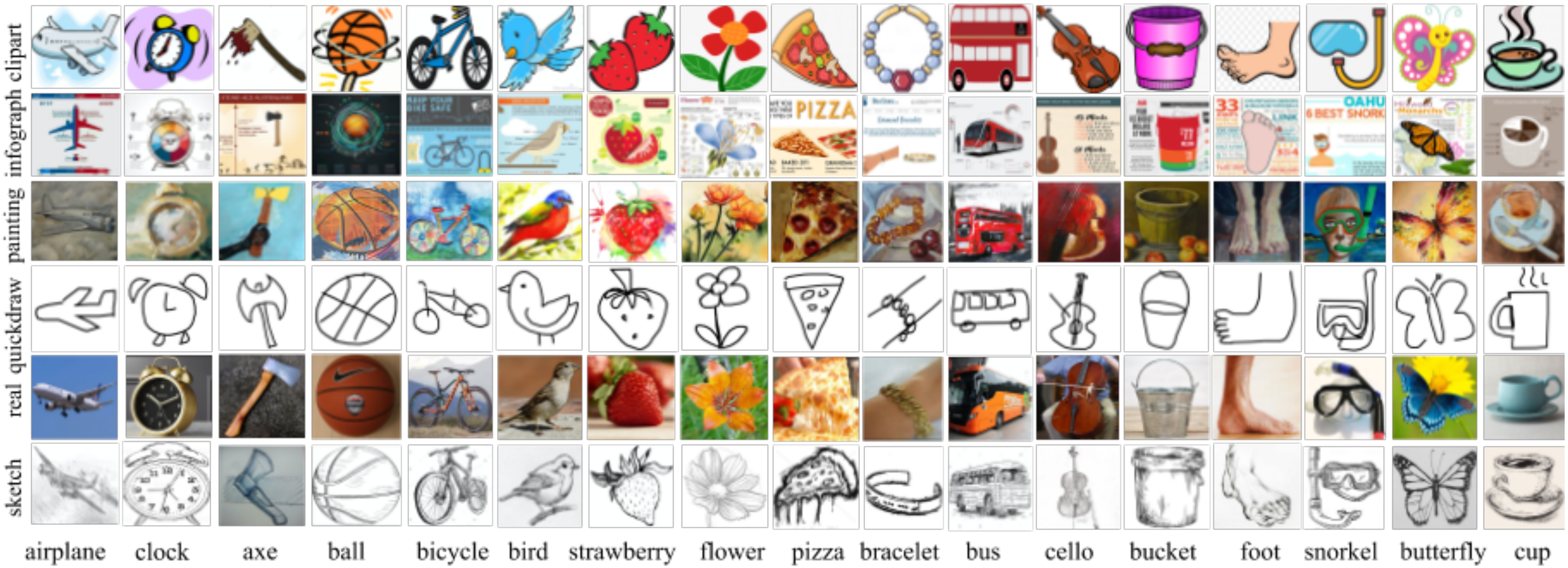
ICML 2021

Background

- In many machine learning tasks we encounter *distribution shifts* and often lots of data we have are *unlabeled.*

- *Unsupervised domain adaptation*: Source distribution S with labeled data (x, y), target distribution T with unlabeled data x.

Example Dataset: DomainNet



Example Dataset: BREEDS

Source

Target

Entity30-Passerine

Source

Target

Entity30-Tableware

Example Dataset: WILDS-FMoW

	Train			Test	
Satellite Image (x)					
Year / Region (d)	2002 / Americas	2009 / Africa	2012 / Europe	2016 / Americas	2017 / Africa
Building / Land Type (y)	shopping mall	multi-unit residential	road bridge	recreational facility	educational institution

Traditional Method: Reweight/Resample

- Consider the fundamental *covariate shift* setting, where $P_S(y\,|\,x) = P_T(y\,|\,x)$.

- Importance sampling: Using the density ratio $\beta(x) = p_T(x)/p_S(x)$ and minimize the reweighed loss $L(w) = \Sigma_{i=1}^{n}\beta(x)\ell(w, x_i)$, where x_i are i.i.d. drawn from S. However, we don't know the distribution p_S or p_T and only have samples.

- MMD distance between distributions: distance of mean in a Hilbert space.

$$D_{MMD}(X_s, X_t) = \left\| \frac{1}{n}\sum_{i=1}^{n}\phi(x_s^i) - \frac{1}{m}\sum_{j=1}^{m}\phi(x_t^j) \right\|_{\mathcal{H}}$$

$$= \left(\sum_{i,j=1}^{n} \frac{k(x_s^i, x_s^j)}{n^2} + \sum_{i,j=1}^{m} \frac{k(x_t^i, x_t^j)}{m^2} - 2\sum_{i,j=1}^{n,m} \frac{k(x_s^i, x_t^j)}{nm} \right)^{\frac{1}{2}}$$

Traditional Method: Reweight/Resample

- Reweight:

$$\min_{\boldsymbol{\beta}} \quad \left\| \frac{1}{n}\sum_{i=1}^{n} \beta_i \phi(\mathbf{x}_s^i) - \frac{1}{m}\sum_{i=1}^{m} \phi(\mathbf{x}_t^i) \right\|^2$$
$$\text{s.t.} \quad \beta_i \in [0, B] \ , \ \forall\, 1 \le i \le n$$
$$\left| \sum_{i=1}^{n} \beta_i - n \right| \le n\epsilon$$

- Resample:

$$\min_{\boldsymbol{\alpha}} \quad \left\| \frac{1}{\sum_{i=1}^{n} \alpha_i} \sum_{i=1}^{n} \alpha_i \phi(\mathbf{x}_s^i) - \frac{1}{m}\sum_{i=1}^{m} \phi(\mathbf{x}_t^i) \right\|^2$$
$$\text{s.t.} \quad \alpha_i \in \{0, 1\} \ , \ \forall\, 1 \le i \le n$$
$$\frac{1}{\sum_{i=1}^{n} \alpha_i} \sum_{i=1}^{n} \alpha_i y_c^i = \frac{1}{n}\sum_{i=1}^{n} y_c^i \ , \ \forall 1 \le c \le C$$

(Gretton et al., JRSS 2012)
(Gong et al., ICML 2013)

- Don’t work If the original distributions are very different.

Classic Theory of Domain Adaptation

- Suppose we have a hypothesis class H of functions h that maps X into Y. We measure the distance between two distributions D, D' by the H-divergence
$$d_H(D, D') = \sup_{h \in H} E_{x \sim D} h(x) - E_{x \sim D'} h(x) \,.$$
- Define the class $H \Delta H = \{ |h_1 - h_2| : h_1, h_2 \in H\}$. The bound depends on the term $d_{H\Delta H}(S, T)$. There is also an empirical divergence $\hat{d}_{H\Delta H}(S, T)$ when the expectation is taken over the empirical samples $x_1^S, \cdots, x_m^S$ and $x_1^T, \cdots, x_m^T$.
- Denote the error $\epsilon_S(h) = E_{x \sim S} |h(x) - y(x)|$ (0-1 loss if $Y = \{0,1\}$). Define the ideal joint hypothesis $h^* = \mathrm{argmin}_{h \in H}\, \epsilon_S(h) + \epsilon_T(h)$, and $\lambda = \epsilon_S(h^*) + \epsilon_T(h^*)$.

(Ben-David et al, 2010)

Classic Theory of Domain adaptation

- Main theorem: For all $h \in H$

$$\epsilon_T(h) \leq \epsilon_S(h) + d_{H\Delta H}(S, T) + \lambda.$$

- Proof:

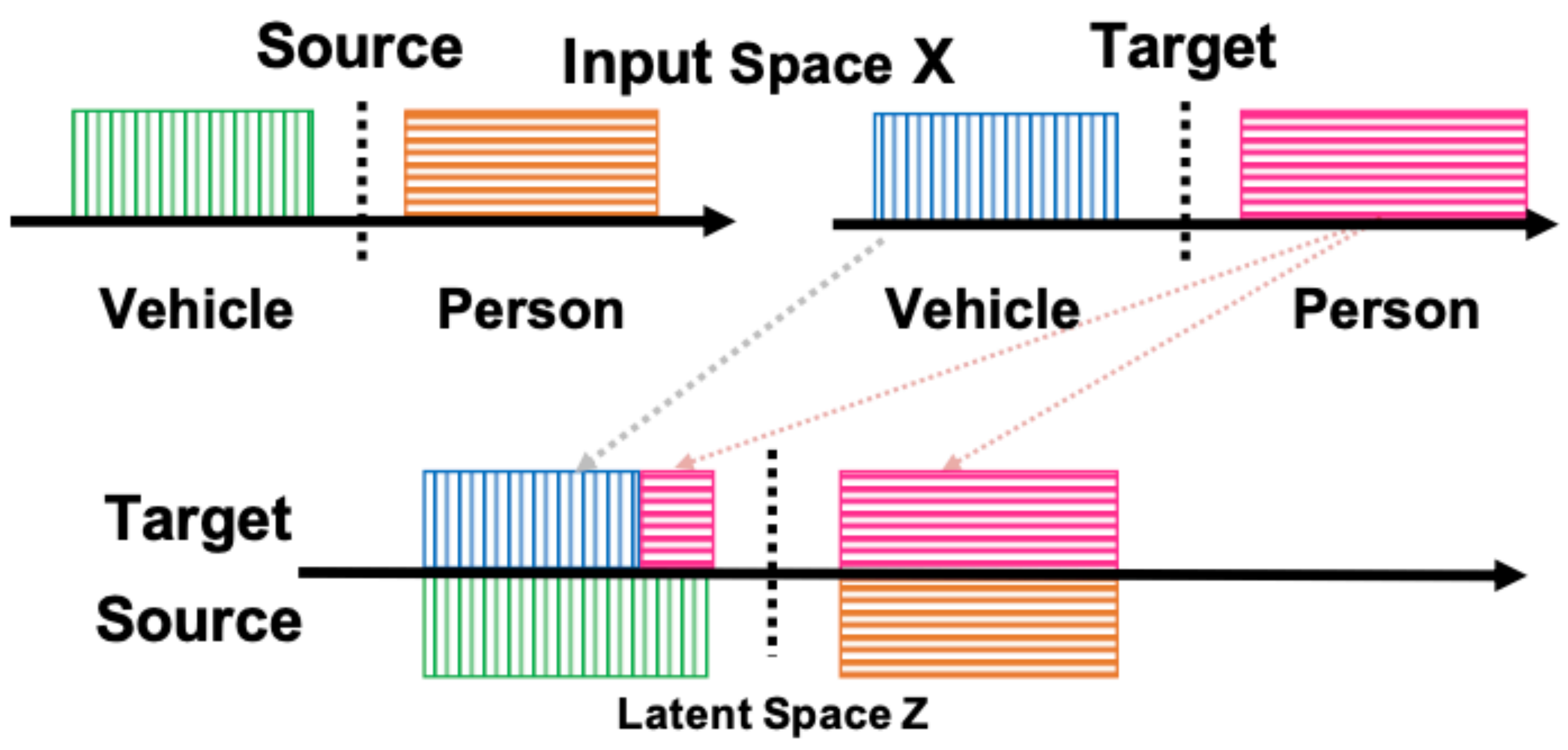
$$\begin{aligned}
\epsilon_T(h) &= E_S(|h(x) - y(x)|) \leq E_S(|h(x) - h^*(x)|) + E_S(|h^*(x) - y(x)|) \\
&\leq E_T(|h(x) - h^*(x)|) + d_{H\Delta H}(S, T) + E_S(|h^*(x) - y(x)|) \\
&\leq E_T(|h(x) - y(x)|) + E_T(|h^*(x) - y(x)|) + d_{H\Delta H}(S, T) + E_S(|h^*(x) - y(x)|) \\
&= \epsilon_S(h) + d_{H\Delta H}(S, T) + \lambda\,.
\end{aligned}$$

Distribution Matching

- In Ben-David bound, the discrepancy $d_{H\Delta H}(S, T)$ can contribute to big error.
- Traditional methods consider all kinds of transforms to make S and T similar.
- Classic distribution matching method in deep learning: Learn an invariant representation $x \to z \to \hat{y}$, where the distribution of $z = g(x)$ for $x \sim S$ and $x \sim T$ are trained to be the same, i.e. for any measurable subset A of the representation space, $P_{x\sim S}[g(x) \in A] = P_{x\sim T}[g(x) \in A]$.
- Theoretical explanation: Take $H = F \circ G$, where the function class G contains g that matches the distributions, then $d_{H\Delta H}(S, T) = 0$ provably.

Caveats for Distribution Matching

- In Ben-David bound, forcing distribution to match minimizes $d_{H\Delta H}(S, T)$ but might cause the other term $\lambda = \min_{h \in H} \epsilon_S(h) + \epsilon_T(h)$ to explode.
- Intuitive explanation: Matching z may not preserve the right information for y.
- Example from [1]: When label shift (shift in the marginal distribution $P(y)$) is present, the classifier over an exactly aligned representation provably fails.
- Example from [2]: Even if there is no label shift, there are many ways of distribution matching that causes y to mismatch.



[1] Domain Adaptation with Asymmetrically-Relaxed Distribution Alignment, Wu et al., ICML 2019.
[2] Rethinking Distributional Matching Based Domain Adaptation, Li et al, 2020.

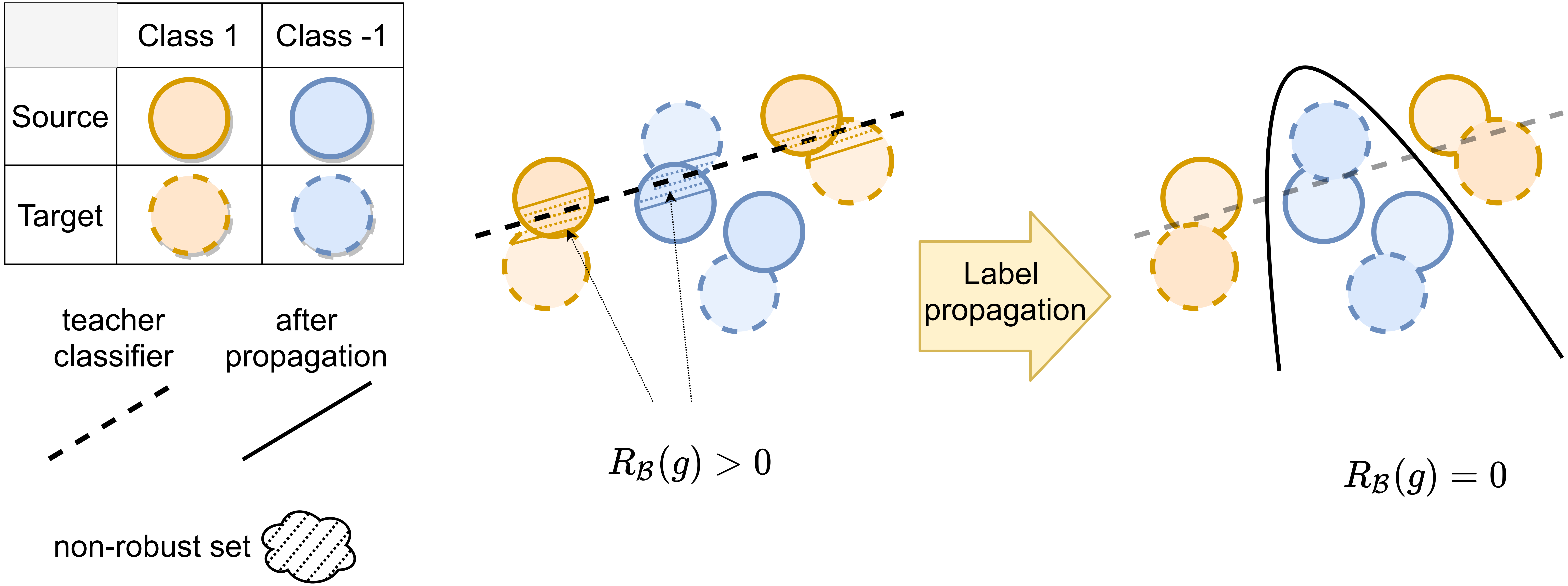
Any possible frameworks other than distribution matching?

Subpopulation Shift

- A new model and framework for distribution shift.
- Characterize source and target by $S = S_1 \cup \cdots \cup S_m$ and $T = T_1 \cup \cdots \cup T_m$, where each S_i and T_i are correspondent in a certain sense.[1]
- Subpopulation shift is ubiquitous in practical tasks, e.g. "Poodles eating dog food" in the source and "Labradors eating meat" in the target. Or it can be implicit and hard to elaborate by words, like from ImageNet to ImageNet-v2.
- Even in the architecture of distribution matching methods, subpopulation shift on the representation z should be allowed to exist.

[1] We abuse the notations S, S_i, etc. to indicate either the distribution or the support set.

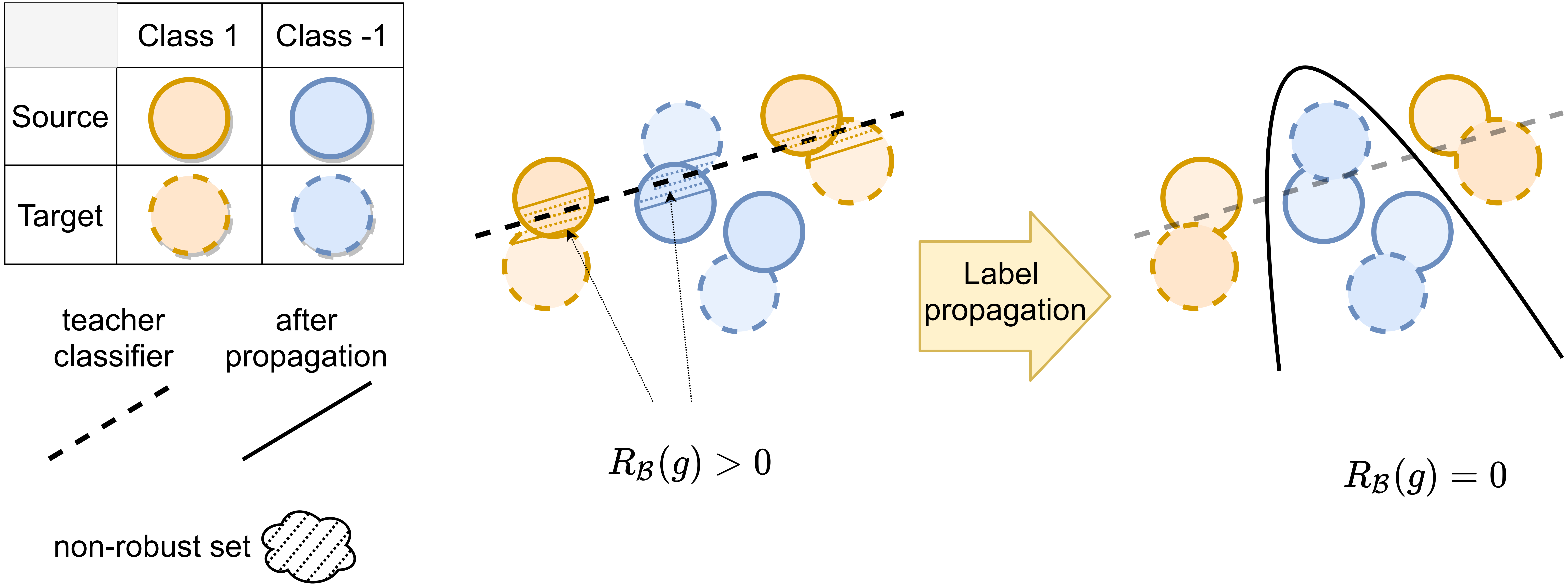
Algorithmic Framework



- Suppose there is a (possibly noisy) teacher classifier g_{tc} on source. Goal: Propagate the label information from S to T based on unlabeled data.

- In this toy illustration, each $S_i \cup T_i$ forms a regular connected component.

Algorithmic Framework



- Consistency regularizer $R_B(g)$ measures the amount of non-robust set of g, i.e. points whose predictions by g is inconsistent in a small neighborhood.

- g_{tc} + A proper consistency regularization = Label propagation!

Formal Assumption on Subpopulations

- We consider a multi-class classification problem $X \to Y = \{1, \cdots, K\}$, S and T the source and target distribution on X. We have a classifier g_{tc} on S that contains all label information.

- Assume $supp(S) = \cup_{i=1}^{m} S_i$, $supp(T) = \cup_{i=1}^{m} T_i$, and $S_i \cap T_j = \emptyset$ for $i \neq j$. We assume the ground truth $g^*(x)$ is consistent on $S_i \cup T_i$, denoted y_i.

- Assume $P_{x \sim S_i}[g_{tc}(x) = y_i] \geq P_{x \sim S_i}[g_{tc}(x) = k] + \gamma, \forall k \in \{1, \cdots, K\} \backslash \{y_i\}$.

- Assume $P_T[T_i]/P_S[S_i] \leq r, \forall i \in \{1, \cdots, m\}$.

Algorithm

- We expect the predictions to be stable under a suitable set of input transformations $B(x) \subset X$, and use the following consistency regularization:

$$R_B(g) := P_{x \sim \frac{1}{2}(S+T)}[\, \exists x' \in B(x), \text{s.t. } g(x) \neq g(x')].$$

- B can be a distance-based neighborhood set or some data augmentations A,
and can take the general form $B(x) = \{x' : \exists A \text{ such that } d(x', A(x)) \leq rad\}$.

- Define $L_{01}^S(g, g_{tc}) := P_{x \sim S}[g(x) \neq g_{tc}(x)]$, our algorithm is

$$g = \operatorname{argmin}_{g: X \to Y, g \in G} L_{01}^S(g, g_{tc}) \text{ s.t. } R_B(g) \leq \mu,$$

where μ is a constant satisfying $R_B(g^*) < \mu$, which is expected to be small.

Assumption: Expansion

- The expansion property proposed in [1], some geometric regularity on $S_i \cup T_i$ w.r.t. B, is needed for local consistency regularization to propagate globally.

- Define the neighborhood set $N(x) := \{x' \,|\, B(x) \cap B(x') \neq \emptyset\}$, and for a set $A \subset X$ define $N(A) := \cup_{x \in A} N(x)$.

- Definition of (a, c)-multiplicative expansion: For $a \in (0,1)$, $c > 1$, any i, any $A \in S_i \cup T_i$ with $P_{\frac{1}{2}(S+T)}[A] \leq a$, we have $P_{\frac{1}{2}(S_i+T_i)}[N(A)] \geq \min(cP_{\frac{1}{2}(S_i+T_i)}[A], 1)$.

[1] Wei, C., Shen, K., Chen, Y., and Ma, T. (2021). Theoretical analysis of self-training with deep networks on unlabeled data.

Upper-Bounding the loss on Target

- *Main theorem*: Guarantee on the target error $\epsilon_T(g) = P_{x\sim T}[g(x) \neq g^*(x)]$.
- Based on $(1/2, c)$-multiplicative expansion, we have

$$\epsilon_T(g) \leq \max\left(\frac{c+1}{c-1}, 3\right)\frac{8r\mu}{\gamma}.$$

- Remark: The accuracy of g can actually improve upon the accuracy on g_{tc}, suppose μ is small.

Proof Sketch

- The *robust set* is $RS(g) := \{x \,|\, g(x) = g(x'), \forall x' \in B(x)\}$.
- The *majority class* on the i-th component is
$$y_i^{Maj} := \text{argmax}_{k\in[K]} P_{\frac{1}{2}(S_i+T_i)}[RS(g) \cap \{x \,|\, g(x) = k\}].$$
- And we let $\widetilde{M} := \cup_{i=1}^{m} (S_i \cup T_i) \cap \{x \,|\, g(x) \neq y_i^{Maj}\}$ be the *minority set*.
- Upper bound the minority set: $P_{\frac{1}{2}(S+T)}[\widetilde{M}] \leq \max((c+1)/(c-1), 3)\mu$.
- Define the inconsistent components
$$I = \{i \in [m] \,|\, P_{x\sim S_i}[g(x) \neq g_{tc}(x)] > P_{x\sim S_i}[g_{tc}(x) \neq y_i] + \gamma/2\}.$$
- Separately bound $\epsilon_T(g) = \Sigma_{i\in I}\epsilon_T^i(g) + \Sigma_{i\in[m]\setminus I}\epsilon_T^i(g) \leq 8rP_{\frac{1}{2}(S+T)}[\widetilde{M}]/\gamma$.

Finite Sample Bound

- Finite-sample bounds can be obtained by off-the-shelf generalization bounds.
- For a neural network $f : X \to R^K$ and its induced classifier g, we use the all-layer margin $m(f, x, y)$ from [2] and the robust margin

$$m_B(f, x) = \min_{x' \in B(x)} m(f, x', \operatorname{argmax}_i f(x)_i).$$

- The algorithm now becomes

$$g = \operatorname{argmin}_{g:X \to Y, g \in G} P_{x \sim \hat{S}}[m(f, x, g_{tc}(x)) \leq t]$$

$$\text{s.t. } P_{x \sim \frac{1}{2}(\hat{S}+\hat{T})}[m_B(f, x) \leq t] \leq \mu \,.$$

[2] Wei, C. and Ma, T. (2019). Improved sample complexities for deep networks and robust classification via an all-layer margin.

Finite Sample Bound

- Based on $(1/2, c)$-multiplicative expansion, we have

$$\epsilon_T(g) \le \frac{8r}{\gamma}\left(\max\left(\frac{c+1}{c-1}, 3\right)\hat{\mu} + \Delta\right).$$

where

$$\Delta = \widetilde{O}\left(\left(P_{x\sim\hat{S}}[m(f^*, x, g_{tc}(x)) \le t] - L_{01}^{\hat{S}}(g^*, g_{tc})\right) + \frac{\sum_i \sqrt{q}\|W_i\|_F}{t\sqrt{n}} + \sqrt{\frac{\log(1/\delta) + p\log n}{n}}\right)$$

$$\hat{\mu} = \mu + \widetilde{O}\left(\frac{\sum_i \sqrt{q}\|W_i\|_F}{t\sqrt{n}} + \sqrt{\frac{\log(1/\delta) + p\log n}{n}}\right).$$

Generalized Subpopulation Shift

- The previous framework can be applied to a much more general setting: As long as we perform consistency regularization on an unlabeled dataset that covers both source and target, we should be able to propagate labels.

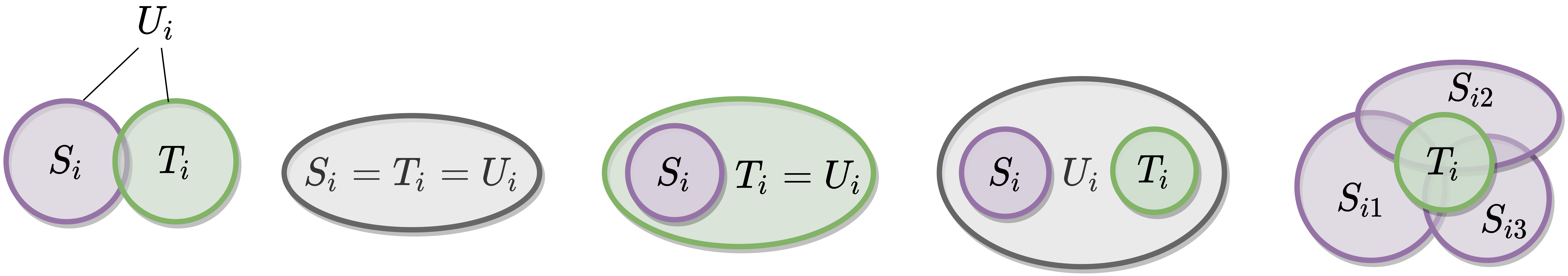
- The distributions are of the following structure: $supp(S) = \cup_{i=1}^{m} S_i$, $supp(T) = \cup_{i=1}^{m} T_i$, $supp(U) = \cup_{i=1}^{m} U_i$, $S_i \cup T_i \subset U_i$, and $U_i \cap U_j = \emptyset$ for $i \neq j$.[1]

- $R_B(g)$ is on U and expansion is assumed to hold on $\{U_i\}_{i=1}^{m}$ now.

- Defining "coverage": There exists a $\kappa \geq 1$ s.t. for any $A \subset X$, we have $P_{S_i}(A) \leq \kappa P_{U_i}(A)$ and $P_{T_i}(A) \leq \kappa P_{U_i}(A)$.

[1] Where the ground-truth labels on U_i is consistent.

Generalized Subpopulation Shift



(a) Unsupervised domain adaptation (b) Semi-supervised learning or self-supervised denoising (c) Domain expansion (d) Domain extrapolation (e) Multi-source domain adaptation or domain generalization

- The previous results hold on a multitude of setting, with an extra multiplicative constant κ in the final bound.

Experiments: Subpopulation Shift Dataset

- ENTITY-30 task from BREEDS tasks. (Subset of ImageNet, where classes don't shift but subclasses shifts.)
- We use FixMatch, an existing consistency regularization method. We also leverage SwAV, an existing unsupervised representation learned from ImageNet, where there can be a better structure of subpopulation shift. We compare with popular distribution matching methods like DANN and MDD.

Method	Source Acc	Target Acc
Train on Source	91.91 ± 0.23	56.73 ± 0.32
DANN (Ganin et al., 2016)	92.81 ± 0.50	61.03 ± 4.63
MDD (Zhang et al., 2019)	92.67 ± 0.54	63.95 ± 0.28
FixMatch (Sohn et al., 2020)	90.87 ± 0.15	72.60 ± 0.51

Experiments: Classic DA Dataset

- Office-31 and Office-home.
- We add consistency regularization (FixMatch) to MDD, and observed improvement to the distribution matching method.

Method	A → W	D → W	W → D	A → D	D → A	W → A	Average
MDD	94.97±0.70	98.78±0.07	100±0	92.77±0.72	75.64±1.53	72.82±0.52	89.16
MDD+FixMatch	95.47±0.95	98.32±0.19	100±0	93.71±0.23	76.64±1.91	74.93±1.15	**89.84**

Table 2: Performance of MDD and MDD+FixMatch on Office-31 dataset.

Method	Ar → Cl	Ar → Pr	Ar → Rw	Cl → Ar	Cl → Pr	Cl → Rw	Pr → Ar	Pr → Cl	Pr → Rw	Rw → Ar	Rw → Cl	Rw → Pr	Average
MDD	54.9±0.7	74.0±0.3	77.7±0.3	60.6±0.4	70.9±0.7	72.1±0.6	60.7±0.8	53.0±1.0	78.0±0.2	71.8±0.4	59.6±0.4	82.9±0.3	68.0
MDD+FixMatch	55.1±0.9	74.7±0.8	78.7±0.5	63.2±1.3	74.1±1.8	75.3±0.1	63.0±0.6	53.0±0.6	80.8±0.4	73.4±0.1	59.4±0.7	84.0±0.5	**69.6**

Table 3: Performance of MDD and MDD+FixMatch on Office-Home dataset.

Takeaway Message

Consistency-based methods (e.g. semi-supervised learning methods like FixMatch) can help domain adaptation, especially in the presence of subpopulation shift!

Thanks!

https://tianle.website/