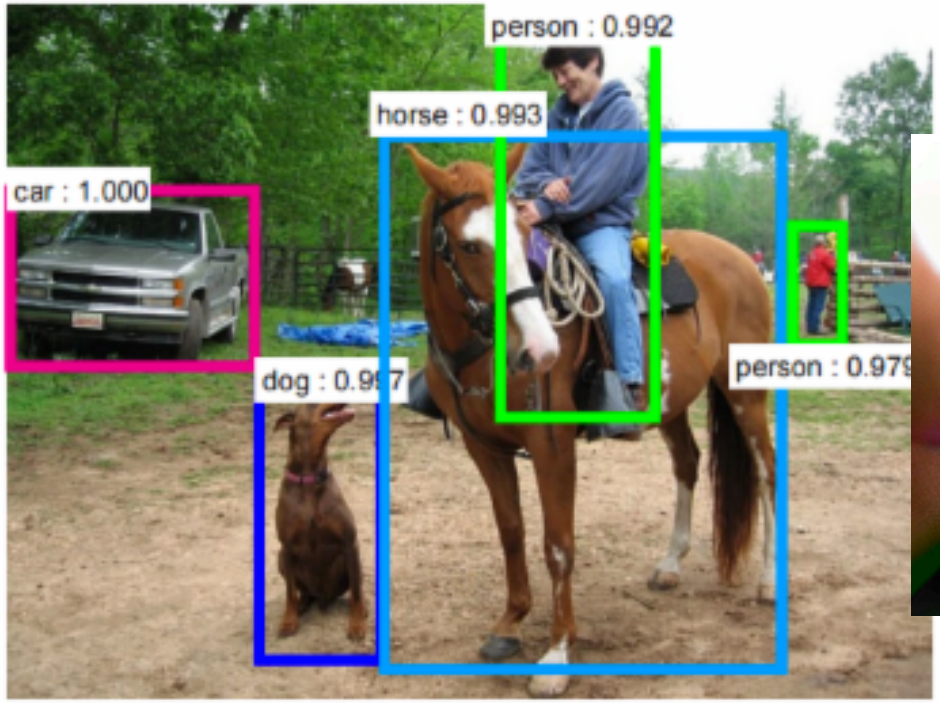
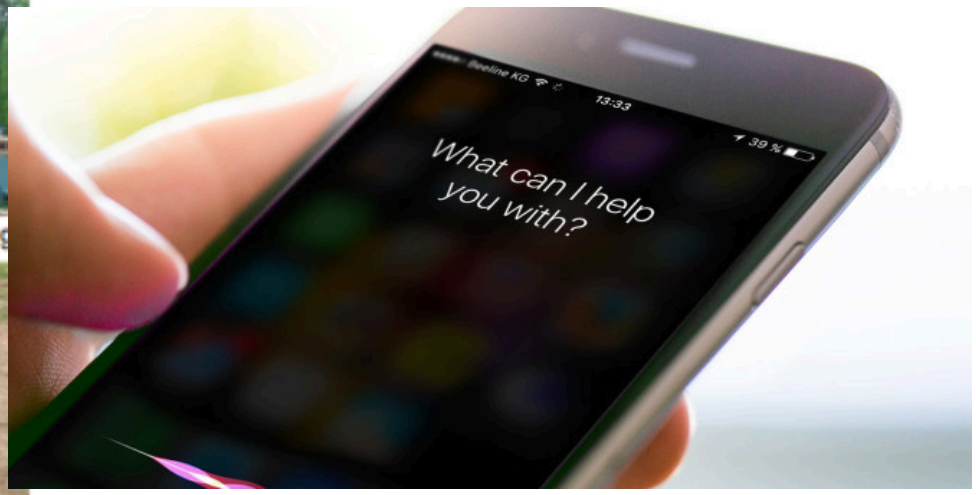
Causal Domain Adaptation

Mingming Gong, Jiaxian Guo, Chenwei Ding

因果读书会, Nov 22, 2020

Deep Learning





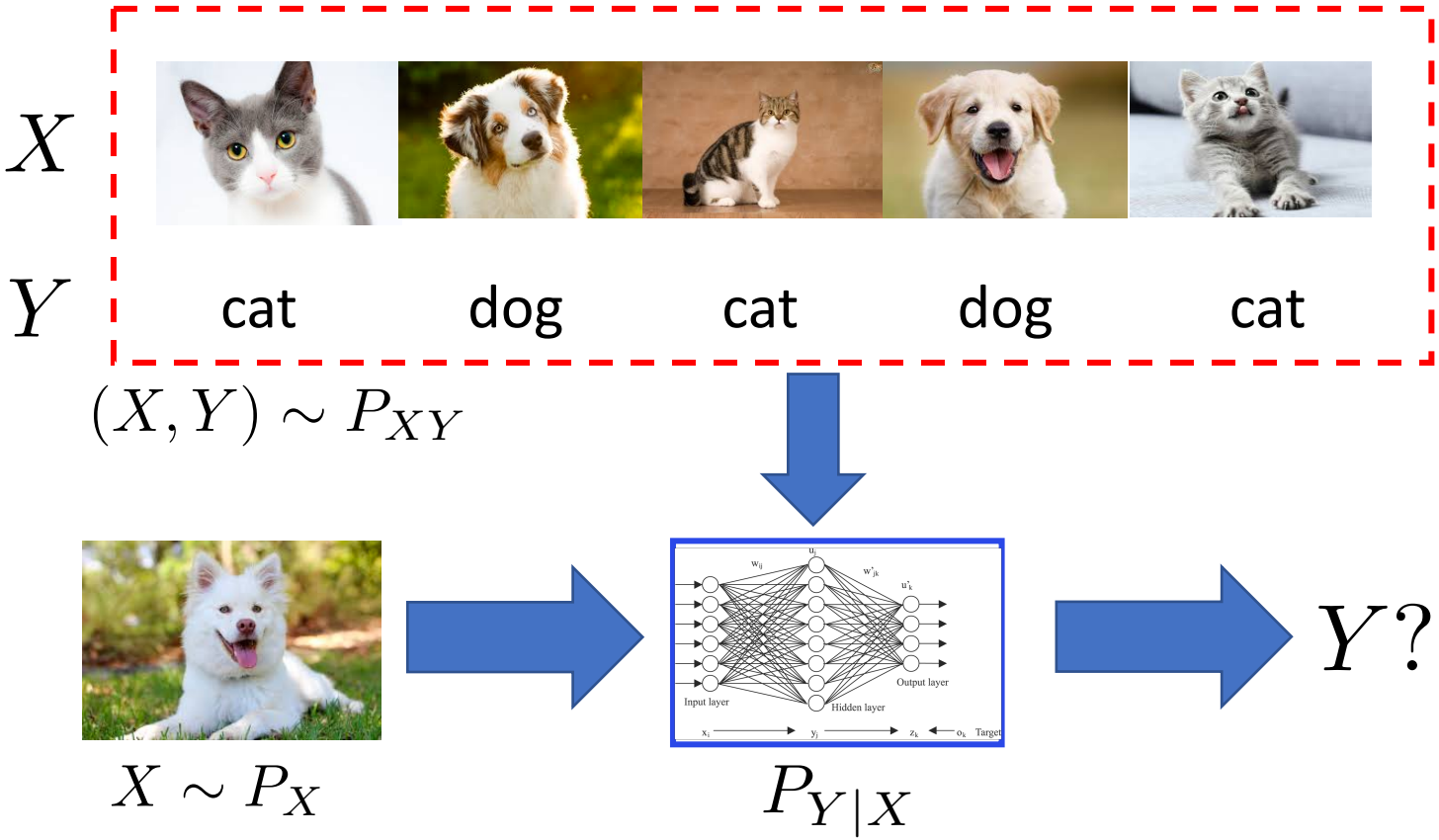
Total price: $231.96

Add all three to Cart

Add all three to List

Supervised Learning

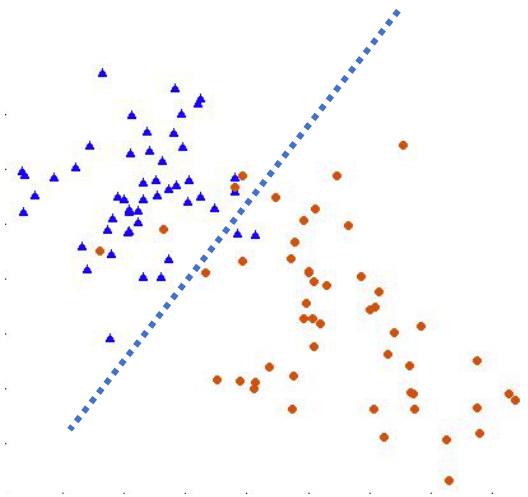
- Machine learning systems often assume training and test set have the same distribution .



Domain Adaptation (DA)

X – feature (covariate)

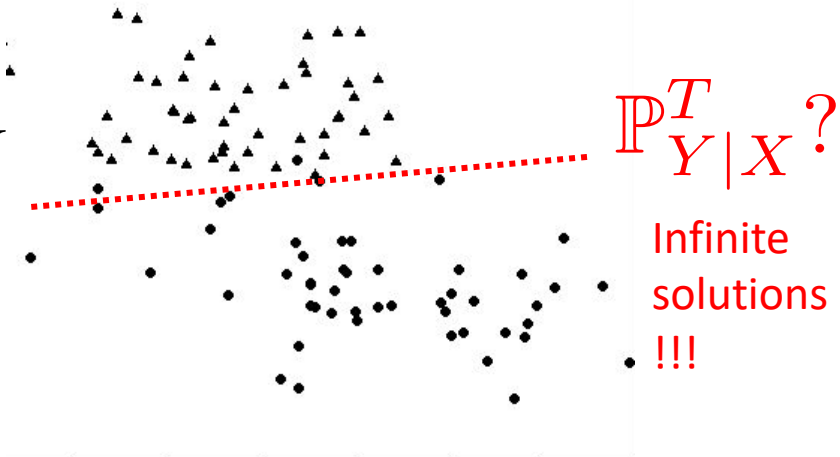
Y – label (target)



$\mathbb{P}^S_{XY}$

Source Domain(Training)

$\mathbb{P}^S_{XY} \neq \mathbb{P}^T_{XY}$



$\mathbb{P}^T_{Y|X}$?

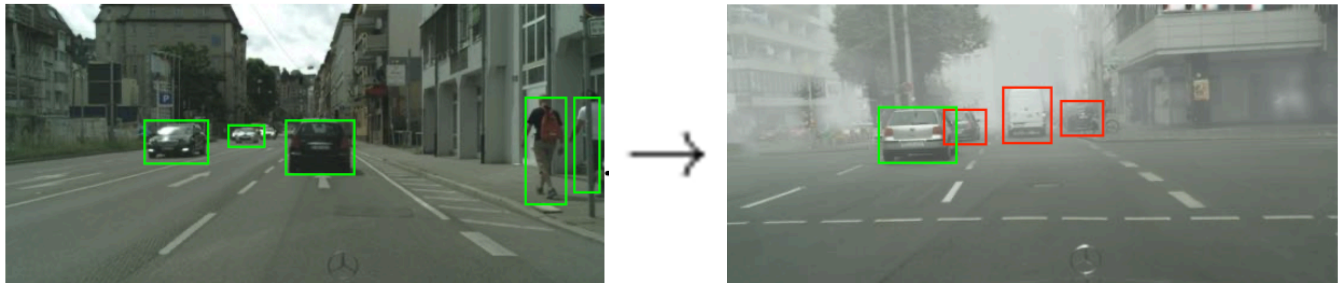
Infinite solutions !!!

$\mathbb{P}^T_X$

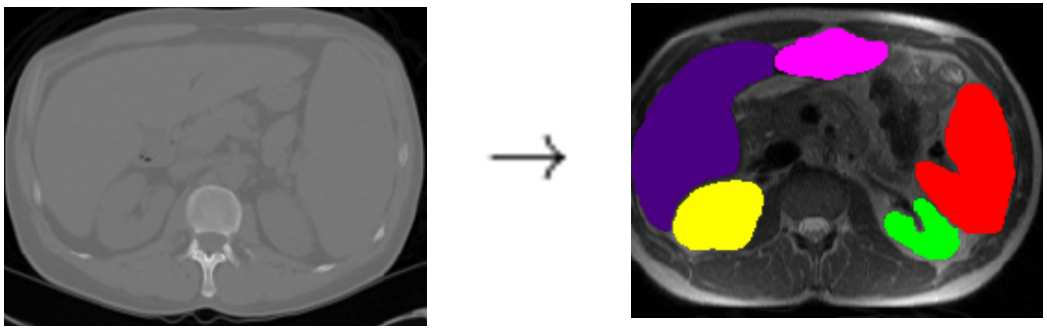
Target Domain(Test)

Examples

Computer Vision



Medical Image Analysis



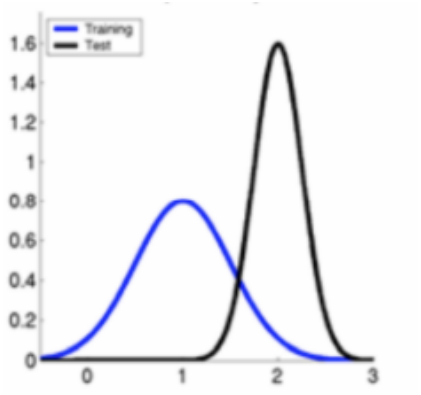
Natural language processing

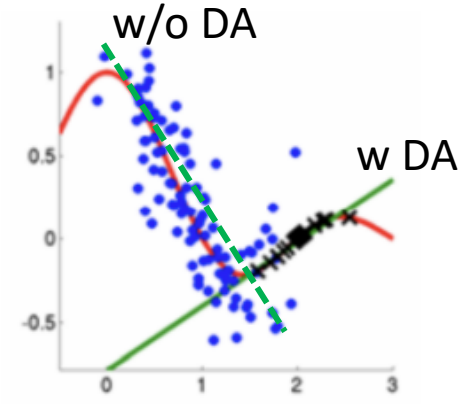
Outline

- Background
- Causal Understanding of DA
- Target Shift correction
- Conditional Invariant Components
- Domain Adaptation as Inference on Graphical Models
- Causal Discovery from Multiple Domains
- Conclusions

DA: Covariate Shift

$$\mathbb{P}_{XY} = \mathbb{P}_X \mathbb{P}_{Y|X} \quad \mathbb{P}_X^S \neq \mathbb{P}_X^T \quad \mathbb{P}_{Y|X}^S = \mathbb{P}_{Y|X}^T$$





Instance reweighting

$$R(f) = \int \mathbb{P}^T(x,y)\ell(f(x),y)dxdy$$

$$= \int \frac{\mathbb{P}^T(x)}{\mathbb{P}^S(x)} \mathbb{P}^S(x,y)\ell(f(x),y)dxdy$$

- Implicit assumption: The support of $\mathbb{P}^T(x)$ contained in that of $\mathbb{P}^S(x)$.
- Why $\mathbb{P}_X^S \neq \mathbb{P}_X^T$ $\mathbb{P}_{Y|X}^S = \mathbb{P}_{Y|X}^T$?
- What if $\mathbb{P}_{Y|X}^S \neq \mathbb{P}_{Y|X}^T$?
- Density ratio estimation

[Shimodaira 2000][Sugiyama 2008][Huang 2007]

DA: Invariant Components (IC)

Relaxed covariate shift $\mathbb{P}^S_X \neq \mathbb{P}^T_X \quad \mathbb{P}^S_{Y|X} \neq \mathbb{P}^T_{Y|X}$

$$\exists X' = h(X) \Rightarrow \begin{array}{l} \mathbb{P}^S(X') \approx \mathbb{P}^T(X') \\ \mathbb{P}^S(Y|X') \approx \mathbb{P}^T(Y|X') \end{array}$$

- Invariant components (feature adaptation) [Pan 2011][Si 2010]

$$min_h Div(\mathbb{P}^S(h(X)), \mathbb{P}^T(h(X)))$$

  - Div: MMD, adversarial, optimal transport… [Courty 2017]
  - h: linear projection, MLP, CNNs, … [Long 2015][Ganin2015]

- Can we always find $\mathbb{P}^S(X') \approx \mathbb{P}^T(X')$?

- No guarantee of $\mathbb{P}^S(Y|X') \approx \mathbb{P}^T(Y|X')$, semi-supervised constraints that relate $\mathbb{P}^T(X') \; and \; \mathbb{P}^T(Y|X')$
  - Self/Co/Tri-training [Kumar 2018][Saito 2017][Xie 2018][Zhang 2019]
  - Virtual adversarial training [Shu 2018]

DA: Domain Mapping

Relaxed covariate shift $\mathbb{P}_X^S \neq \mathbb{P}_X^T \quad \mathbb{P}_{Y|X}^S \neq \mathbb{P}_{Y|X}^T$

$$\exists X' = h(X) \Rightarrow \begin{matrix} \mathbb{P}^S(X') \approx \mathbb{P}^T(X) \\ \mathbb{P}^S(Y|X') \approx \mathbb{P}^T(Y|X) \end{matrix}$$

- Domain mapping (pixel adaptation)

$$min_h Div(\mathbb{P}^S(h(X)), \mathbb{P}^T(X))$$

h →

- Can we always find $\mathbb{P}^S(X') \approx \mathbb{P}^T(X)$?

- No guarantee of $\mathbb{P}^S(Y|X') \approx \mathbb{P}^T(Y|X)$
  - Cycle-consistency [Huffman 2018]
  - Geometry consistency [Fu 2018][Zhao 2019]
  - Content distortion [Bousmalis 2017]

Outline

- Background
- Causal Understanding of DA
- Target Shift correction
- Conditional Invariant Components
- Domain Adaptation as Inference on Graphical Models
- Causal Discovery from Multiple Domains
- Conclusions

Modularity

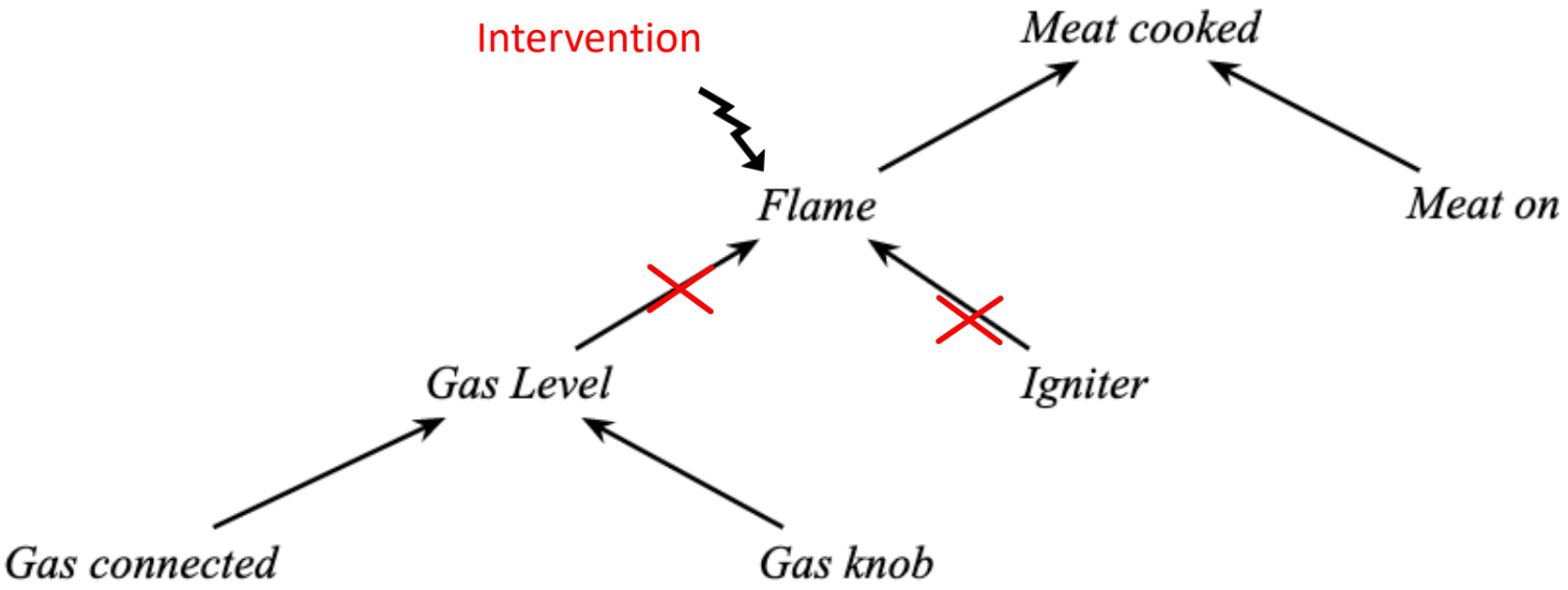


FIGURE 3

Invariance:
P(Gas Level|Gas Connected, Gas knob) is invariant
P(Meat cooked|Flame, Meat on) is invariant

J. Pearl. Causality. Cambridge University Press, 2000.

Modularity

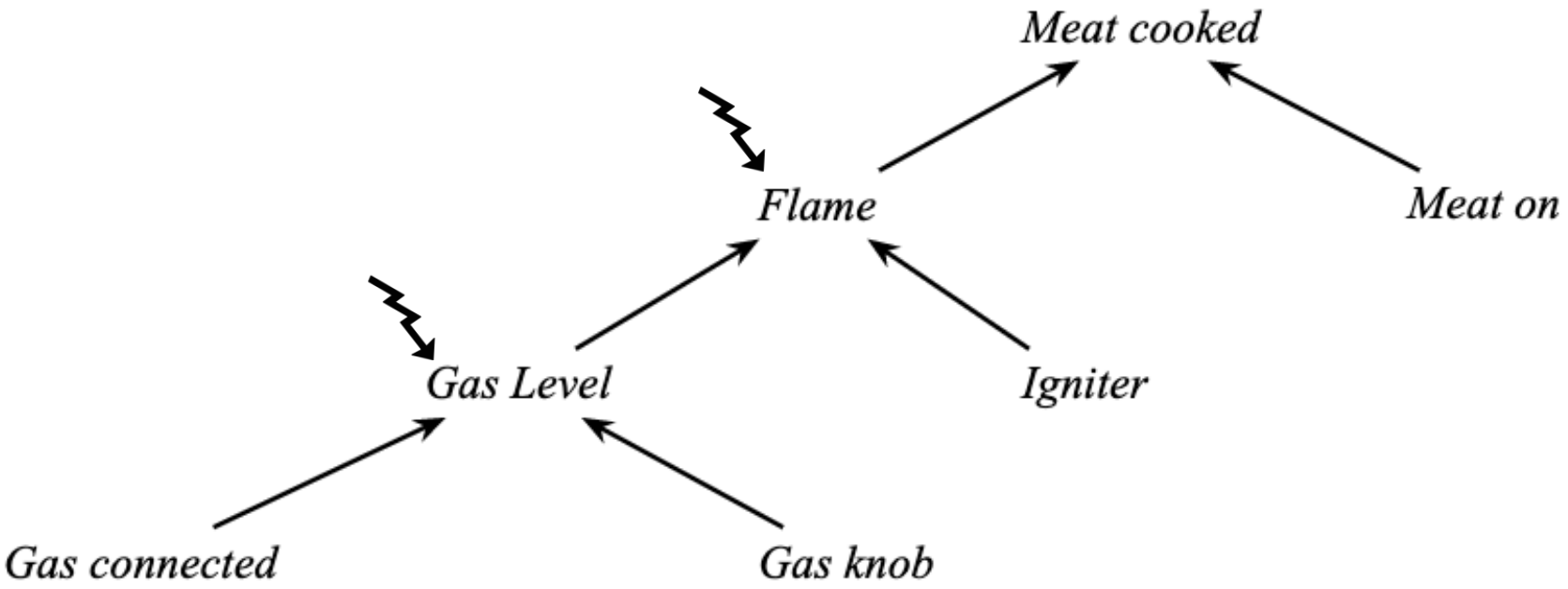


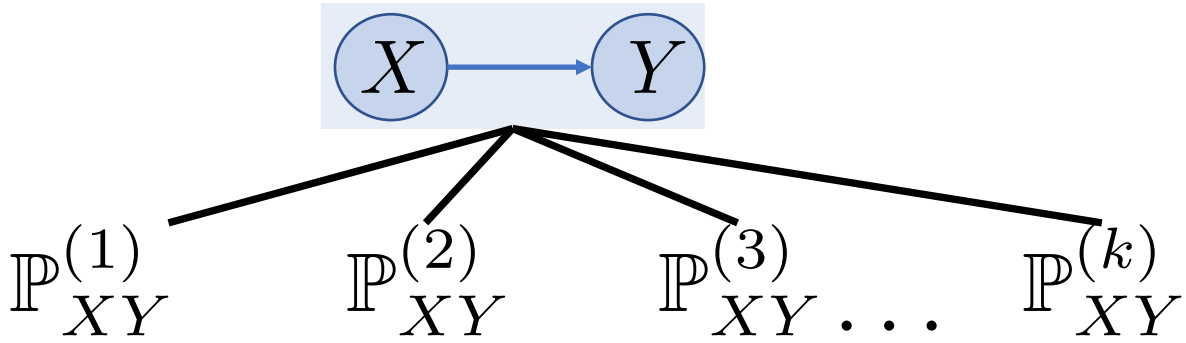
FIGURE 3

Independent mechanism changes:
P(Flame|Gas Level, Igniter) changes
P(Gas Level|Gas Connected, Gas knob) is changes
P(Meat cooked|Flame, Meat on) is invariant

J. Pearl. Causality. Cambridge University Press, 2000.

Causal Model for DA

- Bridge between probability distributions



- Independent causal mechanism

$C - Cause$

$E - Effect$

$\mathbb{P}_C$ $\mathbb{P}_{E|C}$ (causal mechanism)

Without confounder, $\mathbb{P}_C$ and $\mathbb{P}_{E|C}$ do not contain information about each other.

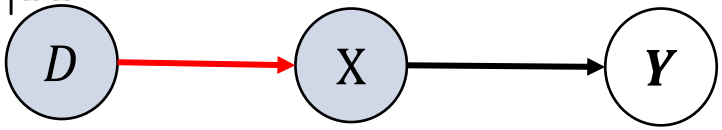
Schölkopf, Bernhard, et al. "On causal and anticausal learning." *arXiv preprint arXiv:1206.6471* (2012).

$$X \rightarrow Y$$

$$\mathbb{P}_{XY} = \mathbb{P}_X \mathbb{P}_{Y|X}$$

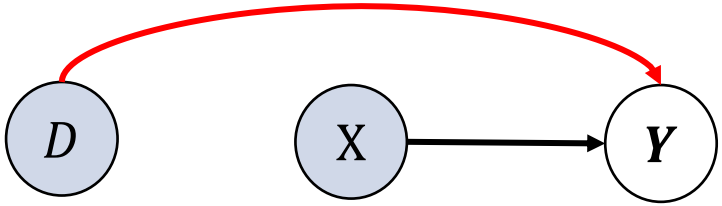
D – Domain Index

Covariate Shift



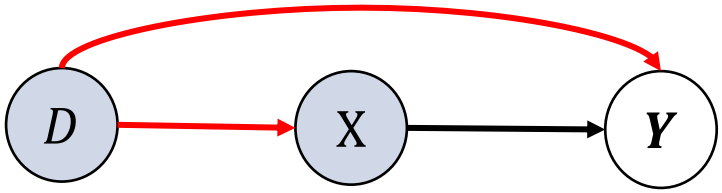
$$\mathbb{P}_X^S \neq \mathbb{P}_X^T \quad \mathbb{P}_{Y|X}^S = \mathbb{P}_{Y|X}^T$$

Classifier Shift



$$\mathbb{P}_X^S = \mathbb{P}_X^T \quad \mathbb{P}_{Y|X}^S \neq \mathbb{P}_{Y|X}^T$$

Covariate + Classifier Shift



$$\mathbb{P}_X^S \neq \mathbb{P}_X^T \quad \mathbb{P}_{Y|X}^S \neq \mathbb{P}_{Y|X}^T$$

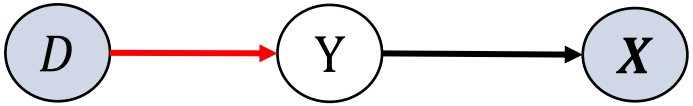
No clue as to find $\mathbb{P}_{Y|X}^T$ with one source domain

$Y \rightarrow X$

- Y is usually the cause of X (especially for classification)

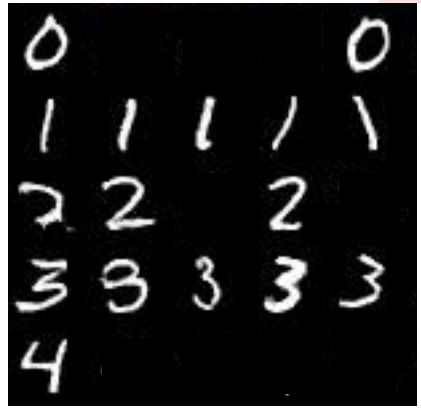
$$\mathbb{P}_{XY} = \mathbb{P}_{X|Y}\mathbb{P}_Y$$

Target Shift



$\mathbb{P}_Y^S \neq \mathbb{P}_Y^T$ $\quad \mathbb{P}_{X|Y}^S = \mathbb{P}_{X|Y}^T$

$$\mathbb{P}_X^S \neq \mathbb{P}_X^T$$
$$\mathbb{P}_{Y|X}^S \neq \mathbb{P}_{Y|X}^T$$



Covariate shift does *not* hold !!!

Gong, Zhang, et al. Domain adaptation with conditional transferable components. ICML16

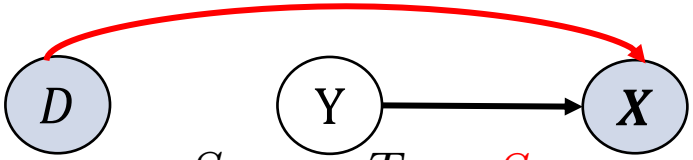
Zhang, Kun, et al. Domain adaptation under target and conditional shift. ICML13

$Y \rightarrow X$

- Y is usually the cause of X (especially for classification)

$\mathbb{P}_{XY} = \mathbb{P}_{X|Y}\mathbb{P}_Y$

Conditional Shift



$\mathbb{P}_Y^S = \mathbb{P}_Y^T \quad \mathbb{P}_{X|Y}^S \neq \mathbb{P}_{X|Y}^T$

$\mathbb{P}_X^S \neq \mathbb{P}_X^T$

$\mathbb{P}_{Y|X}^S \neq \mathbb{P}_{Y|X}^T$

Covariate shift does *not* hold !!!

Gong, Zhang, et al. Domain adaptation with conditional transferable components. ICML16

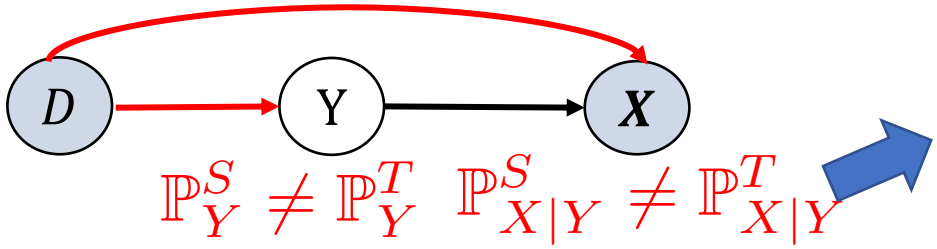
Zhang, Kun, et al. Domain adaptation under target and conditional shift. ICML13

$Y \rightarrow X$

- Y is usually the cause of X (especially for classification)

$$\mathbb{P}_{XY} = \mathbb{P}_{X|Y}\mathbb{P}_Y$$

Generalized Target Shift



$$\mathbb{P}_X^S \neq \mathbb{P}_X^T$$
$$\mathbb{P}_{Y|X}^S \neq \mathbb{P}_{Y|X}^T$$

Covariate shift does *not* hold !!!

Zhang, Kun, et al. Domain adaptation under target and conditional shift. ICML13
Gong, Zhang, et al. Domain adaptation with conditional transferable components. ICML16

Outline

- Background
- Causal Understanding of DA
- Target Shift correction
- Conditional Invariant Components
- Domain Adaptation as Inference on Graphical Models
- Causal Discovery from Multiple Domains
- Conclusions

Scenarios of Target Shift Problem

Discrete label cases (classification tasks):

- Flu prediction: Given symptoms as X, and predict if one patient have flu.
  - The rate of flu P_Y varies with time (e.g. Flu season)
  - But the manifestations of the disease $P_{X|Y}$ might not change.

Continuous label cases (regression tasks):

- Location prediction: Given images as X, and predict the location of some objects.
- Direction prediction: Given images as X, and predict the direction of some objects.

High dimensional label cases:

- 3D human pose estimation: Given 2D pose as X, and estimate the 3D pose of humans.
- 2D location prediction: Given images as X, and predict the 2D direction of some objects.

Existing Methods

- KMM（Zhang 2013.）:
  - 1. Quantify the label distribution P_Y^T at the Test set.
    - It rewrites the test data distribution P_X^T as follows:

$$P_X^T(x) = \int_y P^T(x|y)P^T(y)dy = \int_y P^S(x|y)P^T(y)dy = \underbrace{\int_y P^S(x,y)\frac{P^T(y)}{P^S(y)}dy}_{P_X^{\text{new}}}$$

Zhang etc. “Domain adaptation under target and conditional shift” , ICML, 2013.

Existing Methods

- KMM（Zhang 2013.）:
  - 1. Quantify the label distribution P_Y^T at the Test set.
    - It rewrites the test data distribution P_X^T as follows:

$$P_X^T(x) = \int_y P^T(x|y)P^T(y)dy = \int_y P^S(x|y)P^T(y)dy = \underbrace{\int_y P^S(x,y)\frac{P^T(y)}{P^S(y)}dy}_{P_X^{\text{new}}}$$

    - and then use function $\beta(y)$ to estimate label weights $\frac{P_Y^T}{P_Y^S}$, and build a new data distribution $P_X^{new} = \int_y P^S(x,y)\beta(y)dy$

Zhang etc. “Domain adaptation under target and conditional shift” , ICML, 2013.

Existing Methods

- KMM（Zhang 2013.）:
  - 1. Quantify the label distribution P_Y^T at the Test set.
    - It rewrites the test data distribution P_X^T as follows:

$$P_X^T(x) = \int_y P^T(x|y)P^T(y)dy = \int_y P^S(x|y)P^T(y)dy = \underbrace{\int_y P^S(x,y)\frac{P^T(y)}{P^S(y)}dy}_{P_X^{\text{new}}}$$

    - and then use function $\beta(y)$ to estimate label weights $\frac{P_Y^T}{P_Y^S}$, and build a new data distribution $P_X^{new} = \int_y P^S(x,y)\beta(y)dy$
    - Lastly, $\beta(y)$ is estimated by reducing the MMD distance between Test data distribution P_X^T and P_X^{new}.

$$||\frac{1}{n_t}\sum_{i=1}^{n_t}\psi(x_i^t) - \frac{1}{n_s}\sum_{i=1}^{n_s}\beta(y_i^s)\psi(x_i^s)||_{\mathcal{H}}^2$$

    - As such, the estimated test label distribution is $\hat{P_Y^T} = \beta(y) * P_Y^S$

Zhang etc. “Domain adaptation under target and conditional shift” , ICML, 2013.

Existing Methods

- KMM（Zhang 2013.）:
  - 2. Adapt the trained model F to the Test set according to the quantified label distribution $\hat{P}_Y^T$ using reweight methods.
    - Retrain a new function $F^{weighted}$ with weighted dataset:

$$P_{XY}^{weighted} = P^S(x,y)\beta(y)$$

Zhang etc. “Domain adaptation under target and conditional shift” , ICML, 2013.

Existing Methods

- KMM（Zhang 2013.）:
  - 2. Adapt the trained model F to the Test set according to the quantified label distribution $\hat{P}_Y^T$ using reweight methods.
    - Retrain a new function $F^{weighted}$ with weighted dataset:

$$P_{XY}^{weighted} = P^S(x, y)\beta(y)$$

- Disadvantage:
  - Not compatible with large-scale data because its computational cost is quadratic in the sample size.

Zhang etc. “Domain adaptation under target and conditional shift” , ICML, 2013.

Existing Methods

- BBSE（Lipton 2018）:

  It focus on the classification tasks, and estimates $\beta(y)$ by:

  $$\beta = \hat{\mathbf{C}}^{-1}\hat{\mathbf{q}}$$

  where $\hat{\mathbf{C}}$ is the confusion matrix of the mapping function of $F: X \to Y$ on the training set, and $\hat{\mathbf{q}}$ is the predicted labels of $F: X \to Y$ on the test set.

- Disadvantage:

  - It only works for discrete target shift scenarios.

Lipton etc. “Detecting and Correcting for Label Shift with Black Box Predictors” , ICML, 2018.

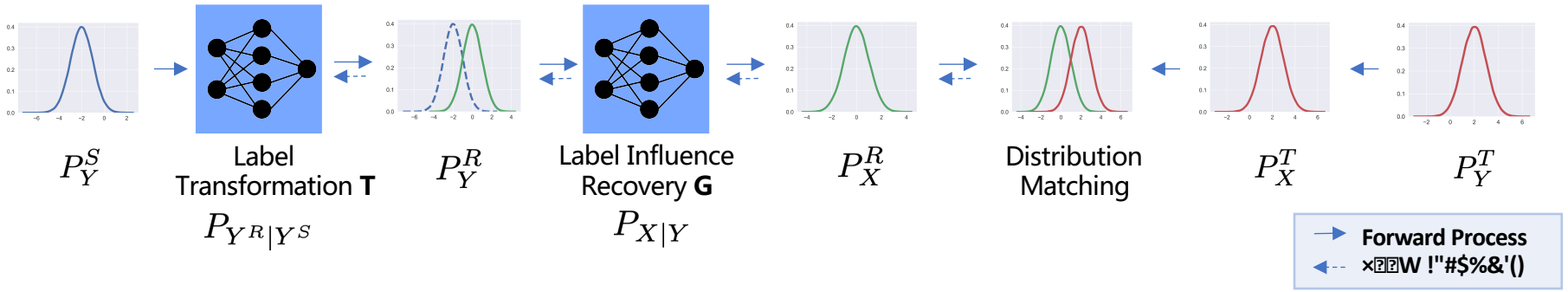
Our Methods

- Motivation:
  - Existing methods are
    - computationally infeasible for large-scale data
    - restricted to shift correction for discrete labels
    - neglect the invariant conditional distribution $P^S_{X|Y} = P^T_{X|Y}$

Our Methods

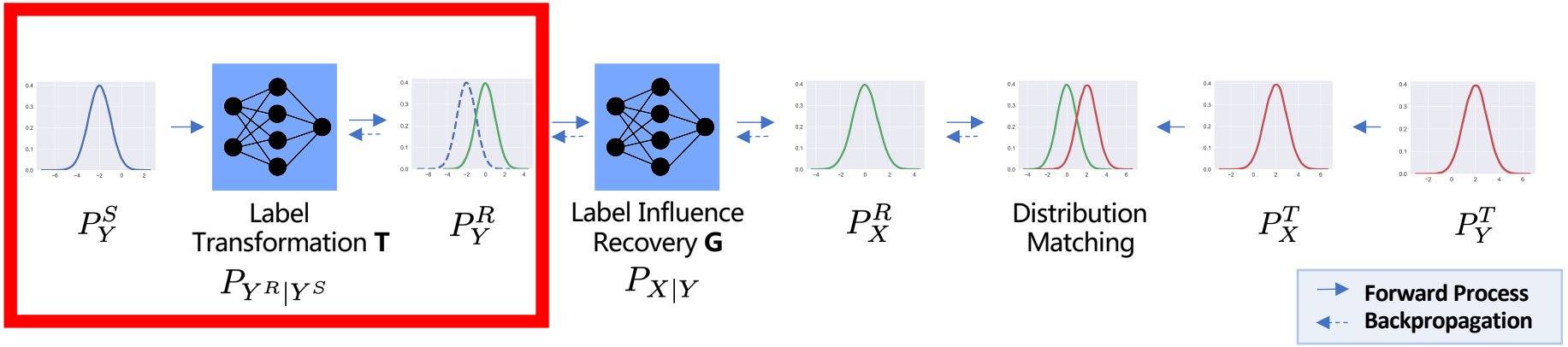
- Motivation:
  - Existing methods are
    - computationally infeasible for large-scale data
    - restricted to shift correction for discrete labels
    - neglect the invariant conditional distribution $P^S_{X|Y} = P^T_{X|Y}$
  - Label Transformation Framework (LTF) based on neural networks.
    - Utilize the invariant conditional distribution
    - can handle continuous, discrete, and even multi-dimensional labels in a unified way
    - Is scalable to large data

Overview



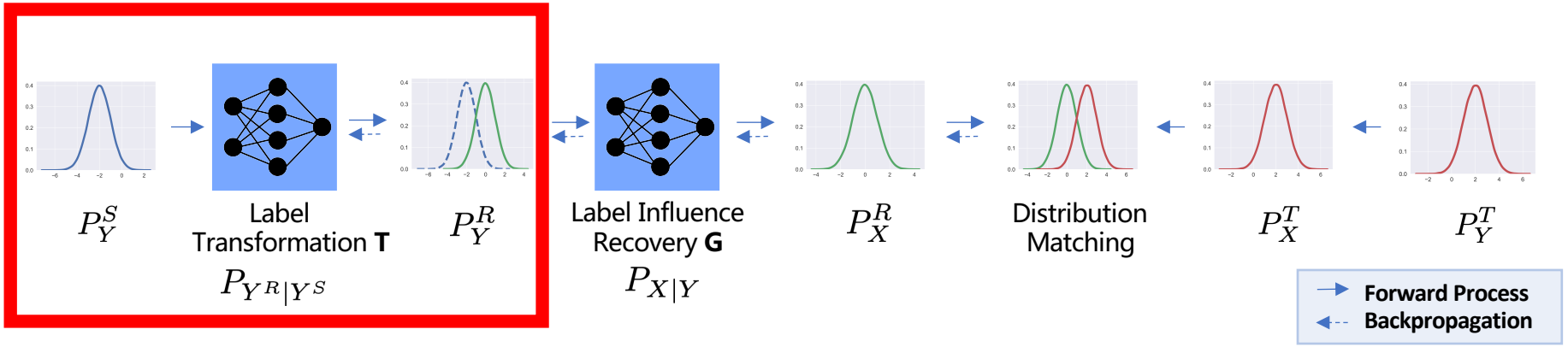
- The illustration of framework. The conditional distribution $P_{X|Y}$ is a constant function

Overview



$$P_Y^R = \int_{y^s} P_{Y^R|Y^S}(y^r|y^s) P_Y^S(y^s) dy^s$$

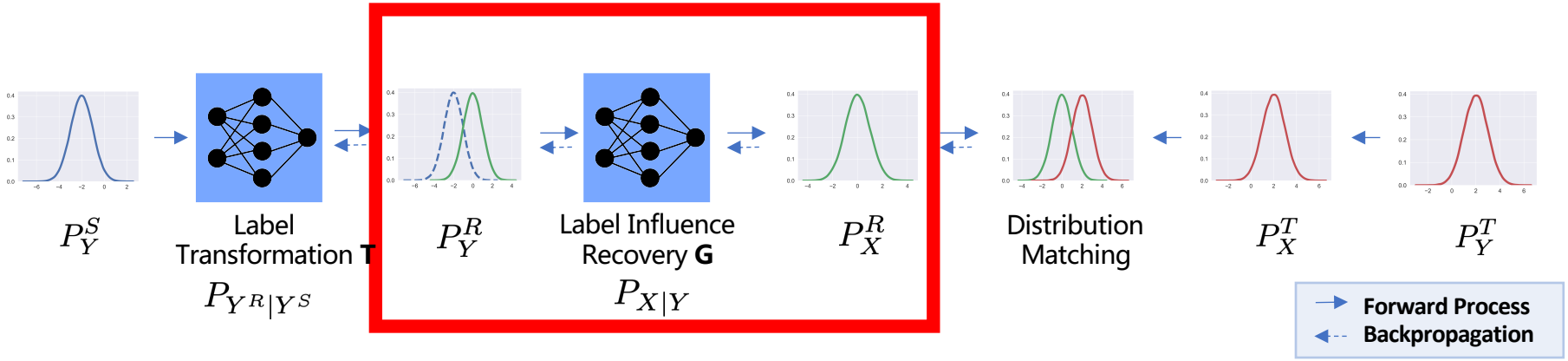
Overview



$$P_Y^R = \int_{y^s} P_{Y^R|Y^S}(y^r|y^s) P_Y^S(y^s) dy^s$$

$$\begin{cases} y^r = a y^s + b & \text{Linear} \\ y^r = y^s * \beta(y^s) = \dfrac{y^r}{y^s} = y^s * (a + \dfrac{b}{y^s}) & \text{Non-Linear} \end{cases}$$

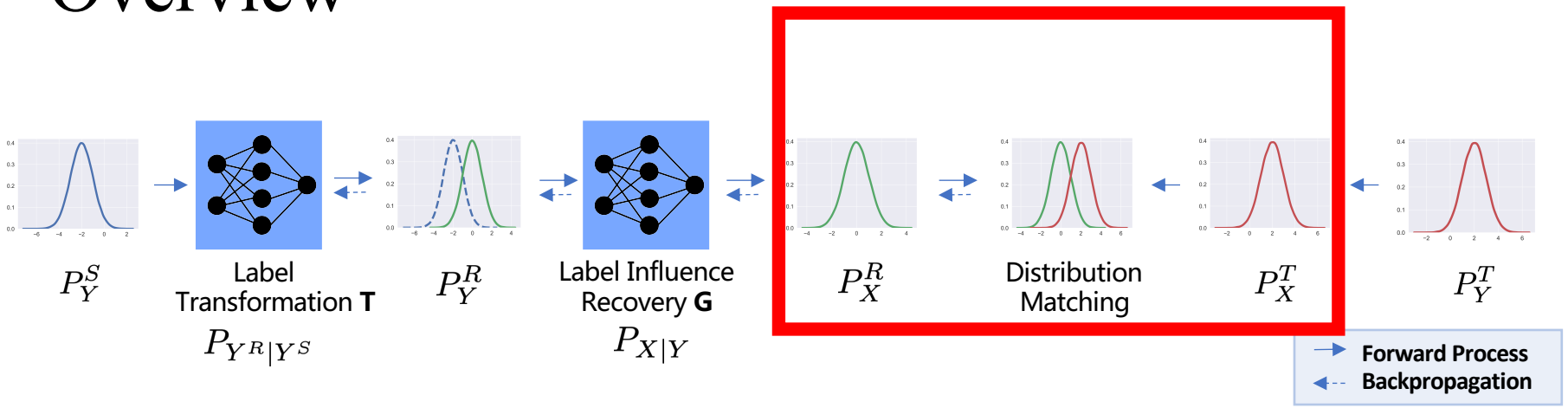
Overview



$$P_X^R = \int_y P_{X|Y}(x|y^r) P_Y^R(y^r) dy^r$$
$$= \int_{y^r} P_{X|Y}(x|y^r) \int_{y^s} P_{Y^R|Y^S}(y^r|y^s) P_Y^S(y^s) dy^s dy^r$$

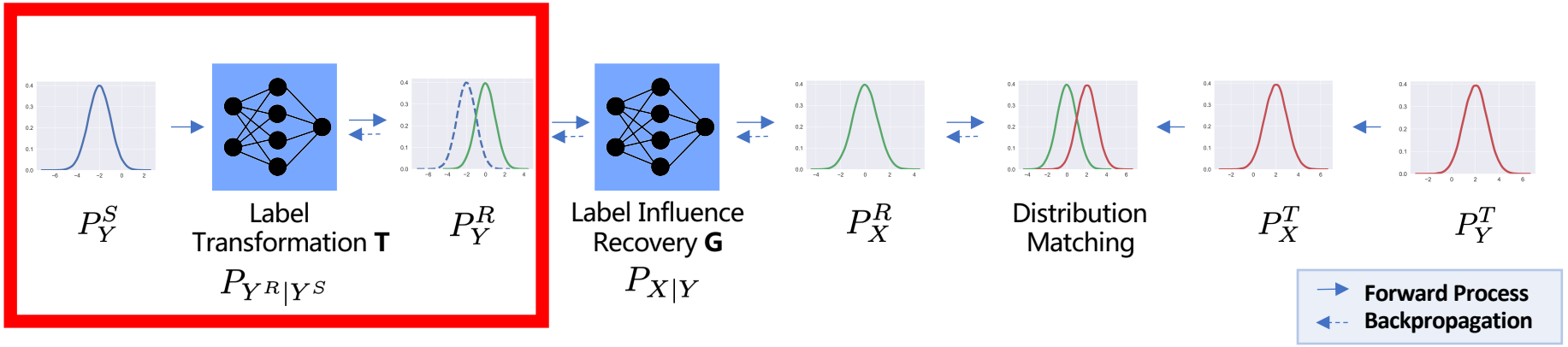
- Then the pre-trained **Label Influence Recovery Model G** generates the new sample distribution P_X^R according to the transformed label distribution P_Y^R .

Overview



- By matching the target domain P_X^T with generated P_X^R, our method implicitly matches P_Y^R with P_Y^T.

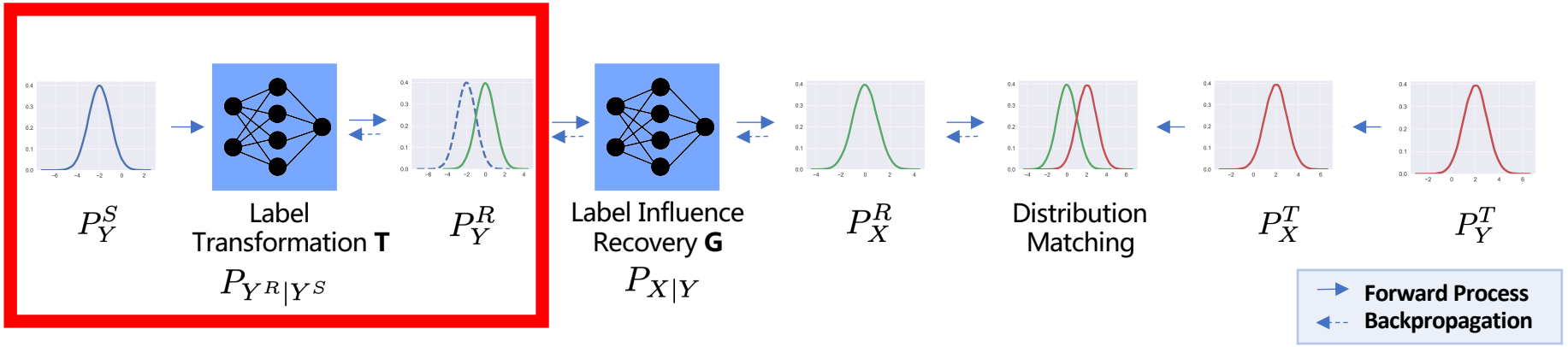
Details: Label Transformation



- Instead of estimating the density ratio $\beta(y)$, our framework uses **Label Transformation Model T** to model the target label distribution implicitly.

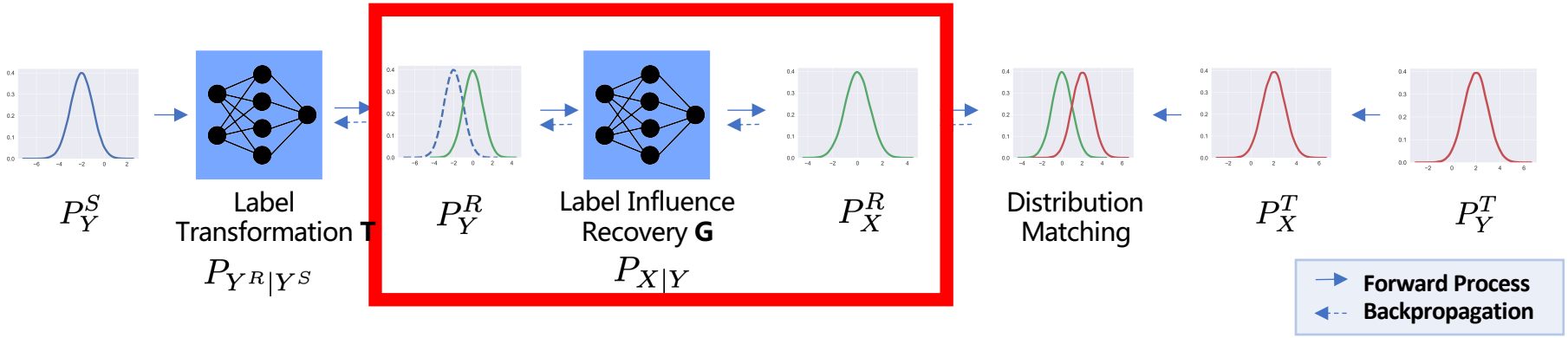
$$Y^R = T(Y^S, Z)$$

Details: Label Transformation



- Because P_X^R captures the influence of P_Y^R , we can possibly estimate P_Y^R (or **T**) by matching P_X^R and P_X^T .
- So we need to transform P_Y^R to a distribution P_X^R using our label influence recovery module **G**.

Details: Label Influence Recovery

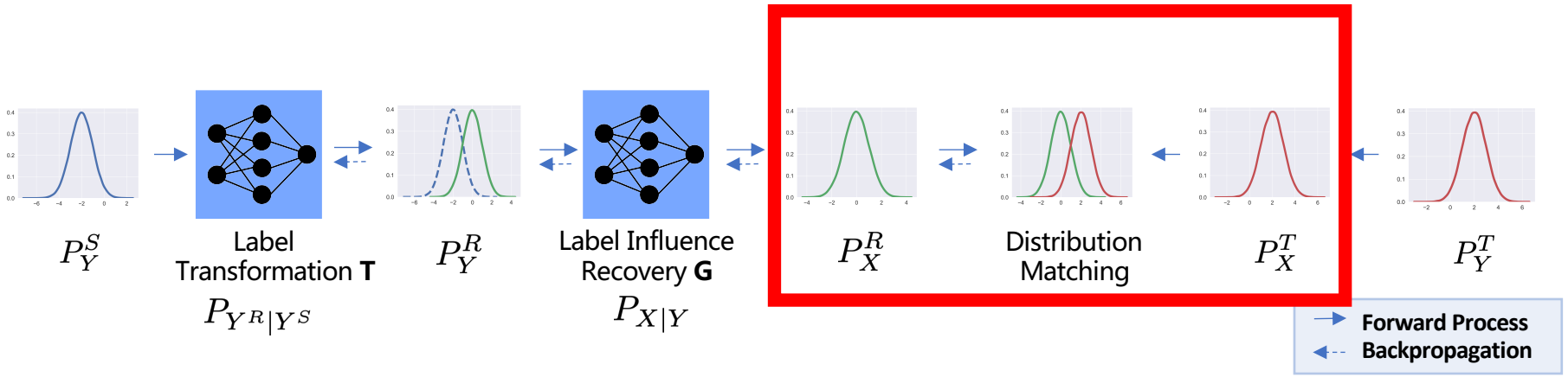


- The label influence recovery module **G** models the invariant conditional distribution $P_{X|Y}$

- Obviously, it can be learned by a conditional GAN. Here we use BigGAN[3] or TAC-GAN[4] to learn **G** using the training set.

$$X^R = G(Y^R, E)$$

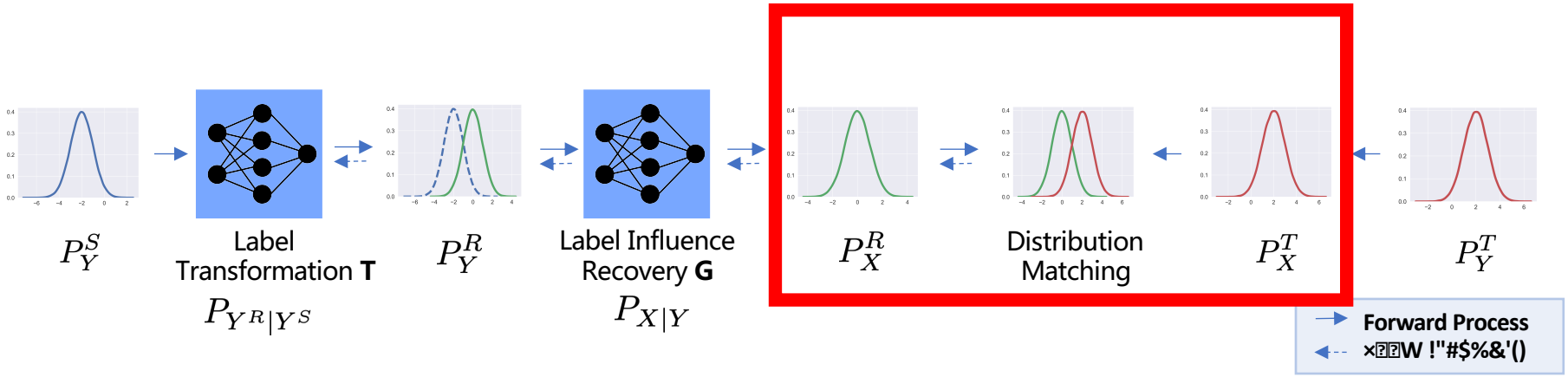
Details: Distribution Matching



- To make it compatible with large-scale data, we introduce a discriminator D_T and do adversarial training with **T**.

$$\min_{T} \max_{D_T} \quad \mathop{\mathbb{E}}_{X \sim P_X^T}[\mathcal{L}(D_T(X))] + \mathop{\mathbb{E}}_{Z \sim P_Z, E \sim P_E, Y^S \sim P_Y^S}[\mathcal{L}(1 - D_T(G(T(Y^S, Z), E)))]$$

Details: Feature Matching



- X (such as image) contains many redundant features X_R , causing unnecessary estimation errors of P_Y^T.

- So we propose a Proposition shows that the pre-trained feature extractor h in classifier or regressor can approximately satisfy $Y \perp\!\!\!\perp X|h(X)$. As such, we can match data on the feature space of h(x).

Shift Correction

- After matching the P_X^R and P_X^T, the Label Transformation Module **T** implicitly models the real target label distribution P_Y^T, and the transformed label y' from T obeys the estimated $\hat{P}_Y^T$.

Shift Correction

- After matching the P_X^R and P_X^T , the Label Transformation Module **T** implicitly models the real target label distribution P_Y^T , and the transformed label y' from T obeys the estimated $\hat{P}_S^T$.

- Instead of retraining a new classifier or regressor, the output layer of pre-trained classifier or regressor at the training set is the only module need to be adjusted given feature extractor h.

Shift Correction

- After matching the P_X^R and P_X^T , the Label Transformation Module **T** implicitly models the real target label distribution P_Y^T , and the transformed label y^T from T obeys the estimated $\hat{P}_Y^T$.
- Instead of retraining a new classifier or regressor, the output layer of pre-trained classifier or regressor at the training set is the only module need to be adjusted given feature extractor h.
- As such, we directly fine-tune the output layer using samples X^R generated from **T** and **G**. The classifier or regressor can be quickly adapted to the Test set.

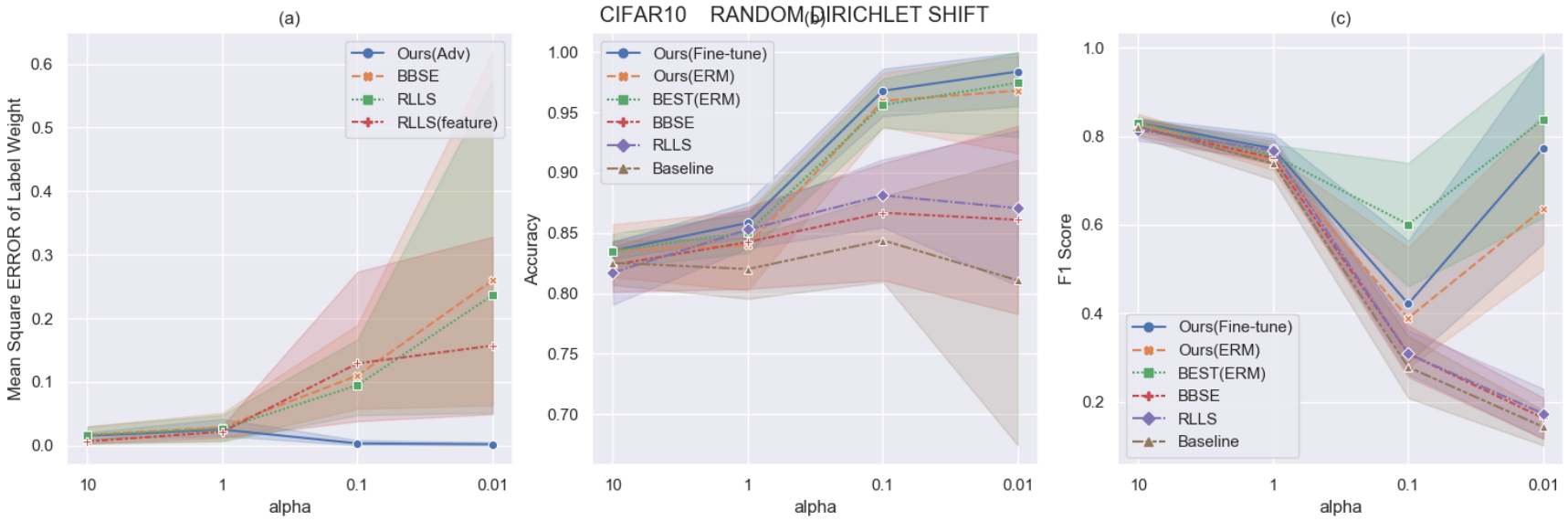
$$X^R = G(T(Y^S, Z), E)$$

Experiments

- Discrete Target Shift
  - Datasets: MNIST, FASHION-MNIST and CIFAR10 datasets.
  - Baselines: BBSE and RLLS
  - Shift settings:
    - Tweak-One Shift (Large label detection)
      - The ratio of one class is set to [0.5, 0.6, 0.7, 0.8, 0.9]., while ratios of other classes are uniform.
    - Minority-Class Shift
      - The ratio of [20%, 30%, 40%,50%] classes is set to 0.001, while the class priors of other classes are uniform.
    - Random Label Shift
      - Generating a label distribution by Dirichlet distribution with different values of the concentration parameter α [10, 1, 0.1, 0.01]. Note that a bigger α corresponding to a smoother label distribution.

Results on Cifar10

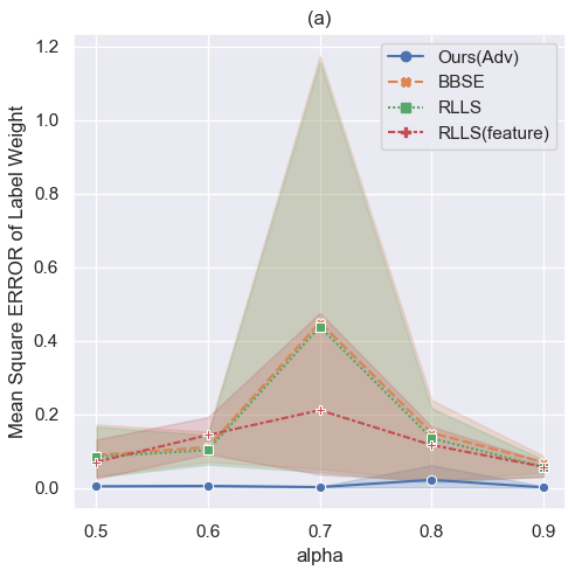
- Random Target Shift



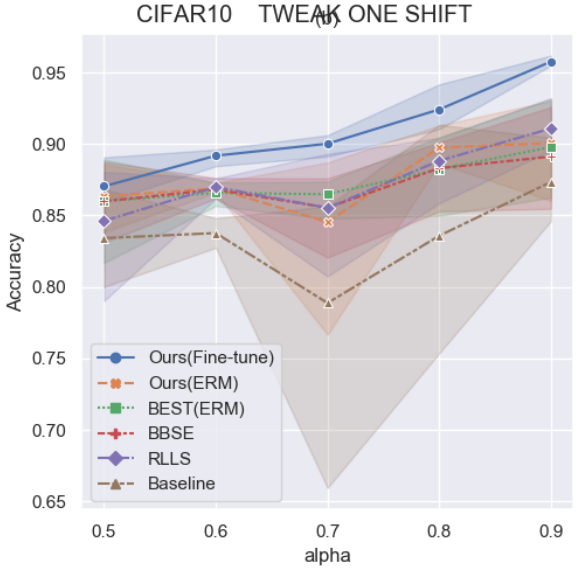
Estimation error | Accuracy | F1 score

Results on Cifar10

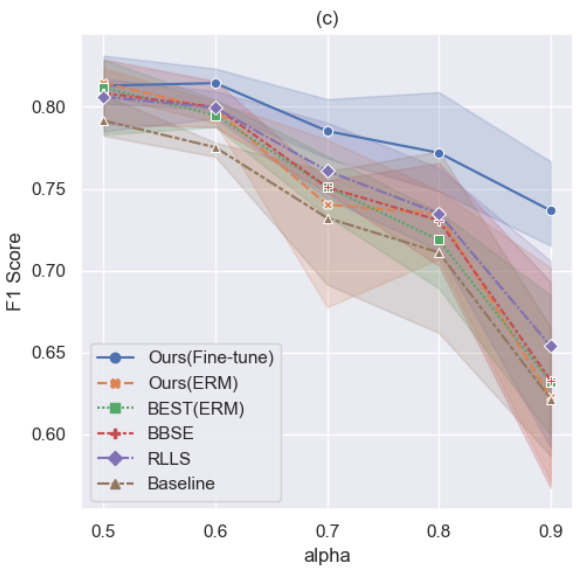
- Tweak-One Shift



Estimation error



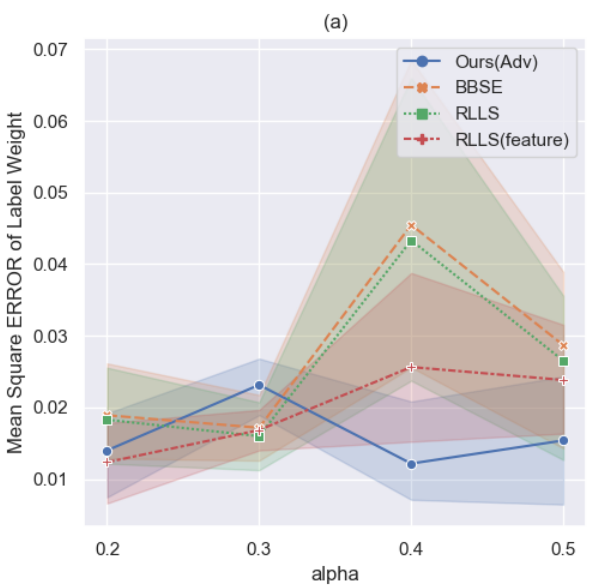
Accuracy



F1 score

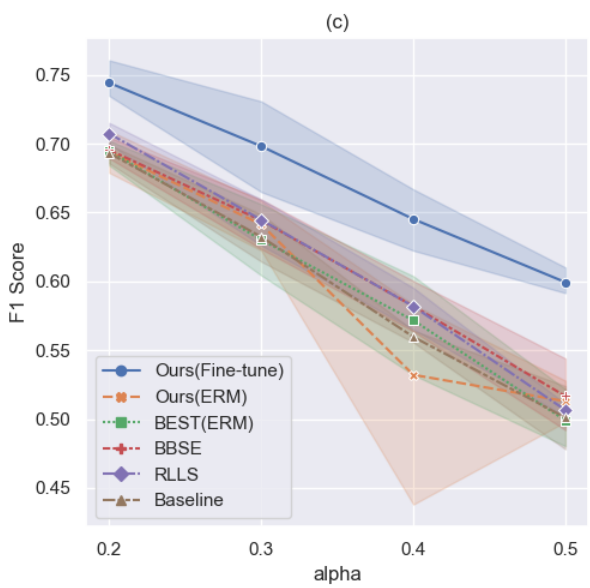
Results on Cifar10

- Minority-Class Shift



Estimation error

Accuracy

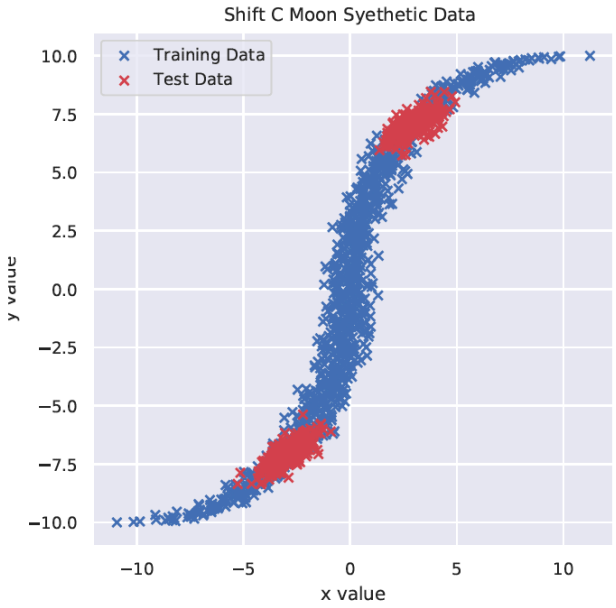


F1 score

Experiments

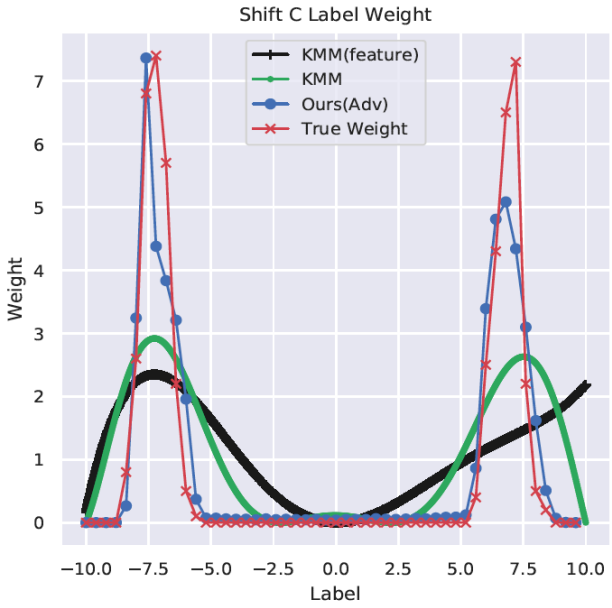
- Continuous Target Shift
  - Datasets: We modify the popular MOON dataset[1] and generates two square circles with radius R (10).
  - Baselines: KMM
  - Shift settings:
    - Shift A
      - The target label distribution is set to a Gaussian distribution with the mean $\frac{\sqrt{2}}{2} * R$ and variance 1.
    - Shift B
      - The target label distribution is set to a Gaussian distribution with the mean $-\frac{\sqrt{2}}{2} * R$ and variance 1.
    - Shift C
      - A mixture Gaussian distribution with Shift A and Shift B.
    - Shift D
      - The target label distribution is a random label distribution generated by a random parameterized neural network.

Illustration



(a)

Data Visualization of Shift C



(b)

Label Weights of models

Results of Continuous label experiments

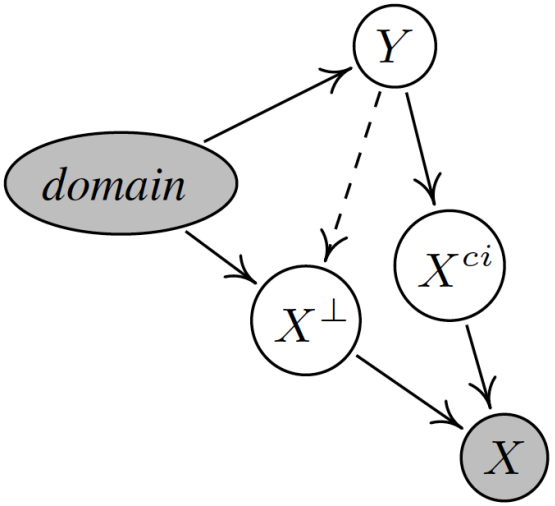
	SHIFT A	SHIFT B	SHIFT C	SHIFT D
Baseline	0.0061 $\pm$ 0.0012	0.0059 $\pm$ 0.0008	0.0055 $\pm$ 0.0009	0.034 $\pm$ 0.0114
KMM	0.0048 $\pm$ 0.0011	0.0044 $\pm$ 0.0005	0.0044 $\pm$ 0.0004	0.0275 $\pm$ 0.0096
KMM (feature)	0.0045 $\pm$ 0.0007	0.0039 $\pm$ 0.0006	0.0043 $\pm$ 0.0004	0.0276 $\pm$ 0.0097
Ours	**0.0036** $\pm$ **0.0002**	**0.0024** $\pm$ **9e-5**	**0.0036** $\pm$ **0.0004**	**0.0251** $\pm$ **0.0121**

Table 1. The results of Continuous Label Shift Synthetic Data Experiments. The value is the mean square error of prediction value and groun truth. The baseline is the original regressor trained on the standard training set, and the KMM is (Zhang et al., 2013)

Outline

- Background
- Causal Understanding of DA
- Target Shift correction
- Conditional Invariant Components
- Domain Adaptation as Inference on Graphical Models
- Causal Discovery from Multiple Domains
- Conclusions

Conditional Invariant Components

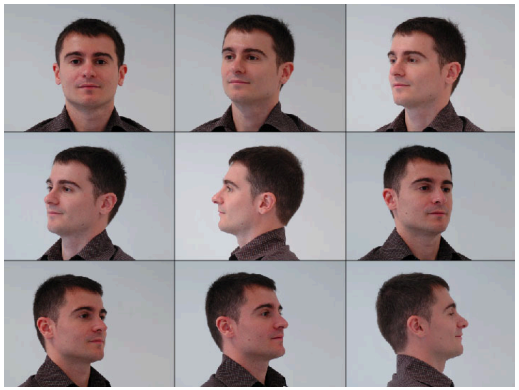


- Conditional invariant components (CIC)

$$X^{ci}: \ \mathbb{P}^{S}_{X^{ci}|Y} = \mathbb{P}^{T}_{X^{ci}|Y}$$

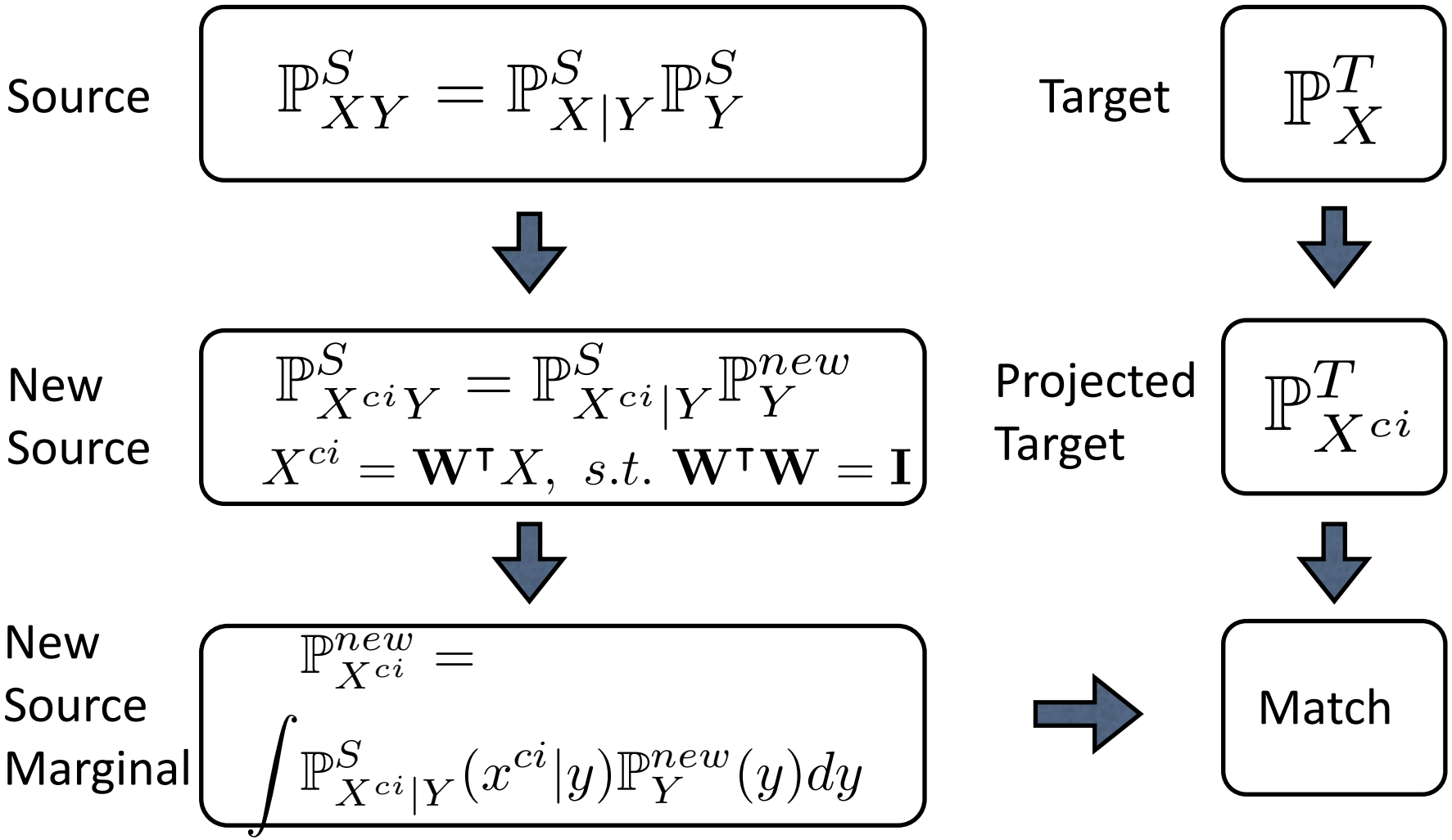
- CIC can be found by a linear transformation

$$X^{ci} = \mathbf{W}^{\intercal} X, \ s.t. \ \mathbf{W}^{\intercal}\mathbf{W} = \mathbf{I}$$



Gong, Zhang, et al. Domain adaptation with conditional transferable components. ICML16

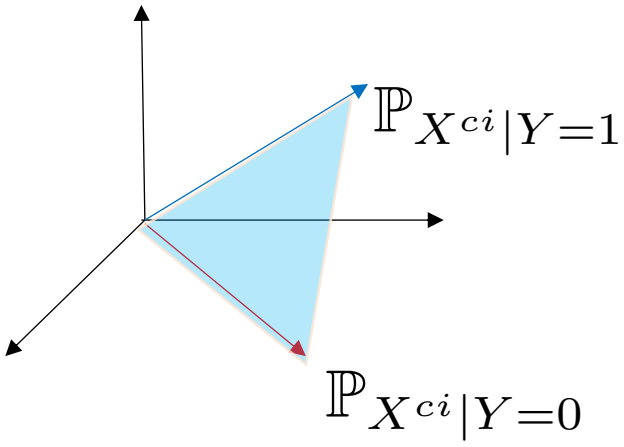
Conditional Invariant Components (CIC)



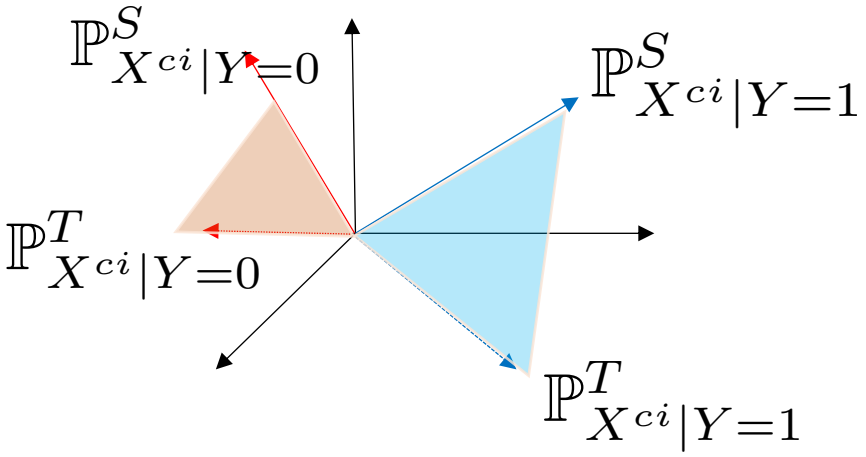
Optimize: $\mathbf{W}$, $\mathbb{P}^{new}_Y$

Identifiability

If $\mathbb{P}^{new}_{X^{ci}} = \mathbb{P}^{te}_{X^{ci}}$, under what conditions can we have $\mathbb{P}^{S}_{X^{ci}|Y} = \mathbb{P}^{T}_{X^{ci}|Y}$ and $\mathbb{P}^{new}_{Y} = \mathbb{P}^{T}_{Y}$?



Non-trivial



Low dimensional change

Relation to IC

$$\exists X' = h(X) \Rightarrow \begin{array}{l} \mathbb{P}^S(X') \approx \mathbb{P}^T(X') \\ \mathbb{P}^S(Y|X') \approx \mathbb{P}^T(Y|X') \end{array}$$

- IC methods fail to find CIC if $\mathbb{P}^S(y) = \mathbb{P}^T(y)$.

$$\mathbb{P}^S(X') = \int \mathbb{P}^S(X'|y)\mathbb{P}^S(y)dxdy$$

$$\mathbb{P}^T(X') = \int \mathbb{P}^T(X'|y)\mathbb{P}^T(y)dxdy$$

- IC methods can possibly find CIC if $\mathbb{P}^S(y) \neq \mathbb{P}^T(y)$.

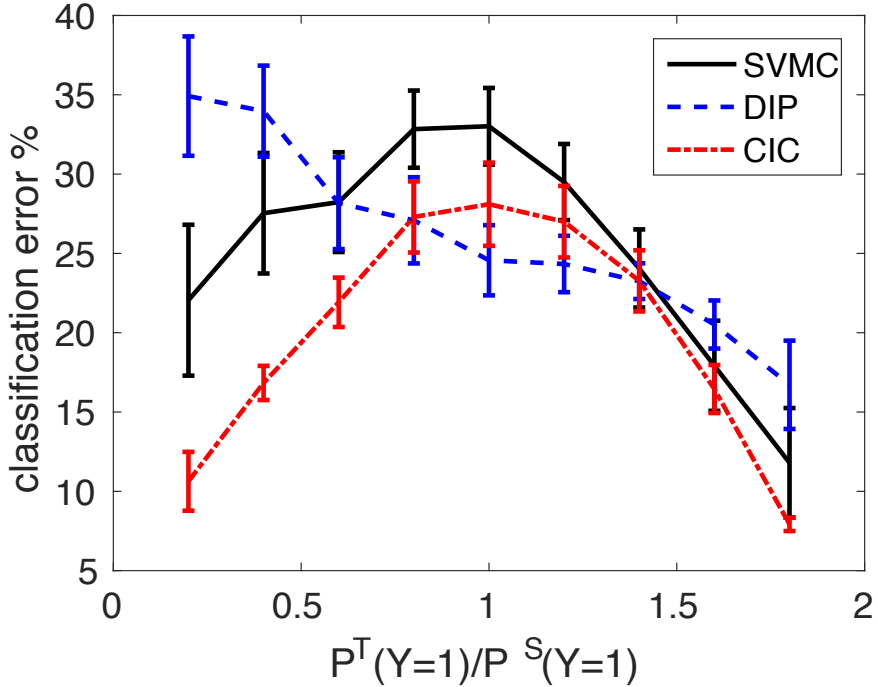
- Finding CIC needs additional constraints.

Simulation

- Binary classification training and test data from a 10-dimensional mixture of Gaussians:

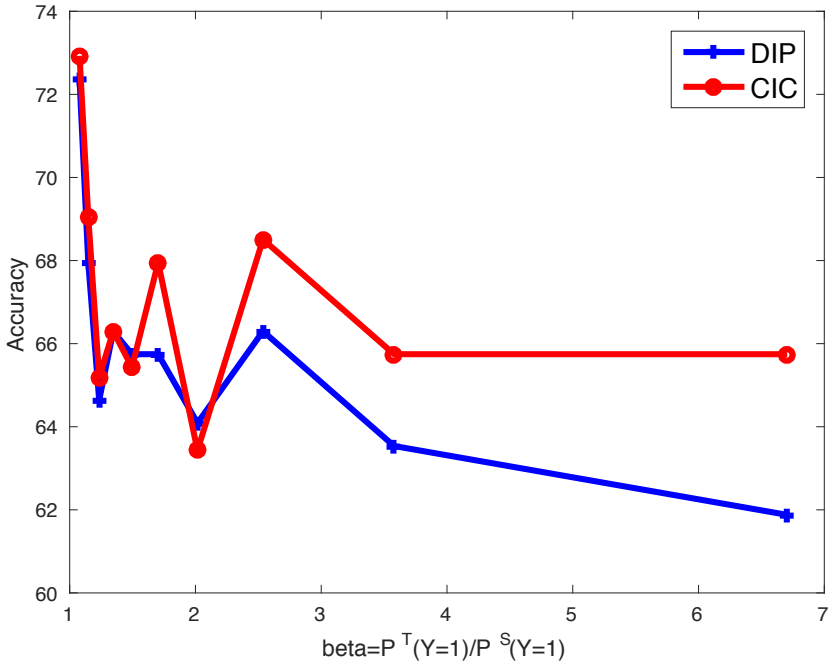
$$x \sim \sum_{i=1}^{2} \pi_i \mathcal{N}(\theta_i, \Sigma_i), \theta_{ij} \sim \mathcal{U}(-0.25, 0.25), \Sigma_i \sim 0.01 * \mathcal{W}(2 * I_{10}, 7)$$

- We compare the methods' sensitiveness to changes in $P(Y)$ and the number of conditional invariant components.



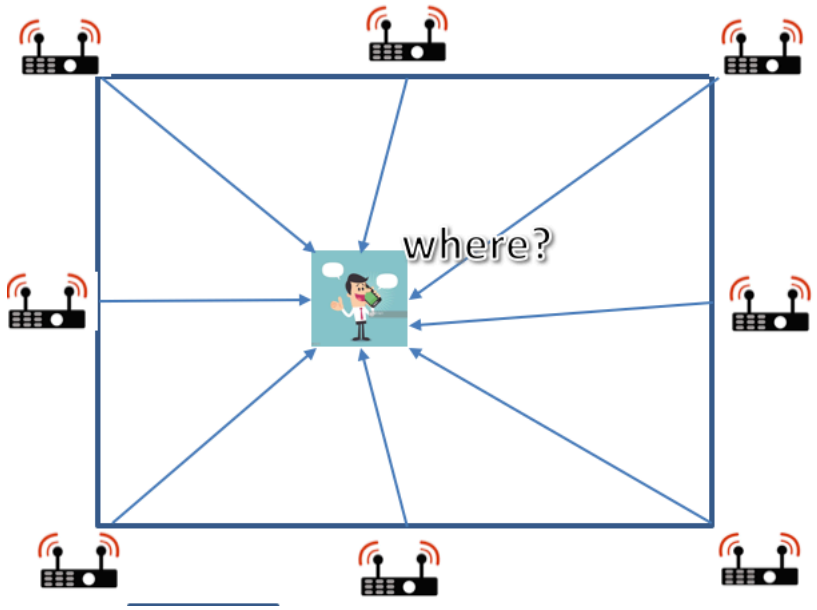
Visual Recognition

	SVM	GFK	TCA	LM	GeTarS	DIP	DIP-CC	CTC	CTC-TIP
A→C	41.7	42.2	35.0	45.5	44.9	47.4	47.2	48.6	**48.8**
A→D	41.4	42.7	36.3	47.1	45.9	50.3	49.0	**52.9**	52.2
A→W	34.2	40.7	27.8	46.1	39.7	47.5	47.8	**49.8**	49.1
C→A	51.8	44.5	41.4	56.7	56.9	55.7	**58.7**	58.1	57.9
C→D	54.1	43.3	45.2	57.3	49.0	60.5	**61.2**	59.2	58.5
C→W	46.8	44.7	32.5	49.5	46.4	58.3	58.0	**58.6**	57.8
W→A	31.1	31.8	24.2	40.2	38.4	42.6	40.9	**43.2**	43.1
W→C	31.5	30.8	22.5	35.4	34.3	34.2	37.2	38.3	**38.8**
W→D	70.7	75.6	80.2	75.2	86.0	88.5	91.7	**94.3**	93.6



Robust to changes in P_Y

WiFi Localization



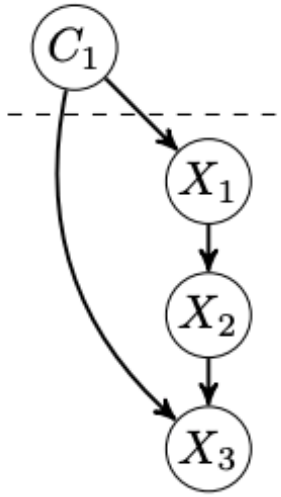
WiFi localization: localize mobile devices from the WiFi signals.

Two transfer tasks:

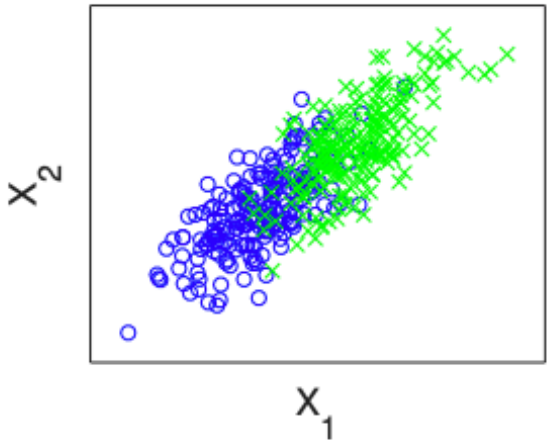
- Transfer between different time periods
- Transfer between different devices

	KRR	TCA	SuK|	DIP	DIP-CC	GeTarS	CTC	CTC-TIP
t1 → t2	80.84 ± 1.14	86.85 ± 1.1	90.36 ± 1.22	87.98 ± 2.33	$\mathbf{91.30 \pm 3.24}$	86.76 ± 1.91	89.36 ± 1.78	89.22 ± 1.66
t1 → t3	76.44 ± 2.66	80.48 ± 2.73	$\mathbf{94.97 \pm 1.29}$	84.20 ± 4.29	84.32 ± 4.57	90.62 ± 2.25	94.80 ± 0.87	92.60 ± 4.50
t2 → t3	67.12 ± 1.28	72.02 ± 1.32	85.83 ± 1.31	80.58 ± 2.10	81.22 ± 4.31	82.68 ± 3.71	87.92 ± 1.87	$\mathbf{89.52 \pm 1.14}$
hallway1	60.02 ± 2.60	65.93 ± 0.86	76.36 ± 2.44	77.48 ± 2.68	76.24 ± 5.14	84.38 ± 1.98	$\mathbf{86.98 \pm 2.02}$	86.78 ± 2.31
hallway2	49.38 ± 2.30	62.44 ± 1.25	64.69 ± 0.77	78.54 ± 1.66	77.8 ± 2.70	77.38 ± 2.09	87.74 ± 1.89	$\mathbf{87.94 \pm 2.07}$
hallway3	48.42 ± 1.32	59.18 ± 0.56	65.73 ± 1.57	75.10 ± 3.39	73.40 ± 4.06	80.64 ± 1.76	$\mathbf{82.02 \pm 2.34}$	81.72 ± 2.25

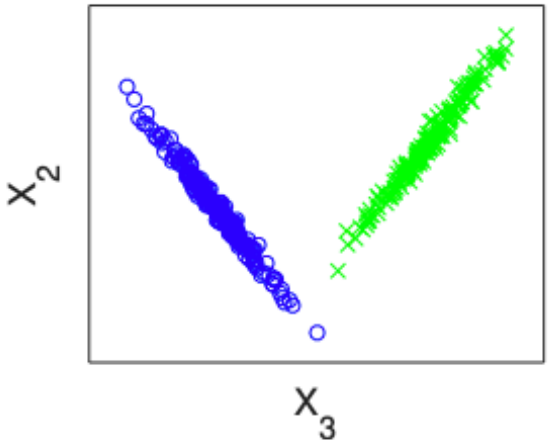
Conditional Invariance from a Learned Causal Model



(a) Causal graph



(b) No distribution shift for $\{X_1\}$: $\mathbb{P}(Y \mid X_1, C_1 = 0) = \mathbb{P}(Y \mid X_1, C_1 = 1)$



(c) Strong distribution shift for $\{X_3\}$: $\mathbb{P}(Y \mid X_3, C_1 = 0) \neq \mathbb{P}(Y \mid X_3, C_1 = 1)$

Magliacane, Sara, et al. "Domain adaptation by using causal inference to predict invariant conditional distributions." *Advances in Neural Information Processing Systems*. 2018.

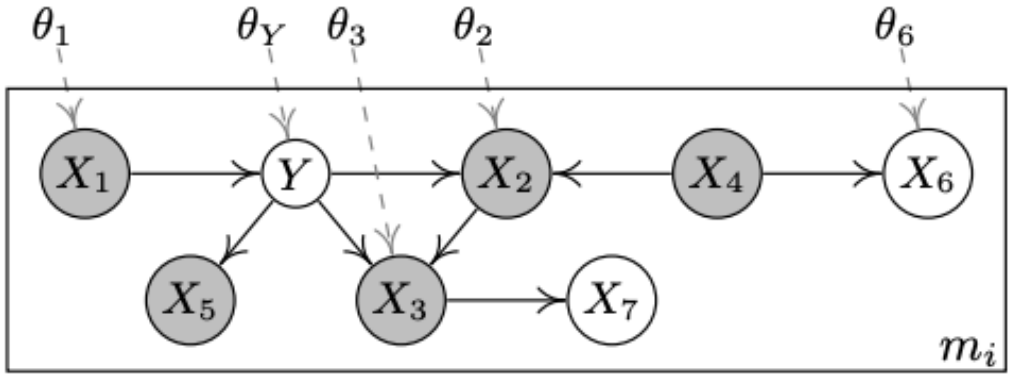
Outline

- Background
- Causal Understanding of DA
- Target Shift correction
- Conditional Invariant Components
- Domain Adaptation as Inference on Graphical Models
- Causal Discovery from Multiple Domains
- Conclusions

Problems of Existing Methods

- The causal graph and the invariance/changing causal modules are assumed to be known
- The algorithms do not make full use of the causal generative process
- Learning causal graphs from observational data is a hard.

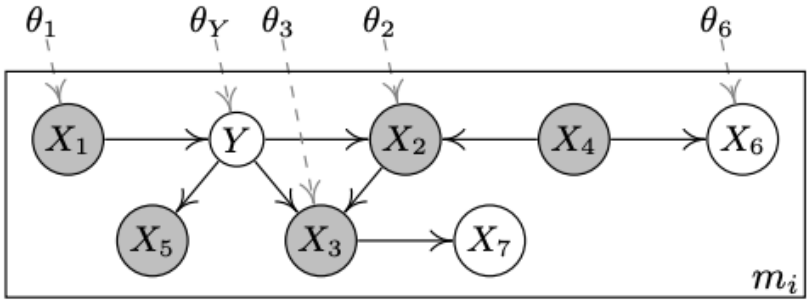
Inference on Graphical Models



- Automated way to model change and invariance properties in the joint distribution
  - Factorize the joint distribution according to an augmented directed acyclic graph (DAG)
  - Formulate domain adaptation as a Bayesian inference problem on the augmented DAG

Zhang, K*., Gong, M.*, et al. (2020). Domain Adaptation As a Problem of Inference on Graphical Models. *NeurIPS 2020.*

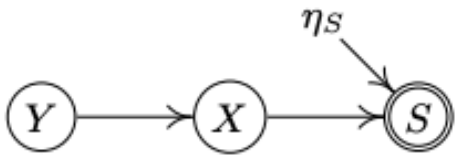
Augmented DAG



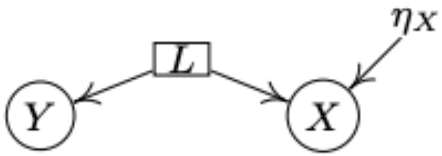
- DAG encodes conditional independence relations
- Encode distribution change by augmenting DAG with $\boldsymbol{\theta}$
  - θ_i are independent – independent change
  - θ follows a prior distribution $P(\boldsymbol{\theta})$
- Data generating process
  - Generate $\boldsymbol{\theta}^{(i)}$ from $P(\boldsymbol{\theta})$
  - Given $\boldsymbol{\theta}^{(i)}$, sample data from the distribution in the i-th domain:

$$P(\mathbf{X}, Y|\boldsymbol{\theta}^{(i)}) = P(X_1|\theta_1^{(i)})P(Y|X_1, \theta_Y^{(i)})P(X_5|Y)P(X_2|Y, X_4, \theta_2^{(i)})P(X_3|Y, X_2, \theta_3^{(i)})\times$$
$$P(X_4)P(X_6|X_4, \theta_6^{(i)})P(X_7|X_3).$$

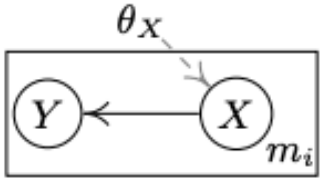
Relation to Causal Graphs



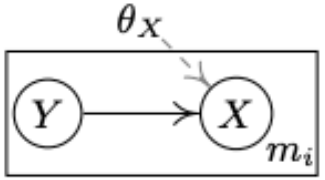
(a) The underlying data generating process of Example 1. Y generates (causes) X, and S denotes the selection variable (a data point is included if and only if $S = 1$).



(c) The generating process of Example 2. L is a confounder; the mechanism of X changes across domains, as indicated by η_X.

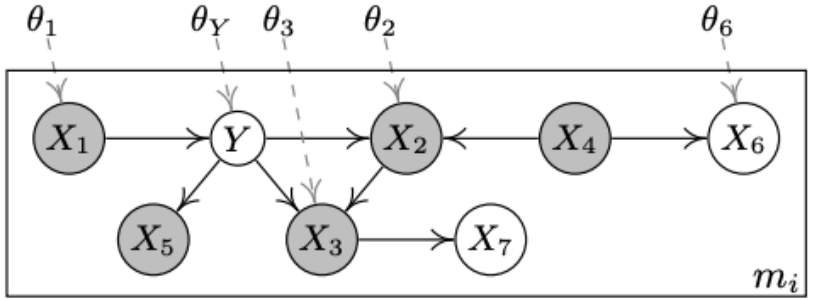


(b) The augmented DAG representation for Example 1 to explain how the data distribution changes across domains.



(d) The augmented DAG representation for Example 2 to explain how the data distribution changes across domains.

Bayesian Inference

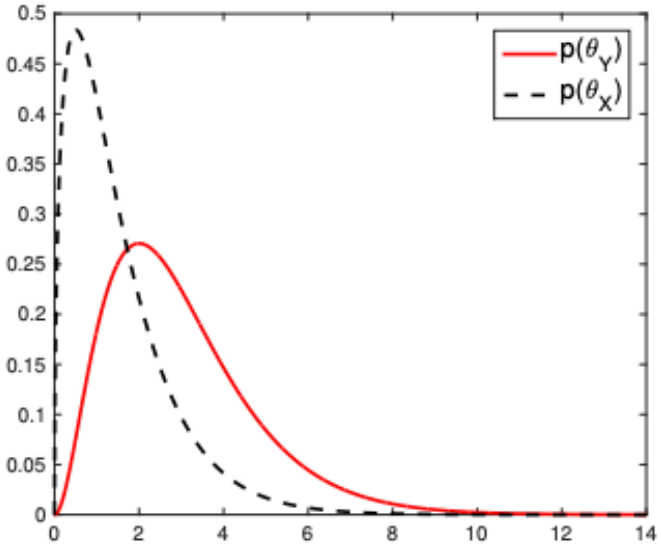


$$P(y_k^\tau \mid \mathbf{x}^\tau) = \int \underbrace{P(y_k^\tau \mid \mathbf{x}_k^\tau, \boldsymbol{\theta})}_{\text{Classifier}} \underbrace{\frac{\prod_k \left[\sum_{y_k^\tau} \prod_{V_j \in \mathbf{V}} \mathcal{C}_{jk}\right] \prod_{V_j \in \mathbf{V}} P(\theta_{V_j})}{\int \prod_k \left[\sum_{y_k^\tau} \prod_{V_j \in \mathbf{V}} \mathcal{C}_{jk}\right] \prod_{V_j \in \mathbf{V}} P(\theta_{V_j}) d\theta_{V_j}}}_{P(\boldsymbol{\theta} \mid \mathbf{x}^\tau)} d\boldsymbol{\theta}.$$

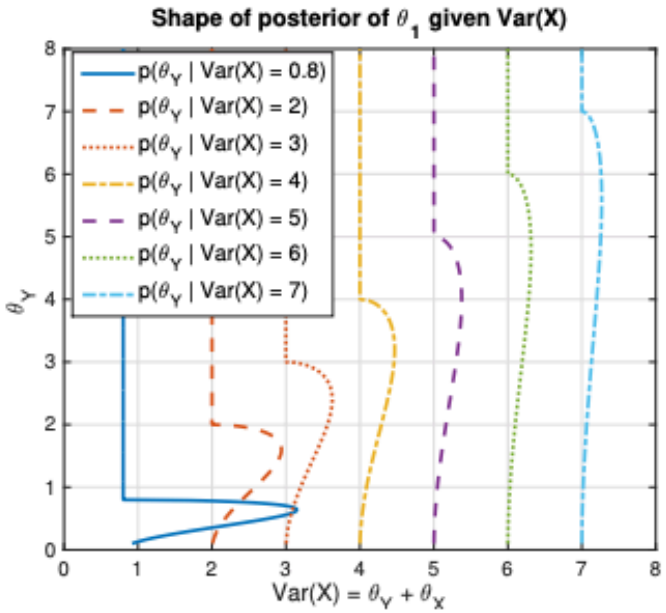
$$\mathcal{C}_{jk} := P(v_{jk}^\tau \mid \mathbb{PA}(v_{jk}^\tau), \theta_{V_j})$$

$$\mathbf{V} = \mathbb{CH}(Y) \cup \{Y\}$$

Benefits of Bayesian Treatment



(a) Prior distributions of $\boldsymbol{\theta}$



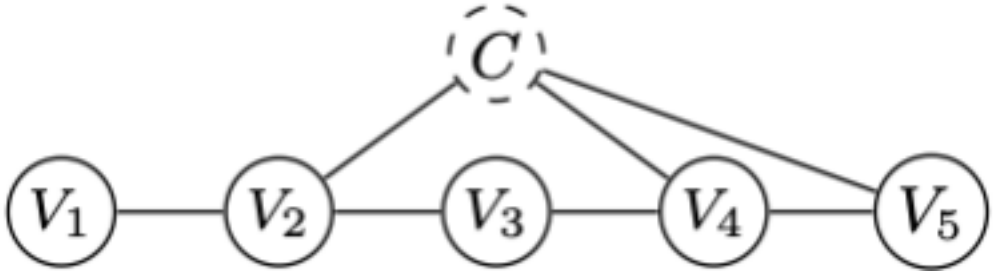
(b) Posterior of θ_Y given $\mathbb{V}\mathrm{ar}(X)$

$$Y \sim N(0, \theta_Y) \qquad X = Y + E$$

$$E \sim N(0, \theta_X) \qquad \mathrm{X} \sim N(0, \theta_X + \theta_Y)$$

Graph Learning: Skeleton learning and changing module detection

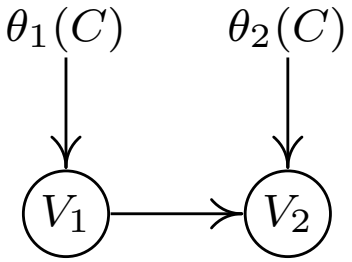
- Using Domain Index C as a surrogate variable and apply Constraint-based search on C and the observed features and labels.
  - Detecting Changing Causal Modules
  - Obtain the Skeleton of the graph



Huang, B., Zhang, K., Zhang, J., Ramsey, J., Sanchez-Romero, R., Glymour, C., & Schölkopf, B. (2020). Causal discovery from heterogeneous/nonstationary data. *Journal of Machine Learning Research*, *21*(89), 1-53.

Graph Learning: Determine edge direction

- Independent changes in P(cause) and P(effect|cause)



Special cases: if $C - V_k - V_l$, since $C \to V_k$, we known

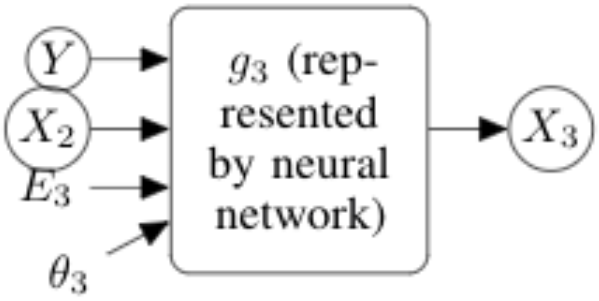
- $C \to V_k \leftarrow V_l$, if $C \perp\!\!\!\perp V_l$ given a variable set **excluding** V_k
- $C \to V_k \to V_l$, if $C \perp\!\!\!\perp V_l$ given a variable set **including** V_k

Invariant cause

Invariant mechanism

Approximate Inference

Latent variable conditional GAN



Approximate inference

$$\log p(\mathcal{D}) \geq -\sum_{i=1}^{s} \mathtt{KL}(q(\boldsymbol{\theta}|\mathcal{D}^i)|p(\boldsymbol{\theta})) + \mathbb{E}_{q(\boldsymbol{\theta}|\mathcal{D}^i)}\Big[\sum_{k=1}^{m_i} \log p_g(\mathbf{x}_k^{(i)}, y_k^{(i)}|\boldsymbol{\theta})\Big]$$

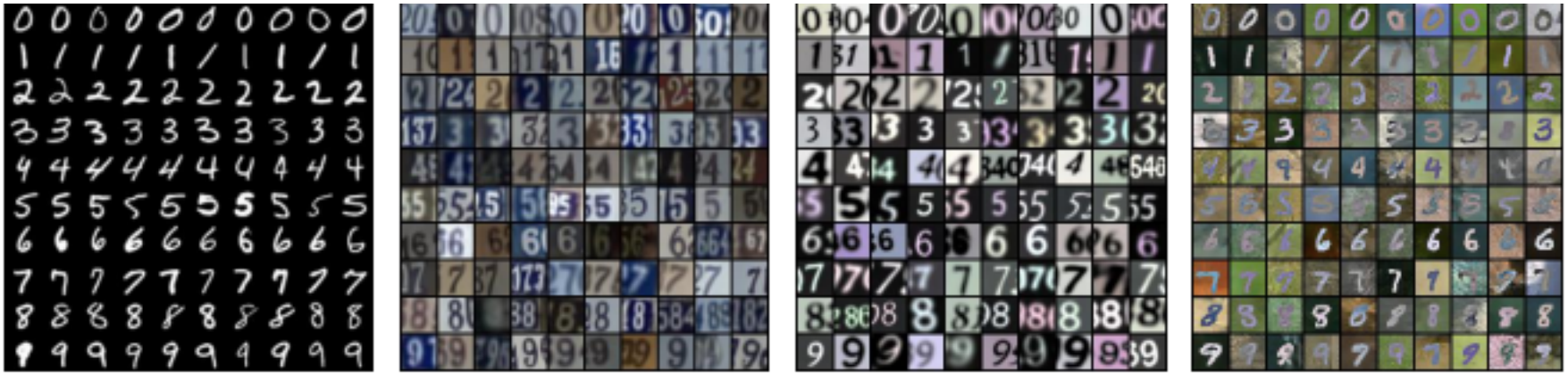
$$- \mathtt{KL}(q(\boldsymbol{\theta}|\mathcal{D}^\tau)|p(\boldsymbol{\theta})) + \mathbb{E}_{q(\boldsymbol{\theta}|\mathcal{D}^\tau)}\Big[\sum_{k=1}^{m} \log p_g(\mathbf{x}_k^{\tau}|\boldsymbol{\theta})\Big].$$

$$q(\boldsymbol{\theta}|\mathcal{D}^i) = \mathcal{N}(\boldsymbol{\theta}|\mu^{(i)}, \sigma^{(i)}), q(\boldsymbol{\theta}|\mathcal{D}^\tau) = \mathcal{N}(\boldsymbol{\theta}|\mu^{\tau}, \sigma^{\tau})$$

Digits Adaptation

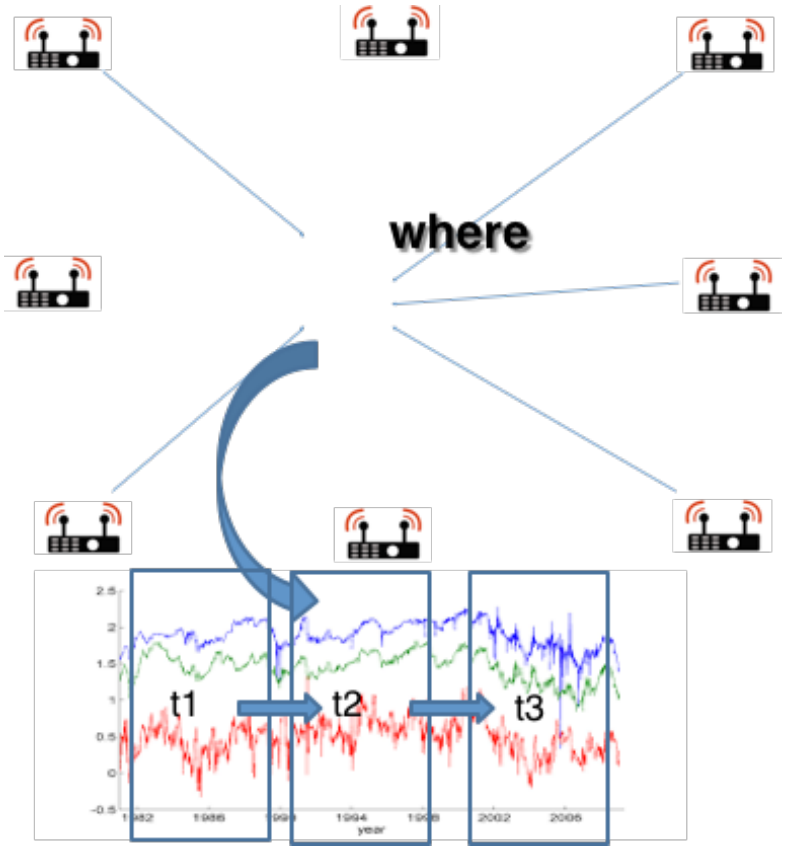
Table 3: Accuracy on the digits data. T: MNIST; M: MNIST-M; S: SVHN; D: SynthDigits.

	weigh	poolNN	poolDANN	Hard-Max	Soft-Max	poolNN_Ours	Infer
$S+M+D/T$	75.5	93.8	92.5	97.6	**97.9**	94.9	96.64
$T+S+D/M$	56.3	56.1	65.1	66.3	68.7	59.6	**89.89**
$M+T+D/S$	60.4	77.1	77.6	80.2	81.6	67.8	**89.34**



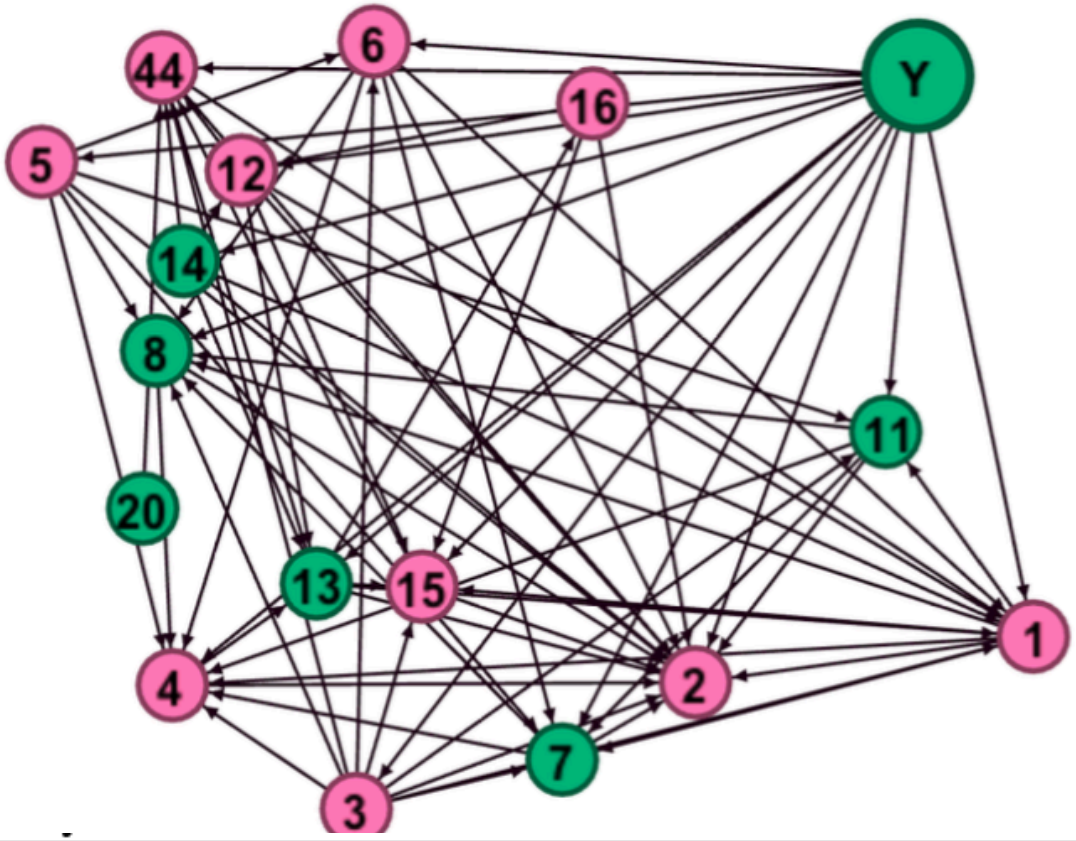
Gong, M., Xu, Y., Li, C., Zhang, K., & Batmanghelich, K. (2019). Twin auxilary classifiers GAN. In *Advances in neural information processing systems* (pp. 1330-1339).

WiFi Localization



- Localize mobile devices from the WiFi signals.
- Transfer between different time periods

WiFi Localization



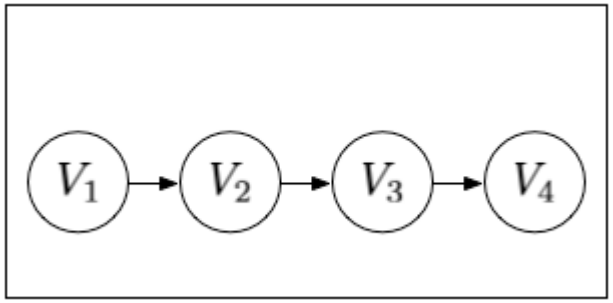
	DICA	weigh	LMP	poolSVM	Soft-Max	poolNN	Infer
t2, t3 → t1	29.32(2.5)	43.71(3.02)	46.80(1.4)	40.25(1.6)	44.86(5.1)	42.88(1.6)	**70.8(2.7)**
t1, t3 → t2	24.5(3.6)	38.19(1.9)	39.11(2.1)	48.70(1.8)	44.95(4.4)	47.41(2.1)	**84.5(2.9)**
t1, t2 → t3	21.7(3.9)	36.03(1.85)	39.28(2.05)	40.46(1.4)	43.63(4.1)	41.00(1.8)	**83.0(7.3)**

Outline

- Background
- Causal Understanding of DA
- Target Shift correction
- Conditional Invariant Components
- Domain Adaptation as Inference on Graphical Models
- Causal Discovery from Multiple Domains
- Conclusions

Background

- How do we represent causal relations?
  - Acyclicity assumption
  - Noises are mutually independent
  - i.i.d. samples



Graphical Causal Models (GCM)

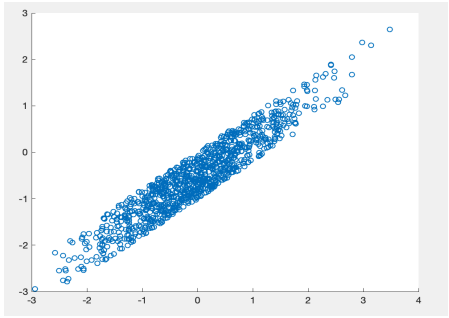
- Causal module
  - Each causal module $P(V_i \mid PA^i)$
  - Joint distribution $P(\mathbf{V}) = \prod_{i=1}^{n} P(V_i \mid PA^i)$

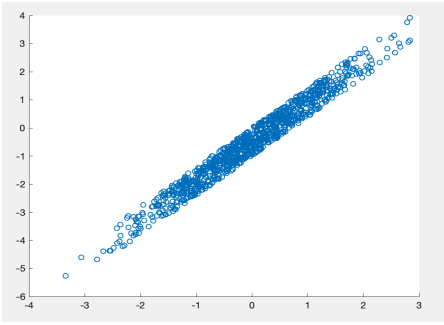
$$V_1 = E_1$$
$$V_2 = f_2(V_1) + E_2$$
$$V_3 = f_3(V_2) + E_3$$
$$V_4 = f_4(V_3) + E_4$$

Structure Equation Models (SEM)

Background

- Distribution shifts
  - ➢ Change of causal strength
  - ➢ Change of causal functions
  - ➢ Change of noise variance





Domain 1:
$X_1 = E_1$
$X_2 = 0.8 * X_1 + E_2$

Domain 2:
$X_1 = E_1$
$X_2 = 1.4 * X_1 + E_2$

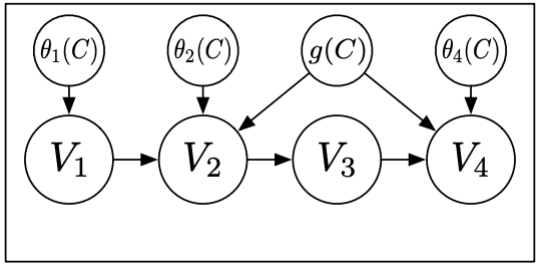
Causal module $P(X_2|X_1)$ changes
remains the same

$$P_1(X_1, Y_1) \neq P_2(X_2, Y_2)$$

Samples are not i.i.d.

Background

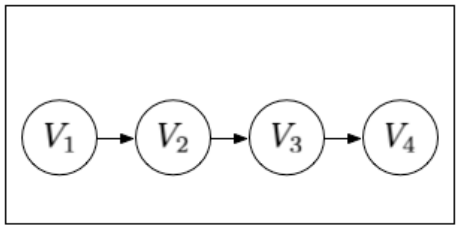
- Distribution shifts
  - Change of causal strength
  - Change of causal functions
  - Change of noise variance



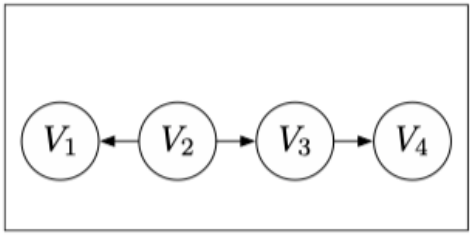
- Representation of distribution shifts
  - $V_i = f_i\left(PA^i, \mathbf{g}^i(C), \theta_i(C), \epsilon_i\right)$
  - $\theta(C)$ are the effective parameters in V_i's causal module that are mutually independent for all variables
  - $g(C)$ can be regarded as confounders

Causal Discovery for Single-domain Data

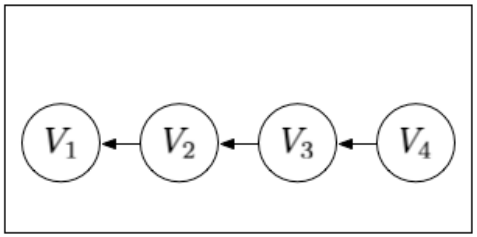
- Assumptions: samples come from the same domain (i.i.d.)
  - Constraint-based algorithms: PC
  - Score-based algorithms: GES
  - Structure equation models: LiNGAM
- Drawbacks
  - Equivalence class (i.e. a set of causal graphs containing the same conditional independence relations)



Ground truth



Estimation 1

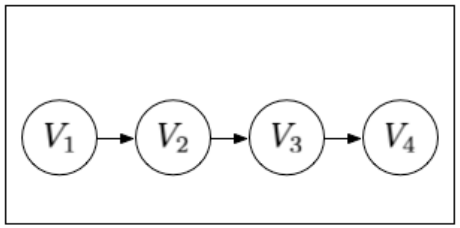


Estimation 2

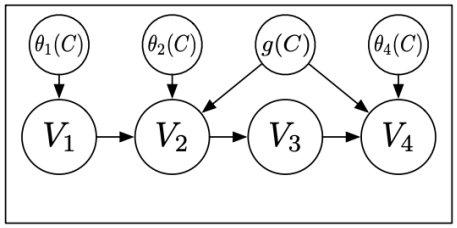
They belong to the same equivalence class!

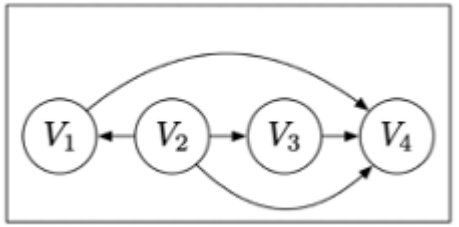
Causal Discovery for Single-domain Data

- Assumptions: samples come from the same domain (i.i.d.)
  - Constraint-based algorithms: PC
  - Score-based algorithms: GES
  - Structure equation models: LiNGAM
- Drawbacks
  - Cannot be used for multiple-domain data



Ground truth



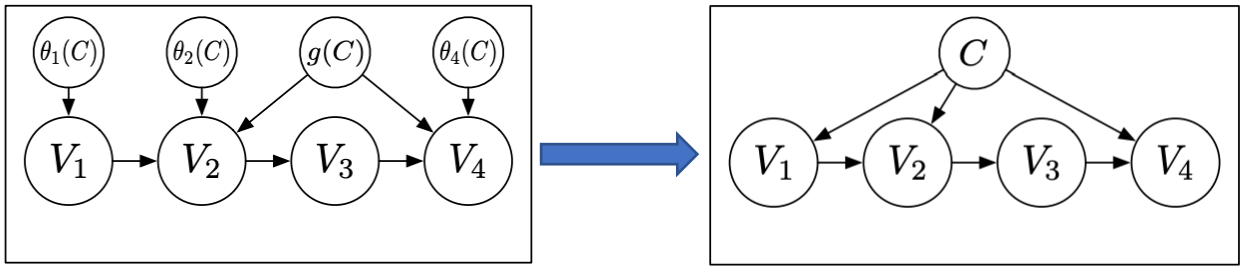


Estimation

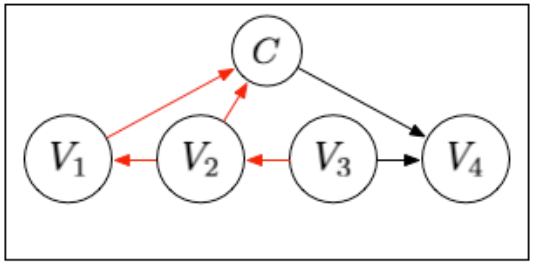
They may produce erroneous edges!

Causal Directions Identification by Distribution Shifts

- Augmented graphs
  - $\theta(C)$ and $\mathrm{g}(C)$ are not available. C is available as domain index.
  - Use C as a "surrogate" variable!
  - Samples from $P(V)$ are not i.i.d., but samples from $P(V, C)$ are i.i.d.

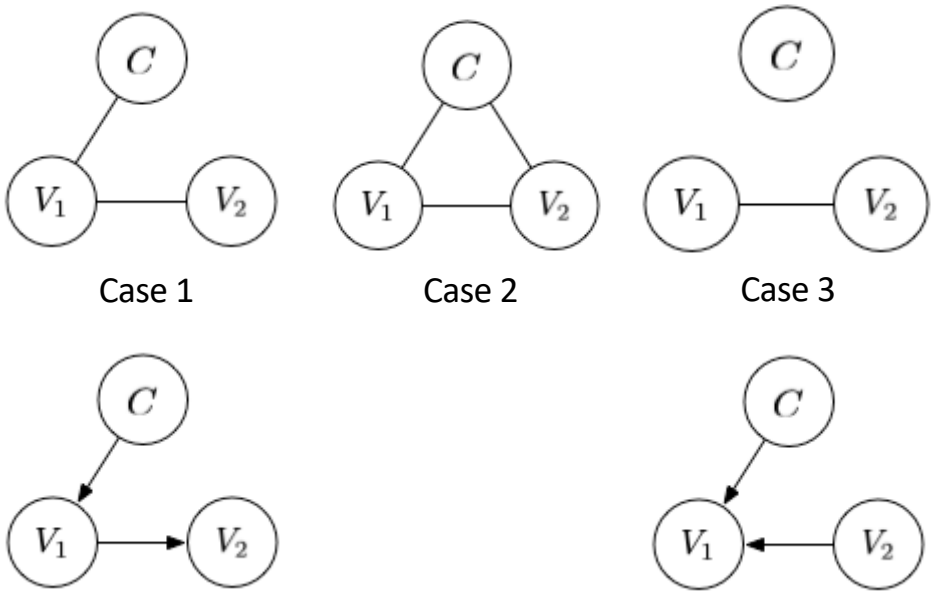


- Equivalence class
  - The directions of some edges still cannot be identified!



Causal Directions Identification by Distribution Shifts

- When can we identify more directions?
  - Case 1: prior knowledge $C \longrightarrow V_1$

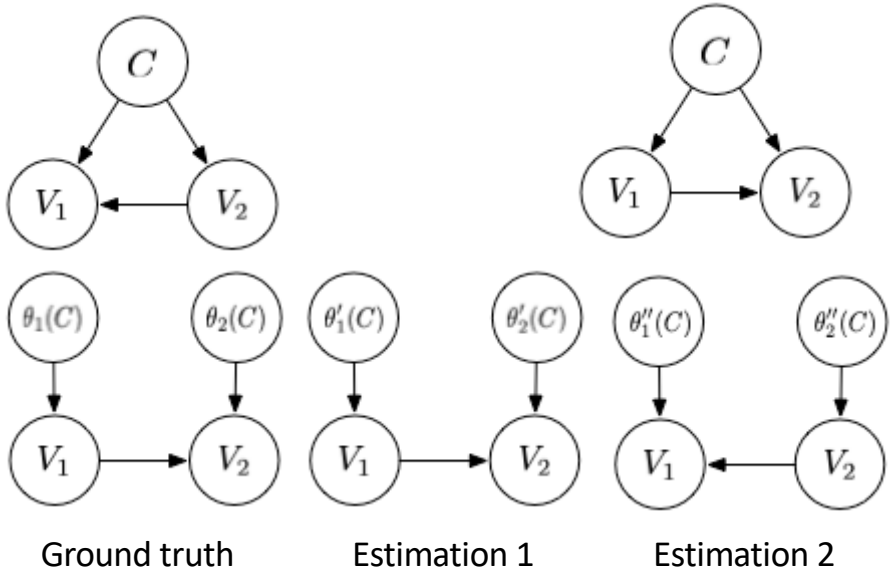


They contain different conditional independence relations!

Causal Directions Identification by Distribution Shifts

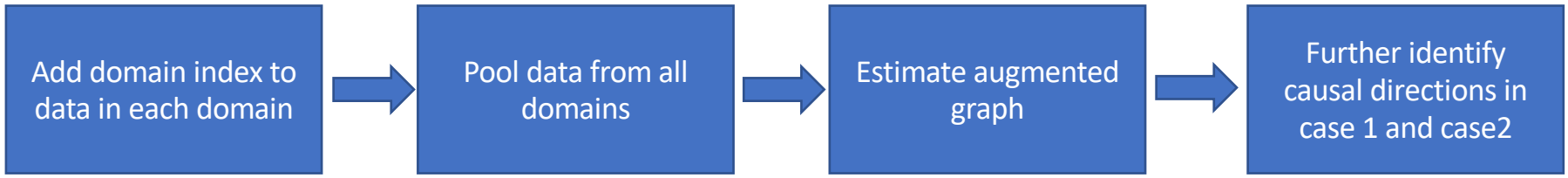
- When can we identify more directions?
  - ➢ Case 2: dependence between $\theta(C)$

Choose the direction with smaller dependence between $\theta_1(C)$ and $\theta_2(C)$!



Ground truth Estimation 1 Estimation 2

Causal Directions Identification by Distribution Shifts



Conclusion

- Causal Generative Process (CGP) provides a compact description of distribution change properties.
- When Y->X, DA can be performed with single source domain even when P(Y|X) changes.
- DA can be casted as a Bayesian inference problem on graphical models, which can be automatically learned from data.