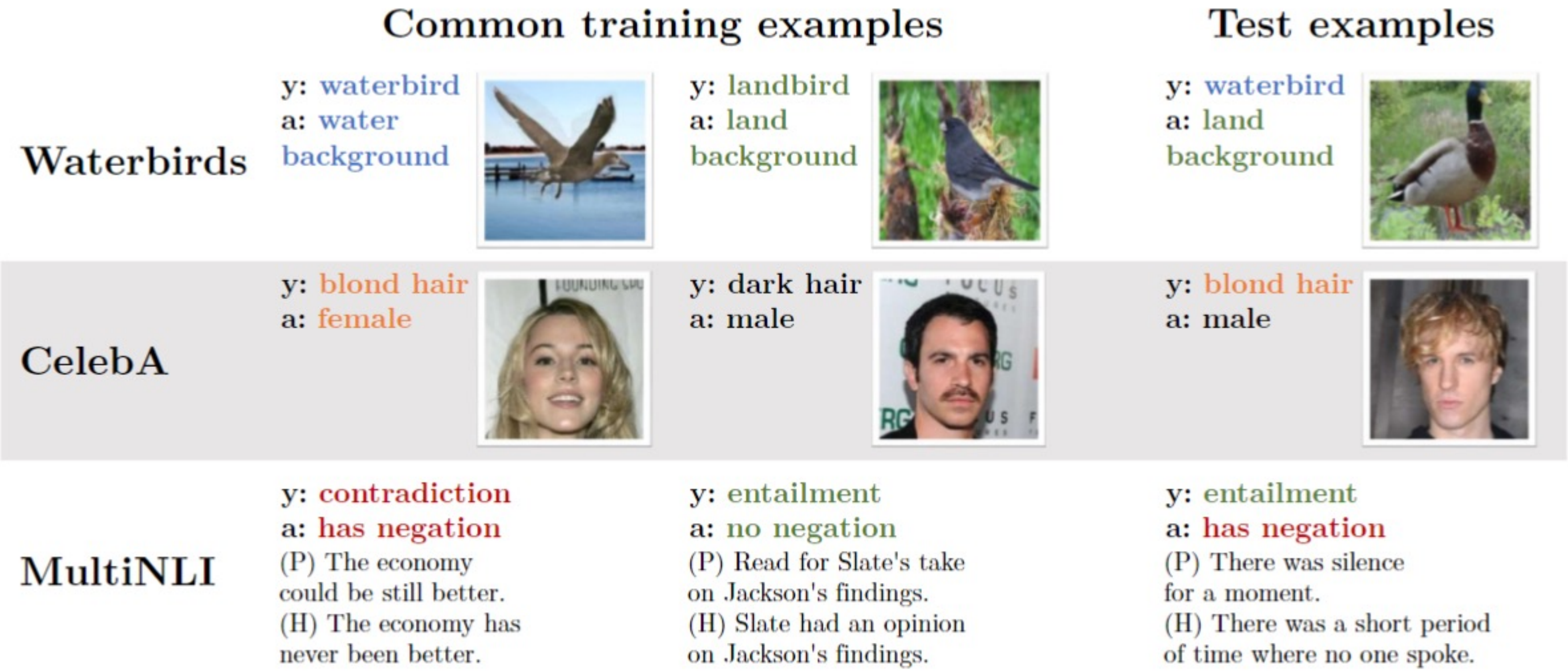
Domain Generalization using Causal Matching

Divyat Mahajan, Shruti Tople, Amit Sharma
Microsoft Research

ICML | 2021
Thirty-eighth International Conference on Machine Learning

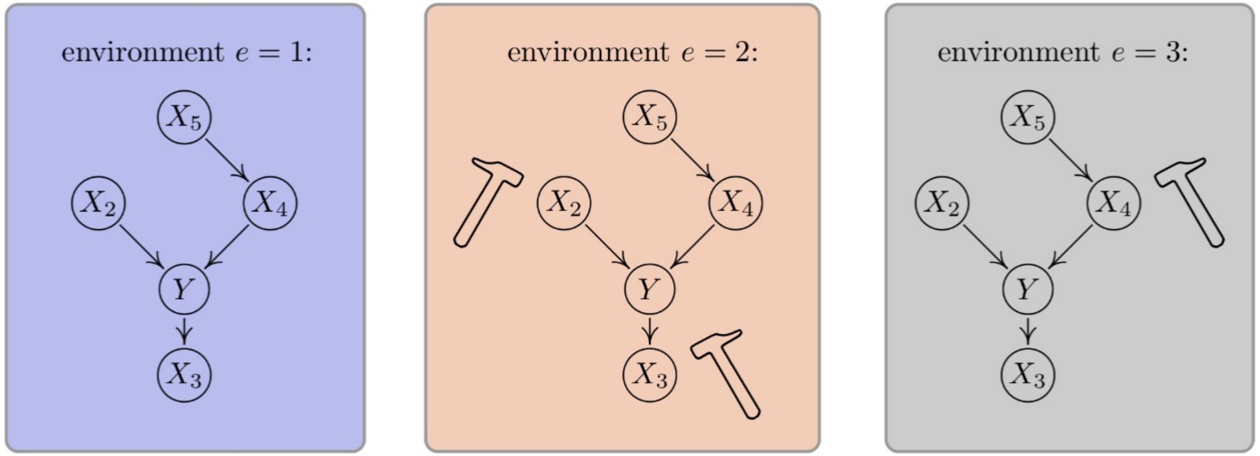
Microsoft

Spurious Correlations



Sagawa et al. (2019): Distributionally Robust Neural Networks

Domain Generalization



Peters et al. (2016): Causal Inference using Invariant Prediction

Goal: Learn a single classifier with training data sampled from M domains that generalizes well to data from unseen domains

Assumption: There exist stable (causal) features X_S which lead to an optimal classifier invariant to the changes in domains

Our Contributions

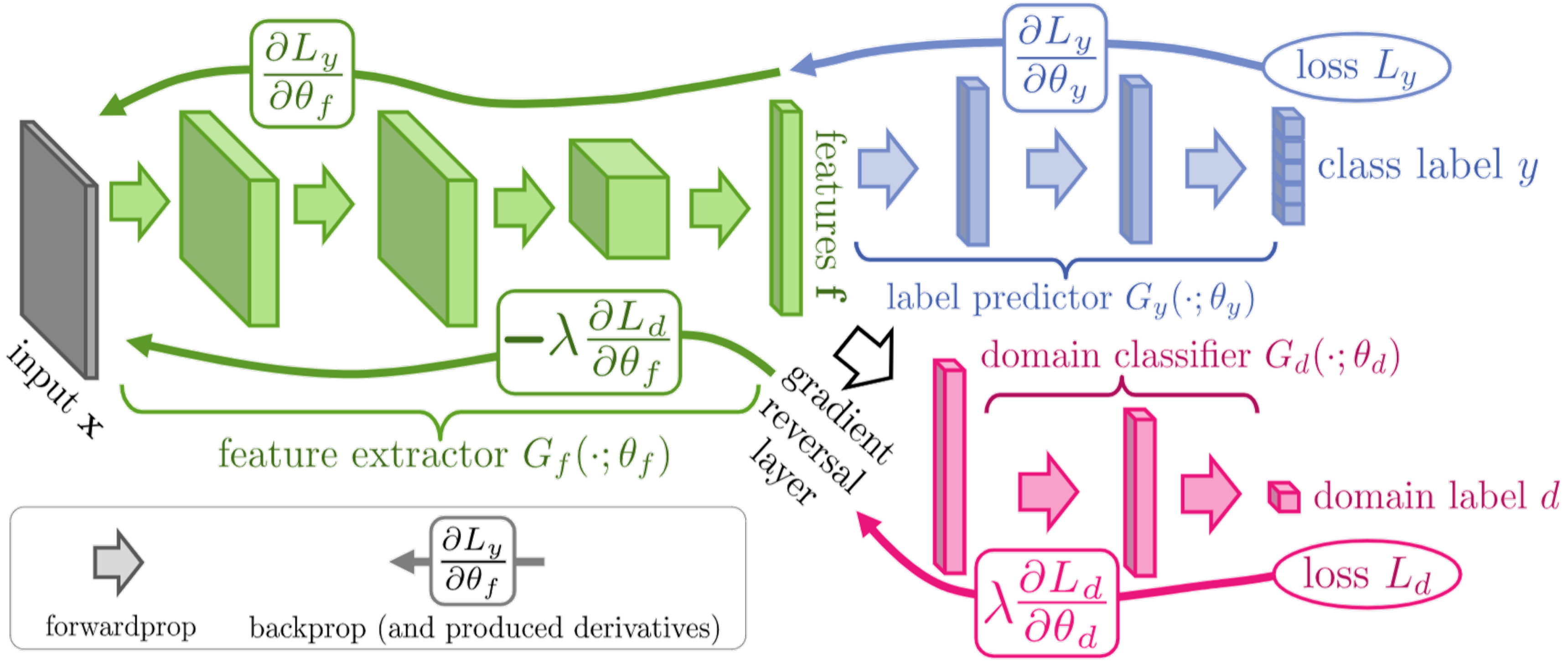
An object-invariant condition for domain generalization that highlights a key limitation of previous approaches

When object information is not available, a two-phase iterative algorithm to approximate object-based matches

Prior works based on domain invariant representation learning

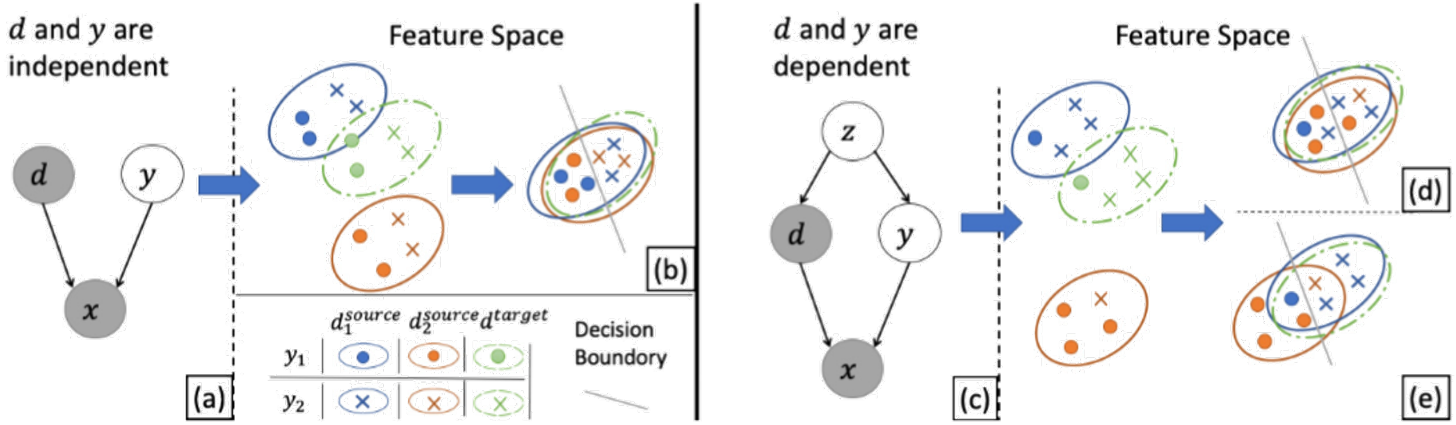
Domain Invariant Representations

Learning representation independent of domain ($\phi(x) \perp d$)



Ganin et al. (2016): Domain Adversarial Training

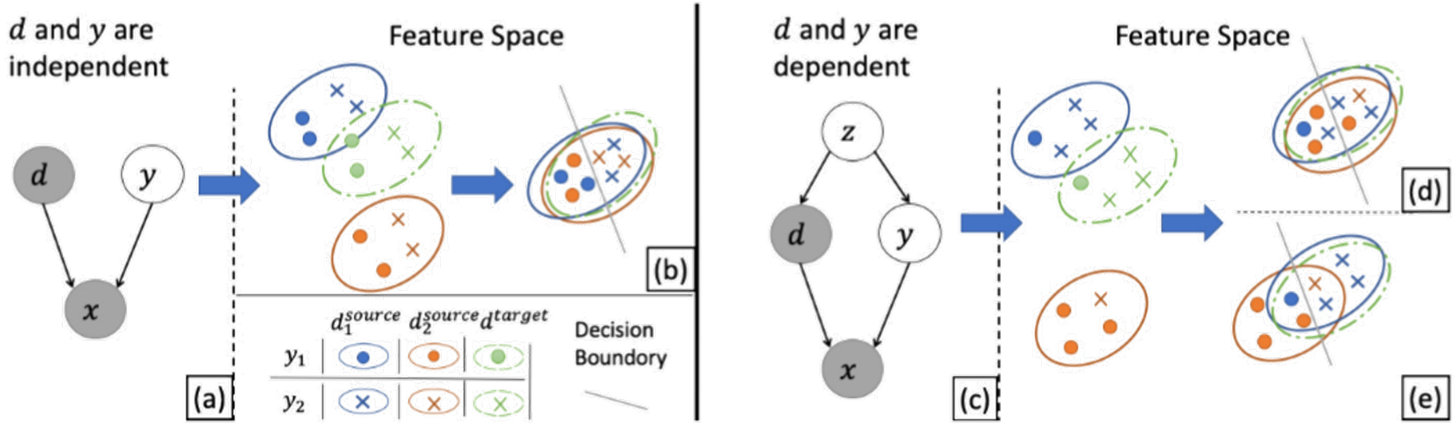
Failure Case: Domain and Label Correlated



Akuzawa et al. (2019): Adversarial Invariant Learning with Accuracy Constraint

Akuzawa et al. (2019): Dependence between domain (d) and label (y) leads to tradeoff between accuracy (predicting y from $\phi(x)$) and invariance ($\phi(x) \perp d$)

Failure Case: Domain and Label Correlated



Akuzawa et al. (2019): Adversarial Invariant Learning with Accuracy Constraint

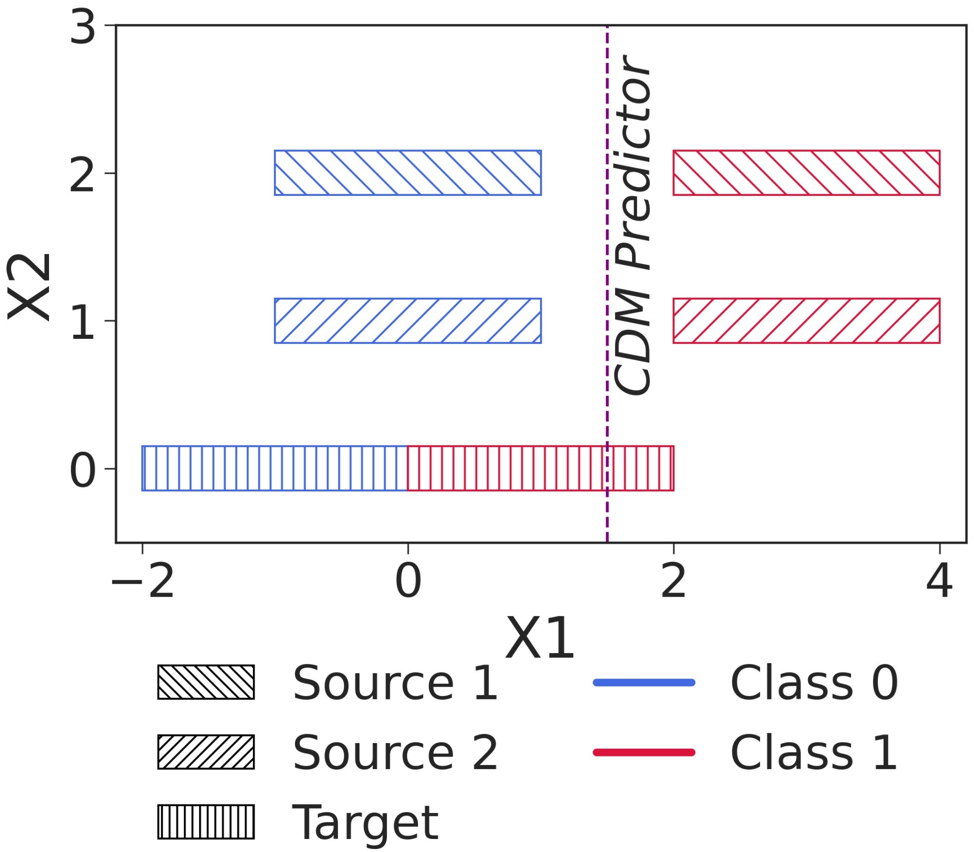
Akuzawa et al. (2019): Dependence between domain (d) and label (y) leads to tradeoff between accuracy (predicting y from $\phi(x)$) and invariance ($\phi(x) \perp d$)

Class-conditional domain invariant representations

Sun et al. (2016), Li et al. (2018): Learning representation independent of domain conditioned on class label ($\phi(x) \perp d \mid y$)

Is the class-conditional domain invariance objective correct?

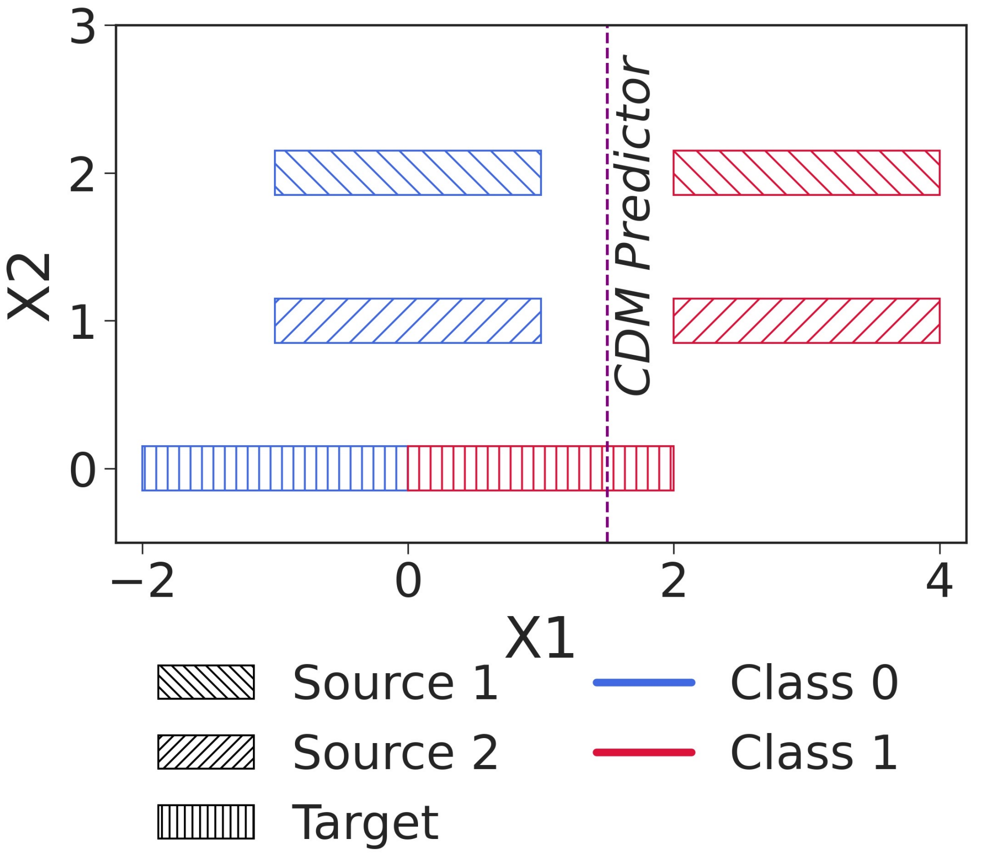
Simple Counter Example



Simple Counter Example

$x_1 = x_c + \alpha_d$; $x_2 = \alpha_d$
where x_c and α_d are unobserved

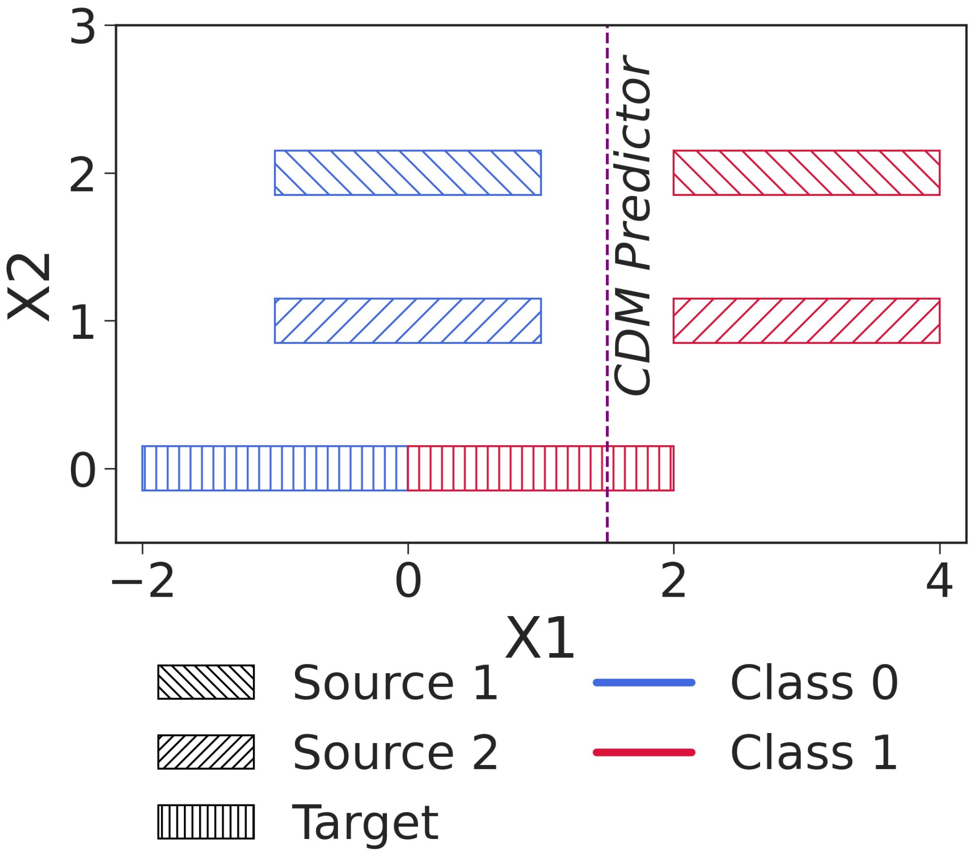
Invariant Predictor
$y = f(x_c) = I(x_c \geq 0)$



Simple Counter Example

$x_1 = x_c + \alpha_d$; $x_2 = \alpha_d$
where x_c and α_d are unobserved

Invariant Predictor
$y = f(x_c) = I(x_c \geq 0)$



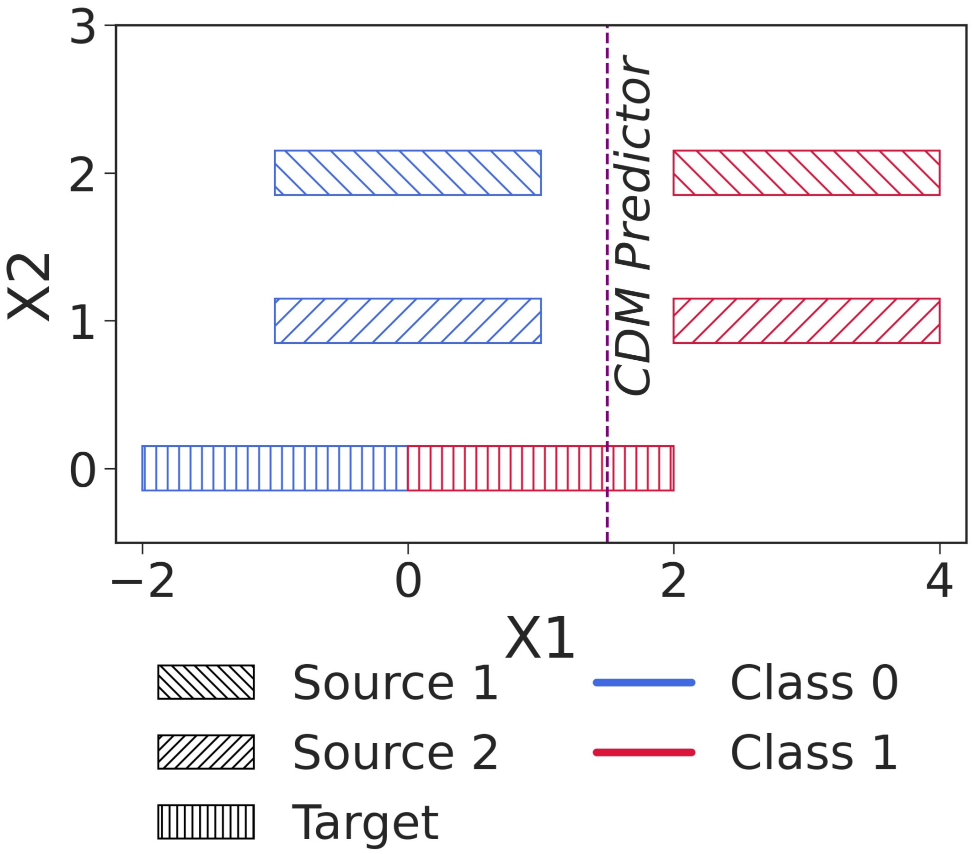
$\phi(x1, x2) = x1$
satisfies $\phi(x) \perp d \mid y$

But the optimal classifier on it gets 62.5% test accuracy

Simple Counter Example

$x_1 = x_c + \alpha_d$; $x_2 = \alpha_d$
where x_c and α_d are unobserved

Invariant Predictor
$y = f(x_c) = I(x_c \geq 0)$



$\phi(x1, x2) = x1$
satisfies $\phi(x) \perp \mathrm{d} \mid \mathrm{y}$

But the optimal classifier on it gets 62.5% test accuracy

Explanation: Distribution of stable features $p(x_c|y)$ changes across domains

Distribution of Stable Features Matter

Proposition: If $P(X_c|Y)$ remains the same across domains, then the class-conditional domain invariance yields a generalizable classifier such that the learnt representation $\phi(x)$ is independent of the domain given x_c

Distribution of Stable Features Matter

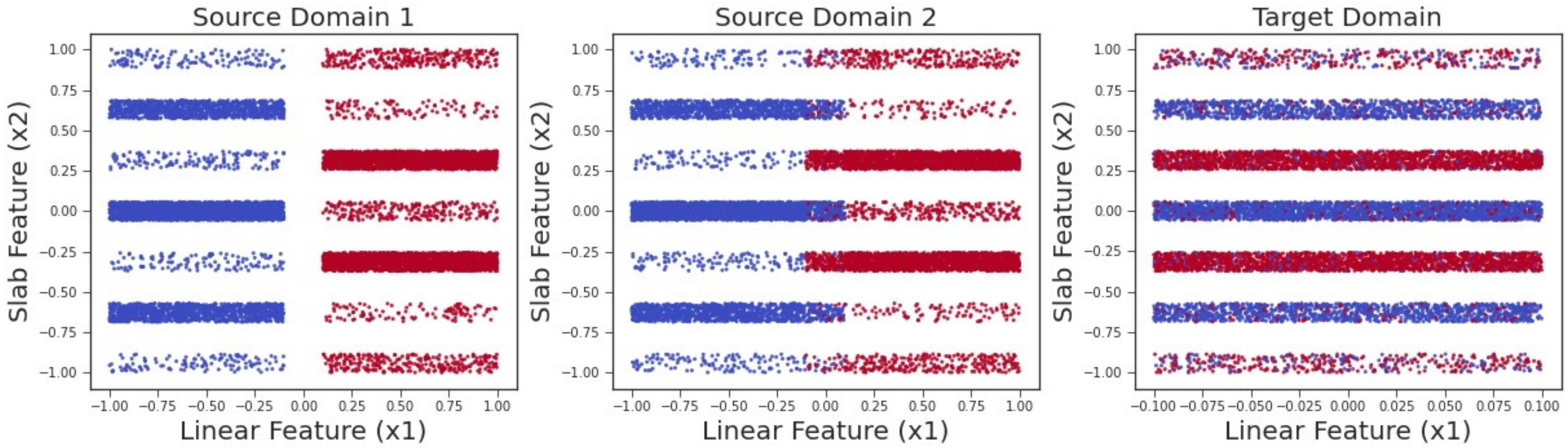
Proposition: If $P(X_c|Y)$ remains the same across domains, then the class-conditional domain invariance yields a generalizable classifier such that the learnt representation $\phi(x)$ is independent of the domain given x_c

Implication: $\phi(x)$ depends only on the stable features x_c if $P(X_c|Y)$ does not change across domains

New Invariance Criteria: $\phi(x) \perp d \mid x_c$

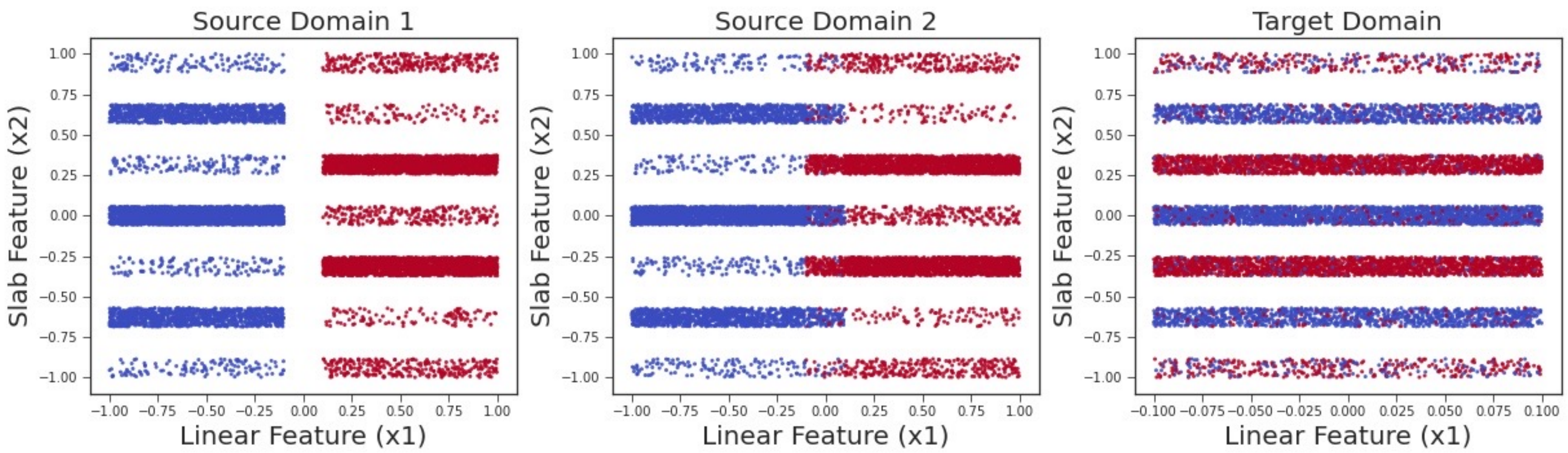
How to identify stable features?

Slab Dataset



Harshay et al. (2020): Simplicity Bias in Neural Networks

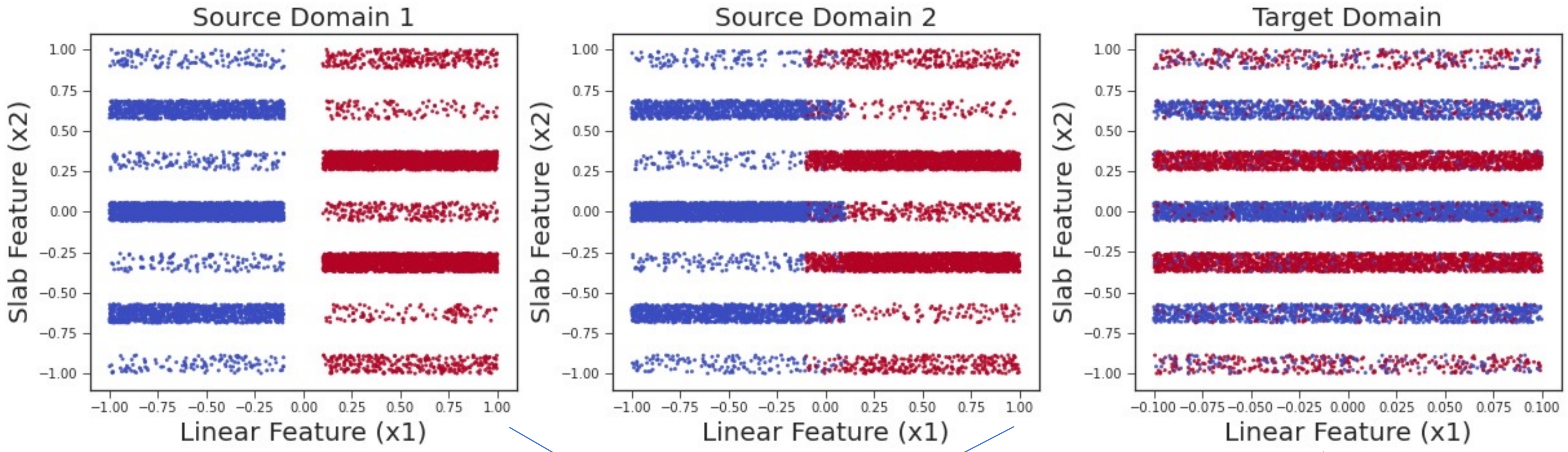
Slab Dataset



Stable Feature

Spurious Feature

Slab Dataset



Stable Feature

Spurious Feature

Low Noise in $(x1, y)$ relationship

High Noise in $(x1, y)$ relationship

Conditioning on stable features

Method	Source 1	Source 2	Target
ERM	**100.0** (0.0)	96.0 (0.25)	57.6 (6.58)
DANN	99.9 (0.07)	94.8 (0.25)	53.0 (1.41)
MMD	99.9 (0.01)	95.9 (0.27)	62.9 (5.01)
CORAL	99.9 (0.01)	96.0 (0.27)	63.1 (5.86)
RandMatch	**100.0** (0.0)	96.1 (0.22)	59.5 (3.50)
CDANN	99.9 (0.01)	96.0 (0.27)	55.9 (2.47)
C-MMD	99.9 (0.01)	96.0 (0.27)	58.9 (3.43)
C-CORAL	99.9 (0.01)	96.0 (0.27)	64.7 (4.69)
PerfMatch	99.9 (0.05)	**97.8** (0.28)	**77.8** (6.01)

Conditioning on stable features

Method	Source 1	Source 2	Target
ERM	**100.0** (0.0)	96.0 (0.25)	57.6 (6.58)
DANN	99.9 (0.07)	94.8 (0.25)	53.0 (1.41)
MMD	99.9 (0.01)	95.9 (0.27)	62.9 (5.01)
CORAL	99.9 (0.01)	96.0 (0.27)	63.1 (5.86)
RandMatch	**100.0** (0.0)	96.1 (0.22)	59.5 (3.50)
CDANN	99.9 (0.01)	96.0 (0.27)	55.9 (2.47)
C-MMD	99.9 (0.01)	96.0 (0.27)	58.9 (3.43)
C-CORAL	99.9 (0.01)	96.0 (0.27)	64.7 (4.69)
PerfMatch	99.9 (0.05)	**97.8** (0.28)	**77.8** (6.01)

Domain Invariant Representations

Class-Conditional Domain Invariant Representations

$\phi(x)$ independent of domain given stable (slab) feature

Conditioning on stable features

Method	Source 1	Source 2	Target
ERM	**100.0** (0.0)	96.0 (0.25)	57.6 (6.58)
DANN	99.9 (0.07)	94.8 (0.25)	53.0 (1.41)
MMD	99.9 (0.01)	95.9 (0.27)	62.9 (5.01)
CORAL	99.9 (0.01)	96.0 (0.27)	63.1 (5.86)
RandMatch	**100.0** (0.0)	96.1 (0.22)	59.5 (3.50)
CDANN	99.9 (0.01)	96.0 (0.27)	55.9 (2.47)
C-MMD	99.9 (0.01)	96.0 (0.27)	58.9 (3.43)
C-CORAL	99.9 (0.01)	96.0 (0.27)	64.7 (4.69)
PerfMatch	99.9 (0.05)	**97.8** (0.28)	**77.8** (6.01)

Domain Invariant Representations

Class-Conditional Domain Invariant Representations

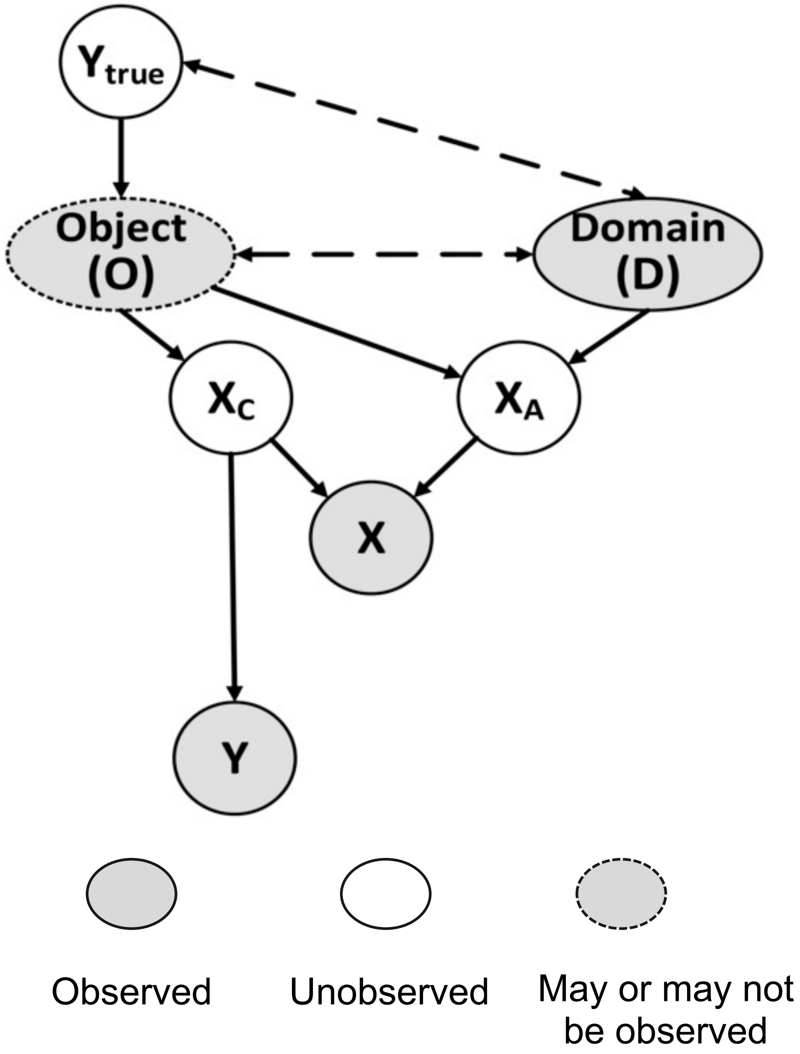
$\phi(x)$ independent of domain given stable (slab) feature

Fail to learn the stable (slab) feature

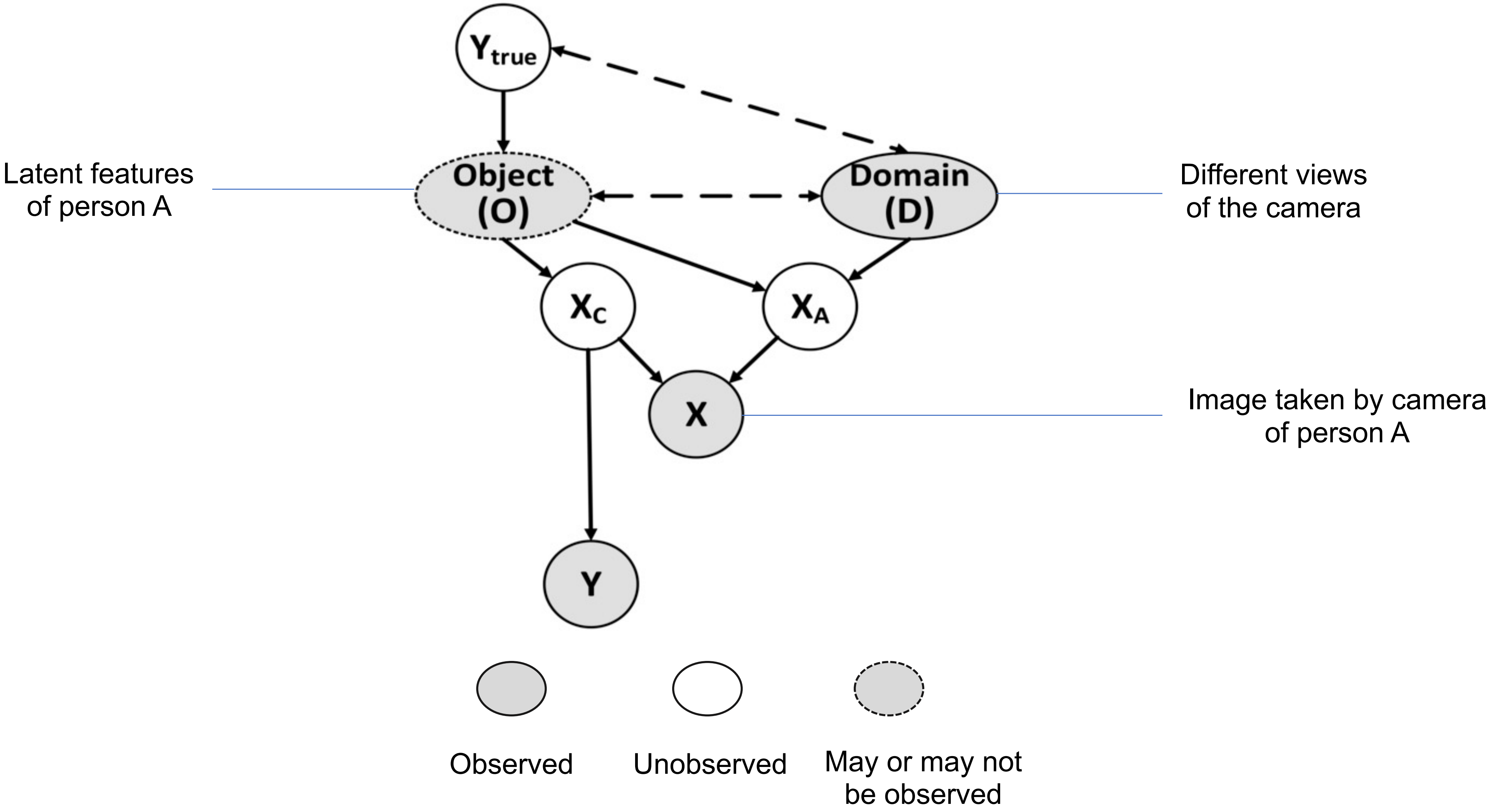
Better than prior approaches at learning the stable (slab) feature

Formalizing the intuition with causal graphs

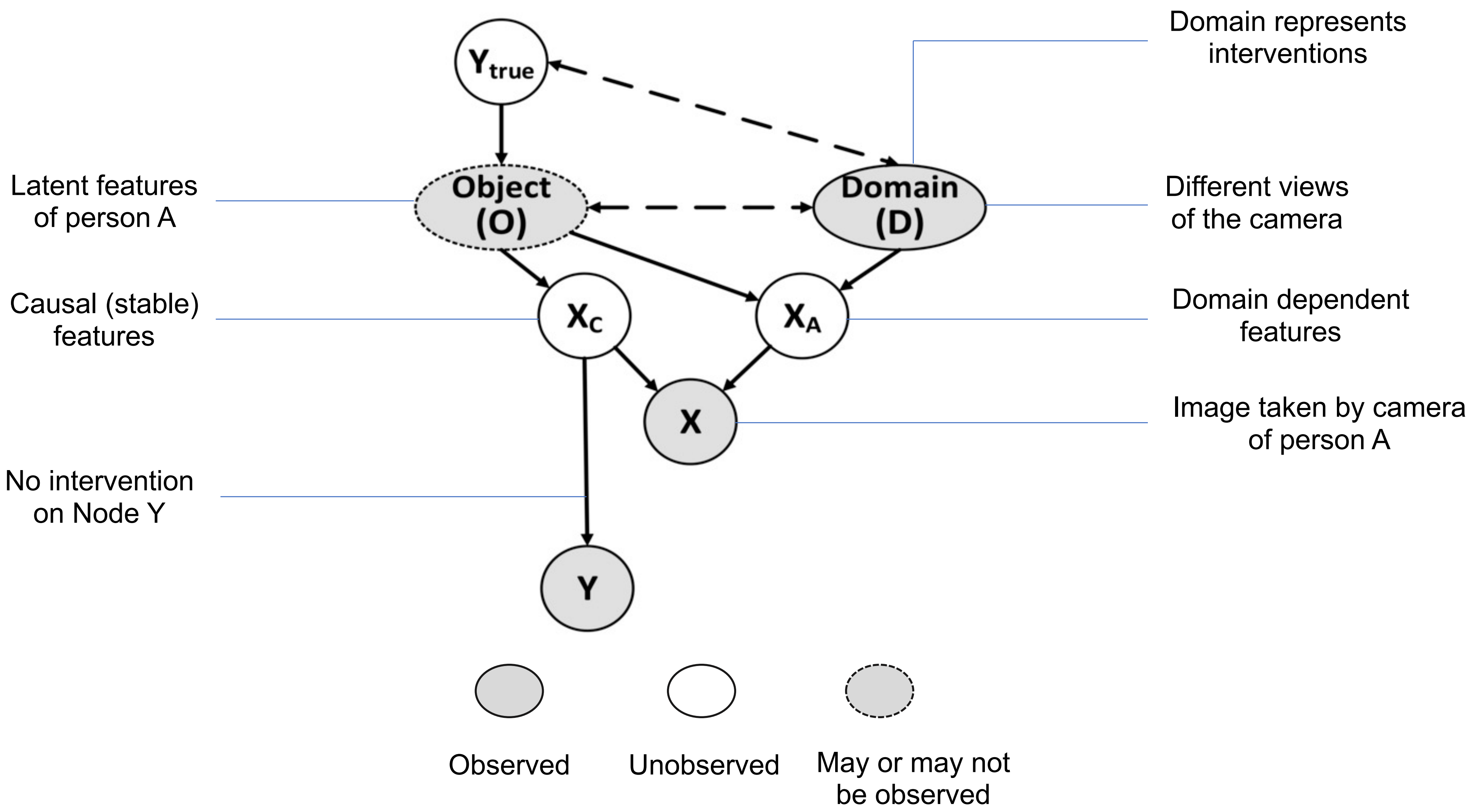
Causal Graph for Data Generating Process



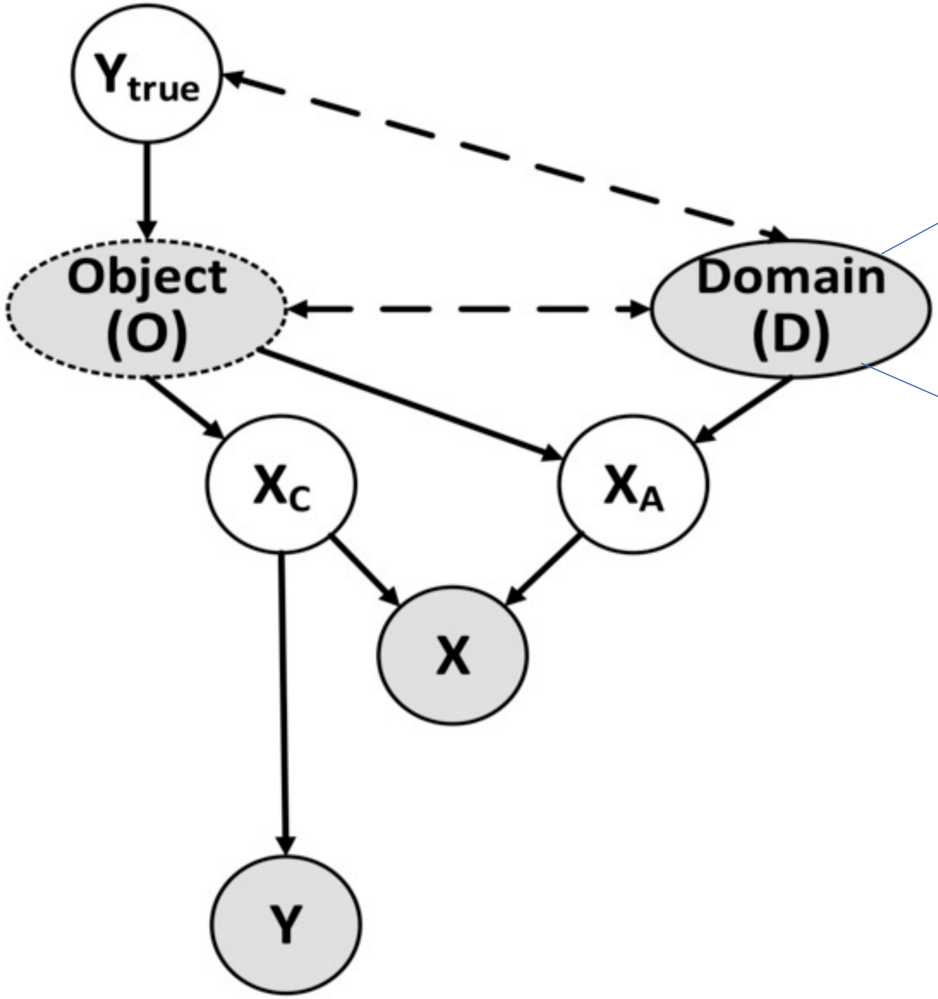
Causal Graph for Data Generating Process



Causal Graph for Data Generating Process



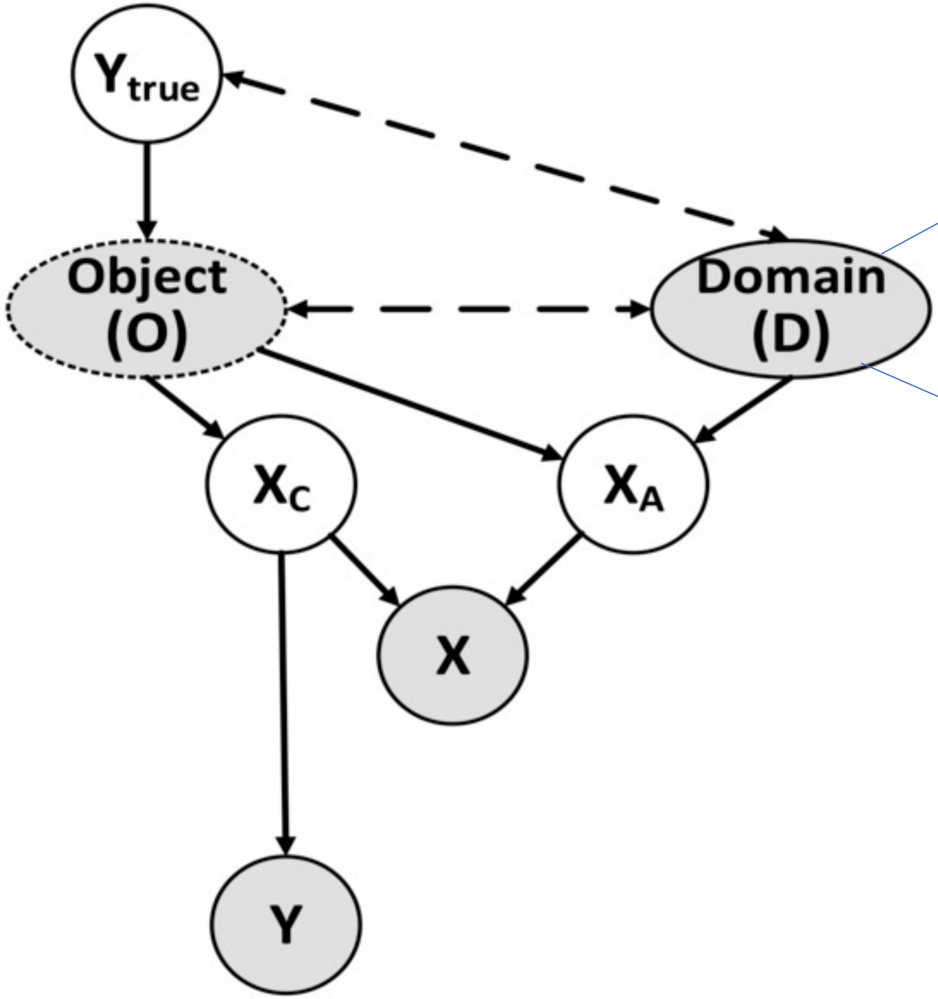
Correct Invariance Criteria



Intervention domain d — o, x_c, d, x^d

Intervention domain d' — $o, x_c, d', x^{d'}$

Correct Invariance Criteria



Intervention domain d $\quad o, x_c, d, x^d$

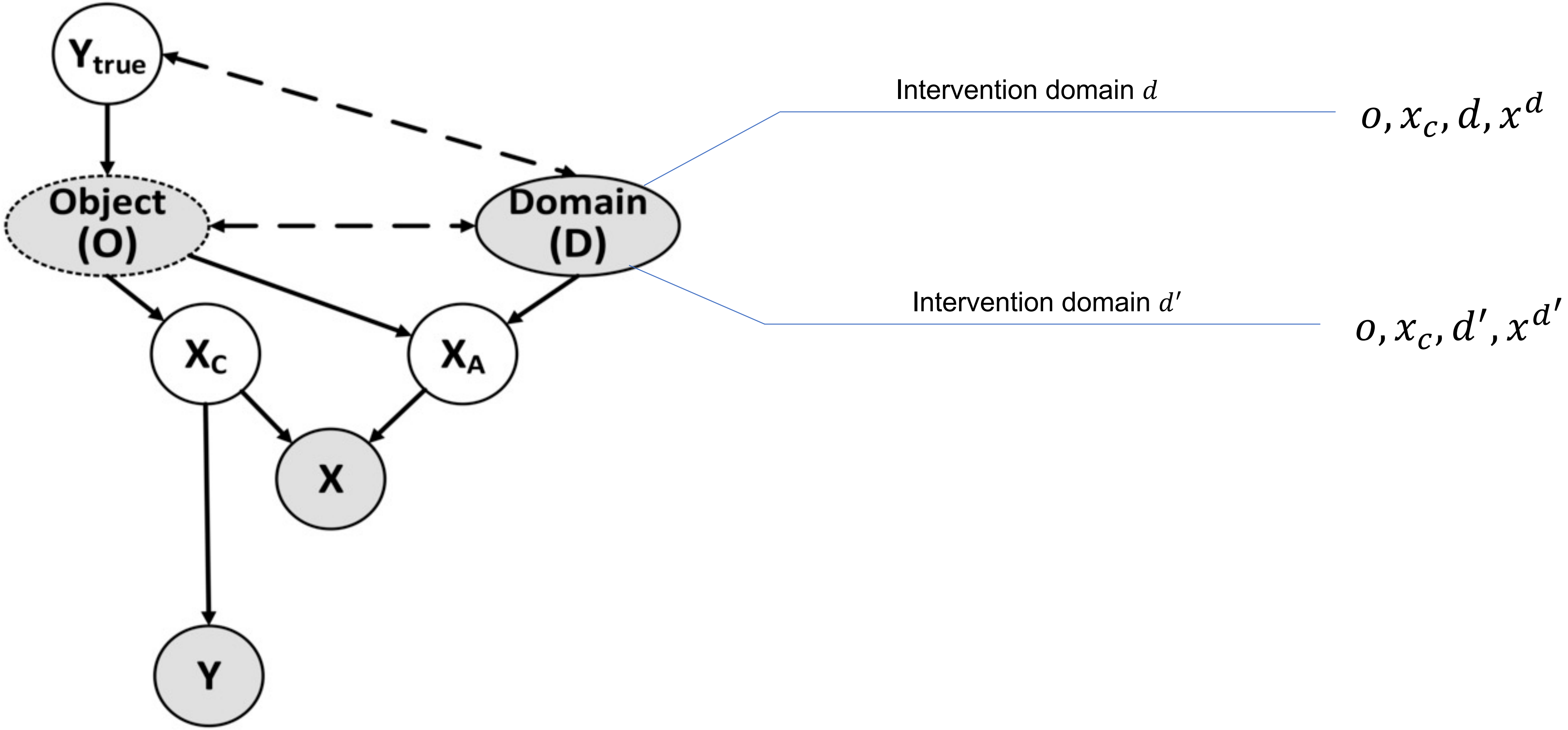
Intervention domain d' $\quad o, x_c, d', x^{d'}$

For each observed x^d, there are a set of counterfactual inputs $x^{d'}$ where $d \neq d'$, but both have similar causal features x_c

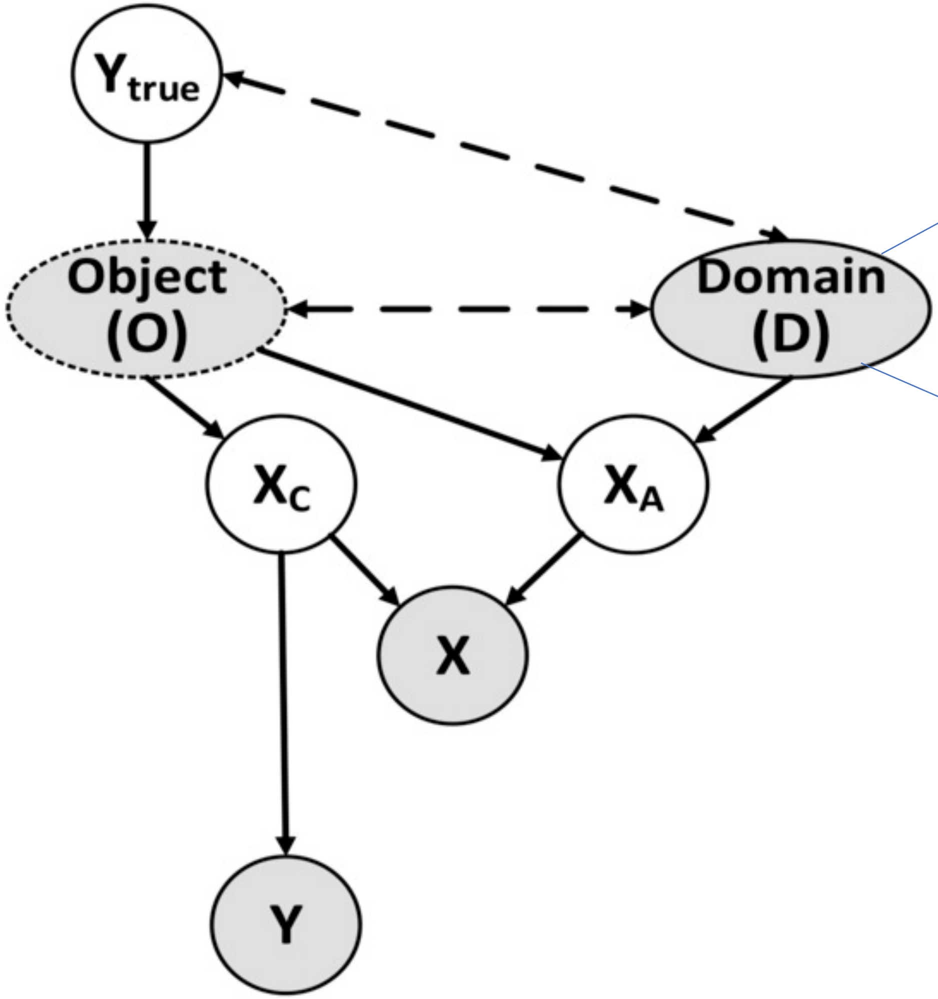
Invariance Criteria: $X_c \perp D \mid O$

How to satisfy the invariance criteria?

Match Counterfactuals



Match Counterfactuals



Intervention domain d — o, x_c, d, x^d

Intervention domain d' — $o, x_c, d', x^{d'}$

Invariance Constraint:
$\sum_{\Omega(j,k)=1} Dist(\phi(x_j^d), \phi(x_k^{d'})) = 0$; $\Omega = 1$ if $o_j^d = o_k^{d'}$,
$\Omega = 0$ otherwise

Note: The causal features (X_C) are unidentifiable and solving the above objective is not guaranteed to return the true causal features.

Perfect Match Approach

Aim: Learn representations $\phi(X)$ that satisfy the invariance criteria and are informative of the label Y across domains D

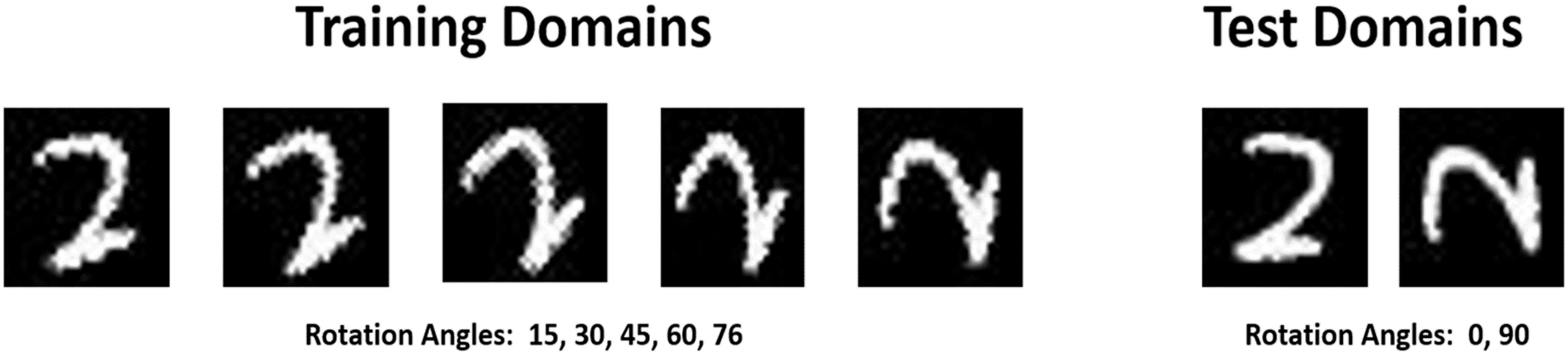
Perfect Match Approach

Aim: Learn representations $\phi(X)$ that satisfy the invariance criteria and are informative of the label Y across domains D

$$f_{perfectmatch} = \arg\min_{h,\phi} \sum_d L_d(h(\phi(X), Y) + \lambda * \sum_{\Omega(\mathrm{j,k})=1} Dist\left(\phi\left(x_j^d\right), \phi\left(x_k^{d'}\right)\right)$$

Theorem: It can be shown the optimal solutions $\phi(X) = X_c$ and $f = f^*$ are contained in the set of solutions obtained by solving $f_{perfectmatch}$

Perfect Match: Application



- **Match Function Known:** Same data point rotated by different angle across domains shares the same causal (stable) feature, hence the same base object
- Perfect match is applicable when we have self augmentations

How to proceed when we do not know the perfect matches across domains?

MatchDG: Matching without known objects

Goal: Learn a match function s.t. $\Omega(x, x') = 1$ when $\mathrm{Dist}(x_c, x_c')$ is low

Assumption: Let (x_i^d, y), $(x_j^{d'}, y)$ be any two points that belong the same class and let (x_k^d, y') be any other point that has a different class label. Then the distance in causal features between (x_i, x_j) and is smaller than that between (x_i, x_k) or (x_j, x_k)

MatchDG: Matching without known objects

Goal: Learn a match function s.t. $\Omega(x, x') = 1$ when $\mathrm{Dist}(x_c, x_c')$ is low

Assumption: Let (x_i^d, y), $(x_j^{d'}, y)$ be any two points that belong the same class and let (x_k^d, y') be any other point that has a different class label. Then the distance in causal features between (x_i, x_j) and is smaller than that between (x_i, x_k) or (x_j, x_k)

Data	Label	Domain	Object
x^1	1	1	$\boldsymbol{o^1}$
x^2	1	2	o^2
x^3	1	2	$\boldsymbol{o^1}$
x^4	0	2	o^3

Assumption

$$Dist(x_c^1, x_c^2) < Dist(x_c^1, x_c^4)$$
$$Dist(x_c^1, x_c^2) < Dist(x_c^2, x_c^4)$$
$$Dist(x_c^1, x_c^3) < Dist(x_c^1, x_c^4)$$
$$Dist(x_c^1, x_c^3) < Dist(x_c^3, x_c^4)$$

MatchDG: Matching without known objects

Contrastive Loss:

- Positive Matches: Specific data points from a different domain that share the same class label as the anchor
- Negative Matches: Any data point with a different class label from the anchor

MatchDG: Matching without known objects

Contrastive Loss:

- Positive Matches: Specific data points from a different domain that share the same class label as the anchor
- Negative Matches: Any data point with a different class label from the anchor

Data	Label	Domain	Object
x^1	1	1	$\boldsymbol{o^1}$
x^2	1	2	o^2
x^3	1	2	$\boldsymbol{o^1}$
x^4	0	2	o^3

Contrastive Loss with x^1 as anchor

Positive Match(x^1)= x^2
Negative Match(x^1)= x^4

$$\min_{\phi} Dist(\phi(x^1), \phi(x^2)) - Dist(\phi(x^1), \phi(x^4))$$

MatchDG: Matching without known objects

Iterative Contrastive Learning:

- Positive matches inferred using Ω are updated during training based on the nearest same-class data points in the representation space ϕ
- Iterative updates aim to account for the intra-class variance across domains

MatchDG: Matching without known objects

Iterative Contrastive Learning:

- Positive matches inferred using Ω are updated during training based on the nearest same-class data points in the representation space ϕ
- Iterative updates aim to account for the intra-class variance across domains

Data	Label	Domain	Object
x^1	1	1	$\boldsymbol{o^1}$
x^2	1	2	o^2
x^3	1	2	$\boldsymbol{o^1}$
x^4	0	2	o^3

Updated positive match for x^1

$$\min_i Dist\big(\phi(x^1), \phi(x^i)\big) \;\forall x^i \in d^2, y^1 = y^i$$

MatchDG: Matching without known objects

Iterative Contrastive Learning:

- Positive matches inferred using Ω are updated during training based on the nearest same-class data points in the representation space ϕ
- Iterative updates aim to account for the intra-class variance across domains

Data	Label	Domain	Object
x^1	1	1	$\boldsymbol{o^1}$
x^2	1	2	o^2
x^3	1	2	$\boldsymbol{o^1}$
x^4	0	2	o^3

Contrastive Loss with updated match

Positive Match(x^1)= x^3
Negative Match(x^1)= x^4

$$\min_{\phi} Dist(\phi(x^1), \phi(x^3)) - Dist(\phi(x^1), \phi(x^4))$$

MatchDG: Matching without known objects

MatchDG Phase 1: Learn a match function Ω using iterative contrastive learning

MatchDG Phase 2: Substitute Ω learnt using Phase 1 in the perfect match loss

$$f_{perfectmatch} = \arg\min_{h,\phi} \sum_d L_d(h(\phi(X), Y) + \lambda * \sum_{\Omega(\mathrm{j,k})=1} Dist\left(\phi\left(x_j^d\right), \phi\left(x_k^{d'}\right)\right)$$

Evaluation on benchmark datasets

MatchDG: OOD Accuracy

Dataset	ERM	Best Prior	Rand Match	MatchDG	MatchDG Hybrid	PerfMatch
Rot MNIST (5)	93.0	94.5	93.4	**95.1**	-	96.0
Rot MNIST (3)	76.2	77.7	78.3	**83.6**		89.7
Fashion MNIST (5)	77.9	78.7	77.0	**80.9**	-	81.6
Fashion MNIST (3)	36.1	37.8	38.4	**43.8**	-	54.0
PACS ResNet-18	81.7	**85.2**	81.9	83.2	84.4	-
PACS ResNet-50	85.7	**87.8**	85.5	86.1	87.5	-

MatchDG: OOD Accuracy

Dataset	ERM	Best Prior	Rand Match	MatchDG	MatchDG Hybrid	PerfMatch
Rot MNIST (5)	93.0	94.5	93.4	**95.1**	-	96.0
Rot MNIST (3)	76.2	77.7	78.3	**83.6**		89.7
Fashion MNIST (5)	77.9	78.7	77.0	**80.9**	-	81.6
Fashion MNIST (3)	36.1	37.8	38.4	**43.8**	-	54.0
PACS ResNet-18	81.7	**85.2**	81.9	83.2	84.4	-
PACS ResNet-50	85.7	**87.8**	85.5	86.1	87.5	-

Gap between MatchDG and baselines increases with fewer training domains

MatchDG: OOD Accuracy

Dataset	ERM	Best Prior	Rand Match	MatchDG	MatchDG Hybrid	PerfMatch
Rot MNIST (5)	93.0	94.5	93.4	**95.1**	-	96.0
Rot MNIST (3)	76.2	77.7	78.3	**83.6**		89.7
Fashion MNIST (5)	77.9	78.7	77.0	**80.9**	-	81.6
Fashion MNIST (3)	36.1	37.8	38.4	**43.8**	-	54.0
PACS ResNet-18	81.7	**85.2**	81.9	83.2	84.4	-
PACS ResNet-50	85.7	**87.8**	85.5	86.1	87.5	-

Simple matching methods competitive to the state-of-the-art methods on PACS

Gap between MatchDG and baselines increases with fewer training domains

MatchDG improves over DomainBed (ERM) with ResNet50 architecture

MatchDG: Stable Features

Dataset	Method	Overlap (%)	Top 10 Overlap (%)	Mean Rank
Rotated MNIST	ERM	15.8	48.8	27.4
	MatchDG (Default)	**28.9**	**64.2**	**18.6**
	MatchDG (PerfMatch)	47.4	83.8	6.2
Fashion MNIST	ERM	2.1	11.1	224.3
	MatchDG (Default)	**17.9**	**43.1**	**89.0**
	MatchDG (PerfMatch)	56.2	87.2	7.3

MatchDG: Stable Features

Fraction of ground truth matches in the learnt match function

Mean position of ground truth matches in the learnt match function

Dataset	Method	Overlap (%)	Top 10 Overlap (%)	Mean Rank
	ERM	15.8	48.8	27.4
Rotated MNIST	MatchDG (Default)	**28.9**	**64.2**	**18.6**
	MatchDG (PerfMatch)	47.4	83.8	6.2
	ERM	2.1	11.1	224.3
Fashion MNIST	MatchDG (Default)	**17.9**	**43.1**	**89.0**
	MatchDG (PerfMatch)	56.2	87.2	7.3

MatchDG: Stable Features

Fraction of ground truth matches in the learnt match function

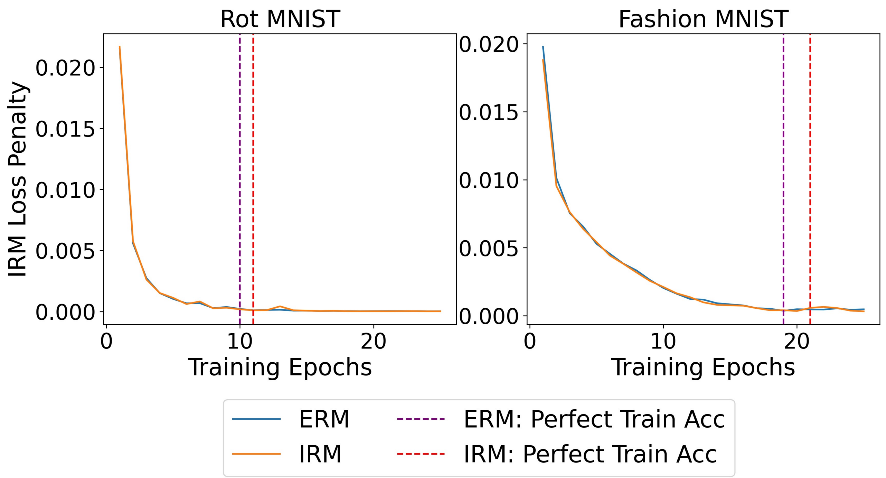
Mean position of ground truth matches in the learnt match function

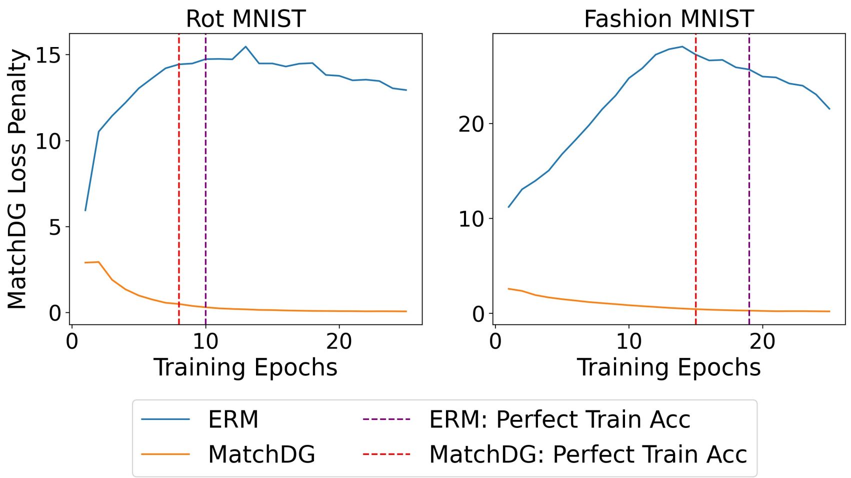
Dataset	Method	Overlap (%)	Top 10 Overlap (%)	Mean Rank
	ERM	15.8	48.8	27.4
Rotated MNIST	MatchDG (Default)	**28.9**	**64.2**	**18.6**
	MatchDG (PerfMatch)	47.4	83.8	6.2
	ERM	2.1	11.1	224.3
Fashion MNIST	MatchDG (Default)	**17.9**	**43.1**	**89.0**
	MatchDG (PerfMatch)	56.2	87.2	7.3

MatchDG has about 50% top-10 overlap on both datasets

MatchDG provides better match function than baseline ERM

MatchDG: Zero Training Error





- Zero training error does not imply similar representations within each class
- Methods with regularization based on comparing loss across domains such as IRM can be satisfied by ERM as the training error goes to zero

Chat more with us during the poster session!