Fairness under Graph Uncertainty: Achieving Interventional Fairness with Partially Known Causal Graphs over Clusters of Variables

uai2026

Yoichi Chikahara[1].

[1]Communication Science Laboratories. NTT, Inc.

This work was supported by JST ACT-X (JPMJAX23CF).

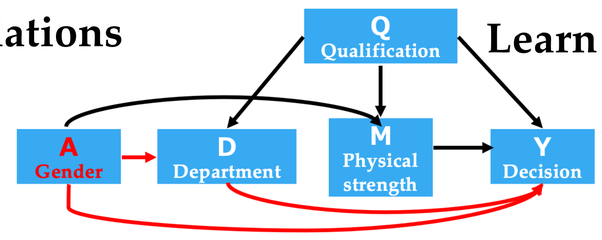
Problem Setup & Motivation: Can we achieve fairness using a *cluster causal graph*?

Input Training data

Sensitive feature

A Gender	Q Qualification	D Department	M Physical strength	Y Decision
Female	A	Economics	B	Accept
Male	B	Literature	B	Accept
Male	C	Science	C	Reject

Infer causal relations → Causal graph



Learn an ML model

Output Fair and accurate ML model

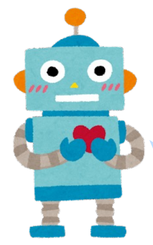
$\hat{Y} = f_\theta(A, Q, D, M)$

Learn model parameters θ

$$\min_\theta \quad \frac{1}{n}\sum_{i=1}^{n} \ell(\boldsymbol{y}_i, h_\theta(\boldsymbol{x}_i)) + \lambda g_\theta(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n),$$

Accurate causal graph inference is challenging in high-dimensional settings ☹

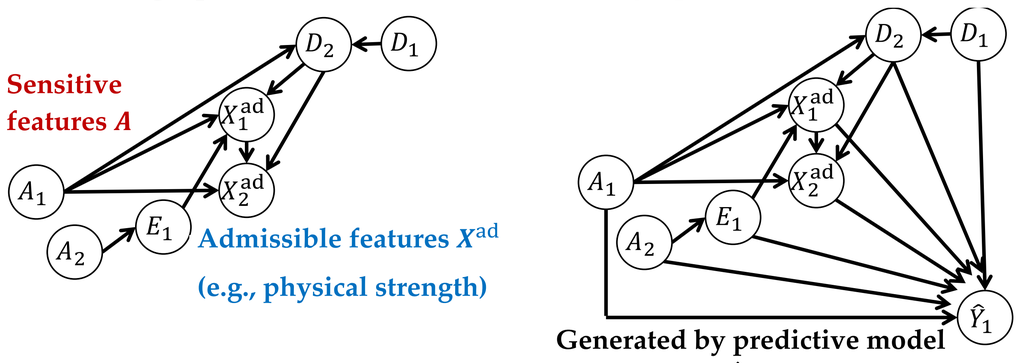
- We are **NOT** interested in the causal graph itself!
- **We just want to ensure fairness**



Q: Can we ensure fairness using **a *high-level* causal graph structure** representing macro-level relations among groups of variables?

Interventional Fairness

(a) Causal graph over features $\boldsymbol{X}$ (b) Causal graph over features $\boldsymbol{X}$ & prediction $\hat{\boldsymbol{Y}}$



Interventional fairness [Salimi+; SIGMOD2019]

$$\mathrm{P}(\hat{\boldsymbol{Y}} = \hat{\boldsymbol{y}} \mid do(\boldsymbol{A} = \boldsymbol{a}), do(\boldsymbol{X}^{\mathrm{ad}} = \boldsymbol{x}^{\mathrm{ad}})) = \mathrm{P}(\hat{\boldsymbol{Y}} = \hat{\boldsymbol{y}} \mid do(\boldsymbol{A} = \boldsymbol{a}'), do(\boldsymbol{X}^{\mathrm{ad}} = \boldsymbol{x}^{\mathrm{ad}}))$$

Graphical Algorithm & Learning Framework

1. Possible parent conditions

When can the siblings be the *possible parents* of $\boldsymbol{A}$ (i.e., the parents of $\boldsymbol{A}$ in some cluster DAGs in the MEC)?

Theorem 3.5: *A set of clusters* $S \subseteq \mathrm{sib}(\boldsymbol{A}, \mathcal{G}) := \{\boldsymbol{C} \in \mathcal{C} : \boldsymbol{C} - \boldsymbol{A} \text{ in } \mathcal{G}\}$ *in a cluster CPDAG* $\mathcal{G}$ *is a possible parent set of* $\boldsymbol{A}$ *if and only if it satisfies the following two conditions:*

1. ***Collider-structure compatibility**: All distinct* $\mathbf{U}, \boldsymbol{V} \in S$ *satisfy either (i)* $\mathbf{U}$ *and* $\boldsymbol{V}$ *are adjacent, or (ii)* $\mathbf{U}$ *and* $\boldsymbol{V}$ *are not adjacent and* $\mathcal{A}_{\langle \mathbf{U}, \boldsymbol{A}, \boldsymbol{V} \rangle} = \mathsf{never}$.
2. ***No back-path condition**: Let* $\mathcal{G}_{\to}$ *be the directed subgraph of* $\mathcal{G}$ *obtained by removing all undirected edges. For every* $\mathbf{U} \in S$ *and every* $\boldsymbol{V} \in \mathrm{sib}(\boldsymbol{A}, \mathcal{G}) \backslash S$, *there is no directed path from* $\boldsymbol{V}$ *to* $\mathbf{U}$ *in* $\mathcal{G}_{\to}$.

$$\boldsymbol{Z}^m := \mathrm{pa}(\boldsymbol{A}) \cup S^m$$

If $\boldsymbol{Z}^{\mathrm{m}}$ contains *connection-marked* clusters…

2. Cluster addition algorithm

Algorithm 1 Adjustment Cluster Completion $(\boldsymbol{Z}^m, \boldsymbol{A}, \mathcal{G})$

Require: $\boldsymbol{Z}^m, \boldsymbol{A}$, and cluster CPDAG $\mathcal{G}$ over $\boldsymbol{X}$

$\boldsymbol{Z} \leftarrow \boldsymbol{Z}^m$ in Eq. (6); Processed $\leftarrow \emptyset$
$\boldsymbol{Z}_{\oplus} \leftarrow \boldsymbol{Z} \cap \{\boldsymbol{C}_x$ in $\oplus \boldsymbol{C}_x$ in $\mathcal{A}_{\langle \boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{R} \rangle}$ for all triples in $\mathcal{G}\}$
Queue $\leftarrow \{(\boldsymbol{P}, \boldsymbol{Q}) : \boldsymbol{P} \in \boldsymbol{Z}_{\oplus}; \boldsymbol{Q}$ adjacent to $\boldsymbol{P}; \boldsymbol{Q} \neq \boldsymbol{A}\}$
while Queue $\neq \emptyset$ **do**
  pop $(\boldsymbol{P}, \boldsymbol{Q})$ from Queue
  if $(\boldsymbol{P}, \boldsymbol{Q}) \in$ Processed **then**
    continue
  Processed $\leftarrow$ Processed $\cup \{(\boldsymbol{P}, \boldsymbol{Q})\}$
  for each $\boldsymbol{R}$ adjacent to $\boldsymbol{Q}$ in $\mathcal{G}$ and $\boldsymbol{R} \neq \boldsymbol{P}$ **do**
    if $\mathcal{A}_{\langle \boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{R} \rangle} = \mathsf{marg}$ **then**
      $\boldsymbol{Z} \leftarrow \boldsymbol{Z} \cup \{\boldsymbol{Q}\}$
    else if $\mathcal{A}_{\langle \boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{R} \rangle} = \mathsf{never}$ **then**
      $\boldsymbol{Z} \leftarrow \boldsymbol{Z} \cup \{\boldsymbol{Q}\}$
      if Some $\boldsymbol{P}_x \in \boldsymbol{Z}$ in $\oplus \boldsymbol{C}_x$ in $\mathcal{A}_{\langle \boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{R} \rangle}$ **then**
        push $(\boldsymbol{Q}, \boldsymbol{R})$ into Queue
    else if $\mathcal{A}_{\langle \boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{R} \rangle} = \mathsf{cond}$ **then**
      $\boldsymbol{E} \leftarrow \{\boldsymbol{Q}\} \cup \mathrm{PossDesc}(\mathcal{G}, \boldsymbol{Q})$; $\boldsymbol{E} \leftarrow \boldsymbol{Z} \cap \boldsymbol{E}$
      if $\boldsymbol{E} \neq \emptyset$ and $\boldsymbol{Z}$ has no cluster in $\oslash \boldsymbol{C}_x$ in $\mathcal{A}_{\langle \boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{R} \rangle}$ **then**
        return Unidentifiable
if FinalCert$(\boldsymbol{Z}, \boldsymbol{A}, \mathcal{G})$[1]returns Unidentifiable **then**
  return Unidentifiable
return $(\boldsymbol{Z}, \mathrm{OK})$

Adjustment cluster sets $\boldsymbol{Z}^1, \ldots, \boldsymbol{Z}^{\mathrm{M}}$

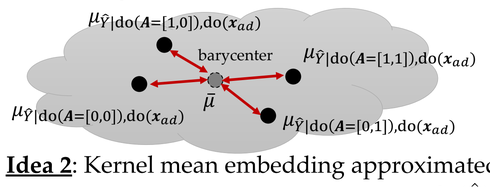
Worst-case unfairness penalization

Use the maximum of the kernel MMDs over $\boldsymbol{Z}^1, \ldots, \boldsymbol{Z}^{\mathrm{M}}$:

$$\max_m \sum_{\boldsymbol{x}^{\mathrm{ad}}} \sum_{\boldsymbol{a}, \boldsymbol{a}'} \mathrm{MMD}_m(\mathrm{P}_{\hat{\boldsymbol{Y}} | do(\boldsymbol{a}), do(\boldsymbol{x}^{\mathrm{ad}})}, \mathrm{P}_{\hat{\boldsymbol{Y}} | do(\boldsymbol{a}'), do(\boldsymbol{x}^{\mathrm{ad}})}),$$

which, however, needs $\mathcal{O}(M N_{\boldsymbol{A}}^2 N_{\boldsymbol{X}^{\mathrm{ad}}} n^2)$ time☹

Idea 1: Barycenter kernel MMD



Idea 2: Kernel mean embedding approximated with RFFs

$$\mu_{\hat{\boldsymbol{Y}} | do(\boldsymbol{a}), do(\boldsymbol{x}^{\mathrm{ad}})} := \mathbb{E}_{\hat{\boldsymbol{Y}} | do(\boldsymbol{A} = \boldsymbol{a}), do(\boldsymbol{X}^{\mathrm{ad}} = \boldsymbol{x}^{\mathrm{ad}})}[\phi(\hat{\boldsymbol{Y}})]$$

$$\phi_{\mathrm{RFF}}(\boldsymbol{y}) = [\cos(\omega_1^\top \boldsymbol{y} + b_1), \ldots, \cos(\omega_{d_{\mathrm{RFF}}}^\top \boldsymbol{y} + b_{d_{\mathrm{RFF}}})]^\top$$

Penalty function with $\mathcal{O}(M N_{\boldsymbol{A}} N_{\boldsymbol{X}^{\mathrm{ad}}} n d_{\mathrm{RFF}})$ time ☺

$$g_\theta(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \max_{m=1,\ldots,M} \sum_{\boldsymbol{x}^{\mathrm{ad}}} \sum_{\boldsymbol{a}} \| \hat{\mu}_{\hat{\boldsymbol{Y}} | do(\boldsymbol{a}), do(\boldsymbol{x}^{\mathrm{ad}})} - \hat{\bar{\mu}}_{\hat{\boldsymbol{Y}} | do(\boldsymbol{x}^{\mathrm{ad}})} \|_2^2,$$

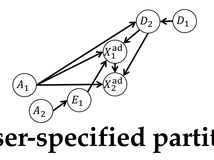
$$\hat{\mu}_{\hat{\boldsymbol{Y}} | do(\boldsymbol{a}), do(\boldsymbol{x}^{\mathrm{ad}})} = \frac{1}{n} \sum_{i=1}^{n} w_i \phi_{\mathrm{RFF}}(\hat{\boldsymbol{y}}_i)$$

IPW weight based on $\boldsymbol{Z}^{\mathrm{m}}$

$$w_i = \frac{\mathbf{I}(\boldsymbol{A}_i = \boldsymbol{a}, \boldsymbol{X}_i^{\mathrm{ad}} = \boldsymbol{x}^{\mathrm{ad}})}{\pi_{\boldsymbol{a}, \boldsymbol{x}^{\mathrm{ad}}}(\boldsymbol{z}_i^m)}$$

Cluster DAGs & Cluster CPDAGs

(a) Causal DAG over $\boldsymbol{X}$



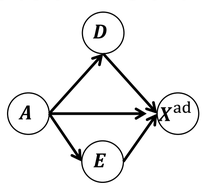
User-specified partition

$\mathcal{C} = \{\boldsymbol{C}_1, \ldots, \boldsymbol{C}_d\}$ $\quad \boldsymbol{C}_i \to \boldsymbol{C}_j$

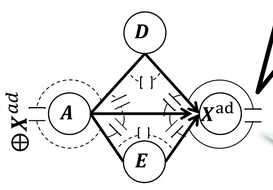
Assumption: No cycle is induced by partition

if some $V_i \in \boldsymbol{C}_i$ and $V_j \in \boldsymbol{C}_j$ have an edge $V_i \to V_j$ in G_{v}

(c) Cluster DAG over $\boldsymbol{X}$



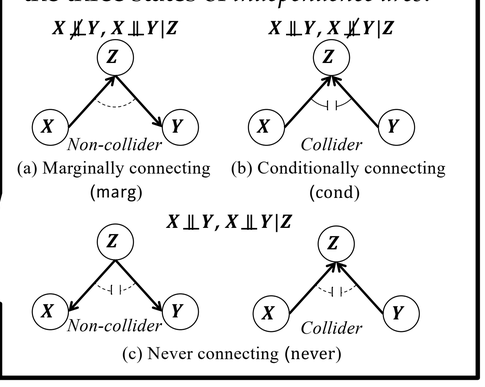
(d) Cluster CPDAG over $\boldsymbol{X}$



Observational Data → CI relations

Represent a class of cluster DAGs with identical CI relations

CI relations are expressed using the three states of *independence arcs*:



Easy to estimate ☺

Number of required CI tests is much smaller

Difficulty & Our Proposed Strategies

Q: How can we infer interventional distributions using a cluster CPDAG?

If we have access to the variable-level DAG:

We can identify back-door adjustment variables $\boldsymbol{Z}$ from the DAG and compute

$$\mathrm{P}(\hat{\boldsymbol{Y}} = \hat{\boldsymbol{y}} \mid do(\boldsymbol{A} = \boldsymbol{a}), do(\boldsymbol{X}^{\mathrm{ad}} = \boldsymbol{x}^{\mathrm{ad}})) = \mathbb{E}_{\boldsymbol{Z}, \boldsymbol{X}^{\mathrm{re}} | \boldsymbol{a}, \boldsymbol{x}^{\mathrm{ad}}} \left[\mathrm{P}(\hat{\boldsymbol{Y}} = \hat{\boldsymbol{y}} | \boldsymbol{A} = \boldsymbol{a}, \boldsymbol{X}^{\mathrm{ad}} = \boldsymbol{x}^{\mathrm{ad}}, \boldsymbol{Z}, \boldsymbol{X}^{\mathrm{re}}) \right]$$

Difficulty: Cluster CPDAG represents **multiple DAGs** (in the cluster MEC) ☹

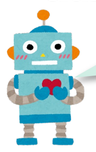
Our idea: We obtain *adjustment candidate sets* $\boldsymbol{Z}^1, \ldots, \boldsymbol{Z}^{\mathrm{M}}$, at least one of which d-separates all back-door paths for the true DAG. We take 2 steps:

1. **Parent Enumeration:** From possible DAGs, enumerate the parents of $\boldsymbol{A}$ as

   Definite & possible **parents**

   $$\boldsymbol{Z}^m := \mathrm{pa}(\boldsymbol{A}) \cup S^m \qquad \text{for } m = 1, \ldots, M$$

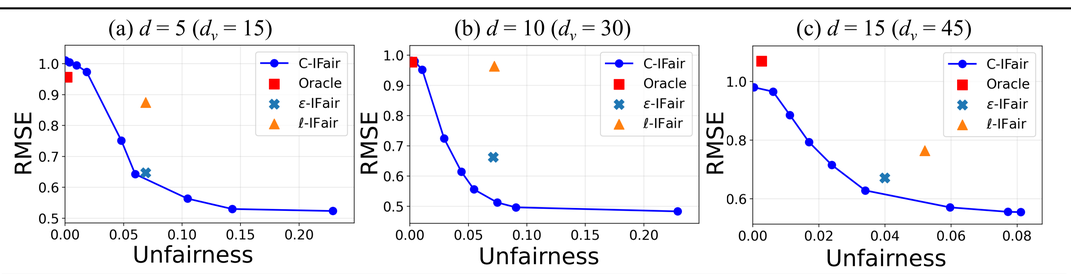
2. **Adjustment Set Completion**: Add clusters that are needed to achieve cluster d-separation [Anand+; NeurIPS2025]



After taking these 2 steps, we formulate unfairness penalty g_θ as the **maximum** of the distributional distances **over** $\boldsymbol{Z}^1, \ldots, \boldsymbol{Z}^{\mathrm{M}}$

Experimental Results

	d=5 (d_v=15)		d=10 (d_v=30)		d=15 (d_v=45)	
LINEAR	RMSE↓	UNFAIRNESS↓	RMSE↓	UNFAIRNESS↓	RMSE↓	UNFAIRNESS↓
ORACLE	0.957 ± 0.037	0.000 ± 0.000	0.977 ± 0.044	0.000 ± 0.000	1.069 ± 0.143	0.000 ± 0.000
FULL	0.523 ± 0.195	0.259 ± 0.129	0.483 ± 0.195	0.229 ± 0.136	0.554 ± 0.210	0.081 ± 0.117
UNAWARE	0.774 ± 0.154	0.071 ± 0.059	0.774 ± 0.154	0.062 ± 0.079	0.793 ± 0.120	0.046 ± 0.104
NO-DESCS	0.739 ± 0.140	0.064 ± 0.089	0.739 ± 0.140	0.065 ± 0.100	0.699 ± 0.135	0.033 ± 0.028
ϵ-IFAIR	0.647 ± 0.030	0.069 ± 0.031	0.663 ± 0.062	0.071 ± 0.011	0.671 ± 0.044	0.040 ± 0.011
ℓ-IFAIR	0.875 ± 0.028	0.069 ± 0.081	0.964 ± 0.049	0.072 ± 0.052	0.764 ± 0.049	0.052 ± 0.052
C-IFAIR	0.643 ± 0.127	0.060 ± 0.054	0.660 ± 0.123	0.056 ± 0.052	0.669 ± 0.171	0.020 ± 0.036



	ADULT		GERMAN		OULAD	
	AUC↑	UNFAIRNESS↓	AUC↑	UNFAIRNESS↓	AUC↑	UNFAIRNESS↓
ORACLE	0.709 ± 0.009	0.000 ± 0.000	0.582 ± 0.004	0.000 ± 0.000	0.628 ± 0.015	0.000 ± 0.000
FULL	0.874 ± 0.004	0.030 ± 0.024	0.762 ± 0.060	0.173 ± 0.011	0.697 ± 0.024	0.061 ± 0.001
UNAWARE	0.805 ± 0.049	0.018 ± 0.014	0.695 ± 0.061	0.101 ± 0.013	0.654 ± 0.019	0.004 ± 0.000
NO-DESCS	0.815 ± 0.007	0.026 ± 0.022	0.726 ± 0.059	0.085 ± 0.008	0.654 ± 0.020	0.004 ± 0.000
ϵ-IFAIR	0.812 ± 0.006	0.015 ± 0.012	0.756 ± 0.008	0.082 ± 0.000	0.641 ± 0.005	0.005 ± 0.002
ℓ-IFAIR	0.764 ± 0.002	0.015 ± 0.008	0.750 ± 0.035	0.151 ± 0.011	0.639 ± 0.011	0.003 ± 0.001
C-IFAIR	0.829 ± 0.021	0.014 ± 0.010	0.760 ± 0.061	0.065 ± 0.008	0.660 ± 0.018	0.001 ± 0.000