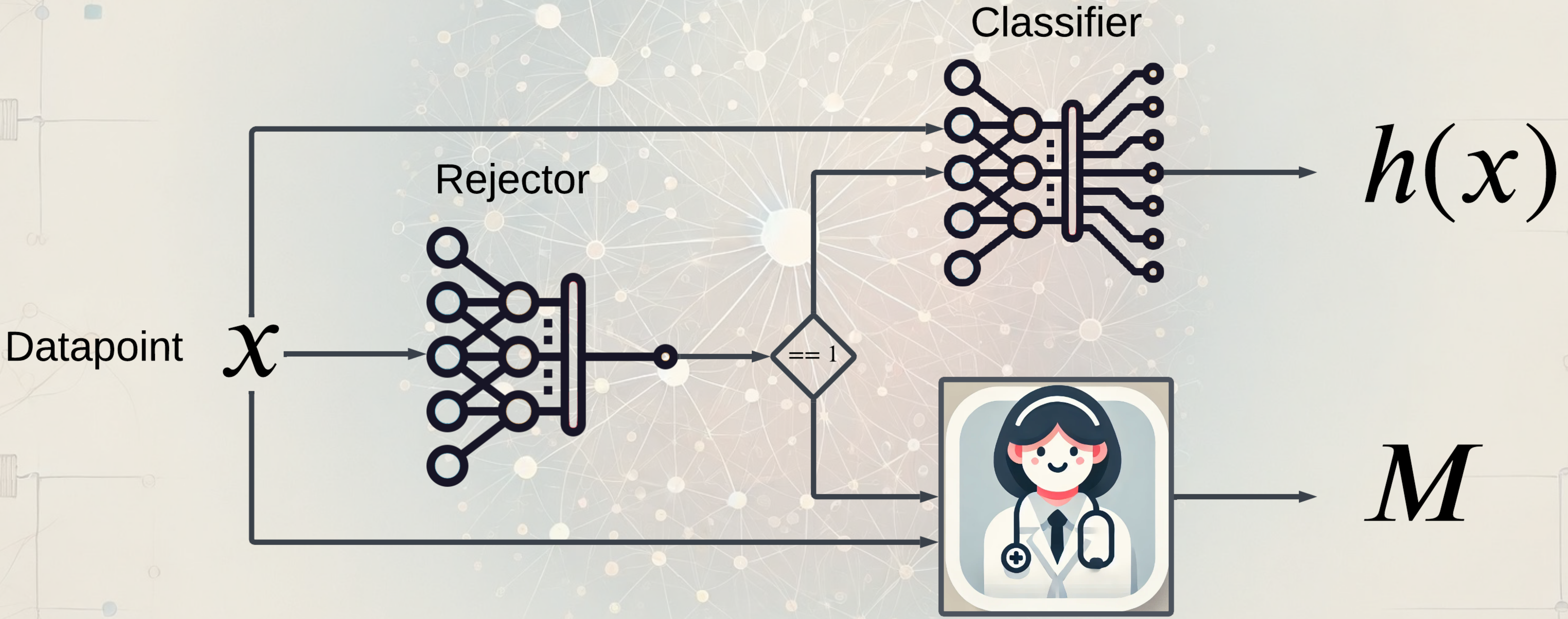
Optimal Multi-Objective Learn-to-Defer:
Possibility, Complexity, and a Post-Processing Framework

Amin Charusaie

January 2025

Learn-to-Defer (L2D) Problem



Optimal L2D

Optimal L2D

- Deferral loss

$$L_{\text{def}}^{0-1}(h, r) = \mathbb{E}\left[\mathbb{1}_{r(X)=0}\mathbb{1}_{h(X)\neq Y} + \mathbb{1}_{r(X)=1}\mathbb{1}_{M\neq Y}\right]$$

Optimal L2D

- Deferral loss

$$L_{\text{def}}^{0-1}(h, r) = \mathbb{E}\left[\mathbb{1}_{r(X)=0}\mathbb{1}_{h(X)\neq Y} + \mathbb{1}_{r(X)=1}\mathbb{1}_{M\neq Y}\right]$$

- Classifier $h(x)$

Optimal L2D

- Deferral loss

$$L_{\text{def}}^{0-1}(h, r) = \mathbb{E}\left[\mathbb{1}_{r(X)=0}\mathbb{1}_{h(X)\neq Y} + \mathbb{1}_{r(X)=1}\mathbb{1}_{M\neq Y}\right]$$

- Classifier $h(x)$
- Rejector $r(x)$

Optimal L2D

- Deferral loss

$$L_{\text{def}}^{0-1}(h, r) = \mathbb{E}\left[\mathbb{1}_{r(X)=0}\mathbb{1}_{h(X)\neq Y} + \mathbb{1}_{r(X)=1}\mathbb{1}_{M\neq Y}\right]$$

- Classifier $h(x)$
- Rejector $r(x)$
- Features ***X***, labels ***Y*** and human decisions ***M***

Optimal L2D

- Deferral loss

$$L_{\text{def}}^{0-1}(h, r) = \mathbb{E}\left[\mathbb{I}_{r(X)=0}\mathbb{I}_{h(X)\neq Y} + \mathbb{I}_{r(X)=1}\mathbb{I}_{M\neq Y}\right]$$

- Classifier $h(x)$
- Rejector $r(x)$
- Features ***X***, labels ***Y*** and human decisions ***M***
- **Constrained L2D:** $\inf_{h,r\in\mathcal{H}\times\mathcal{R}} \mathbb{E}_{X,Y}[\ell_{\text{def}}(h(X), r(X), Y, M)]$ subjected to $\mathbb{E}_{X,Y}[\ell_{\text{c}}(h(X), r(X), X, Y, M)] \leq \delta$

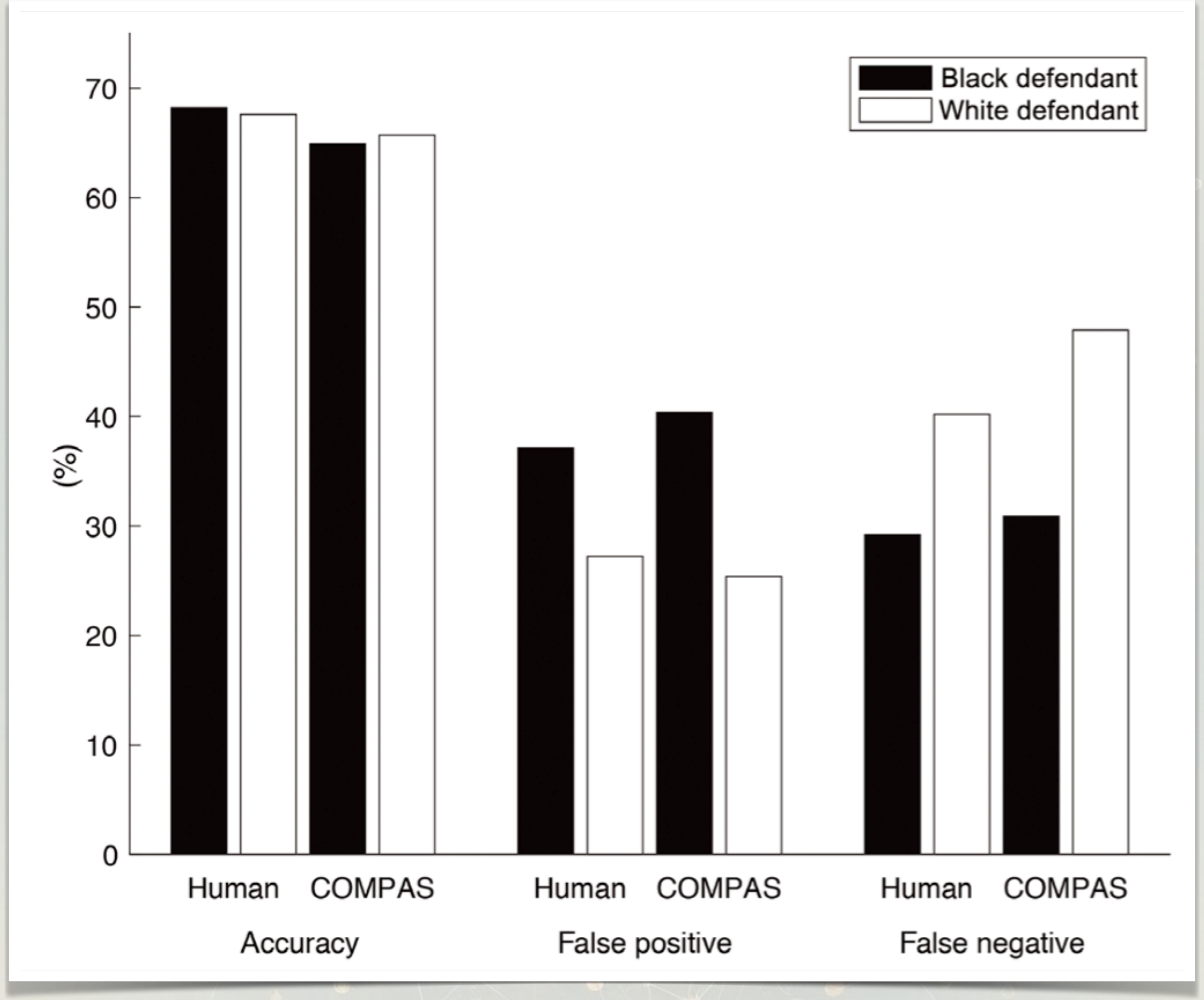
Optimal L2D

- Deferral loss

$$L_{\text{def}}^{0-1}(h, r) = \mathbb{E}\left[\mathbb{1}_{r(X)=0}\mathbb{1}_{h(X)\neq Y} + \mathbb{1}_{r(X)=1}\mathbb{1}_{M\neq Y}\right]$$

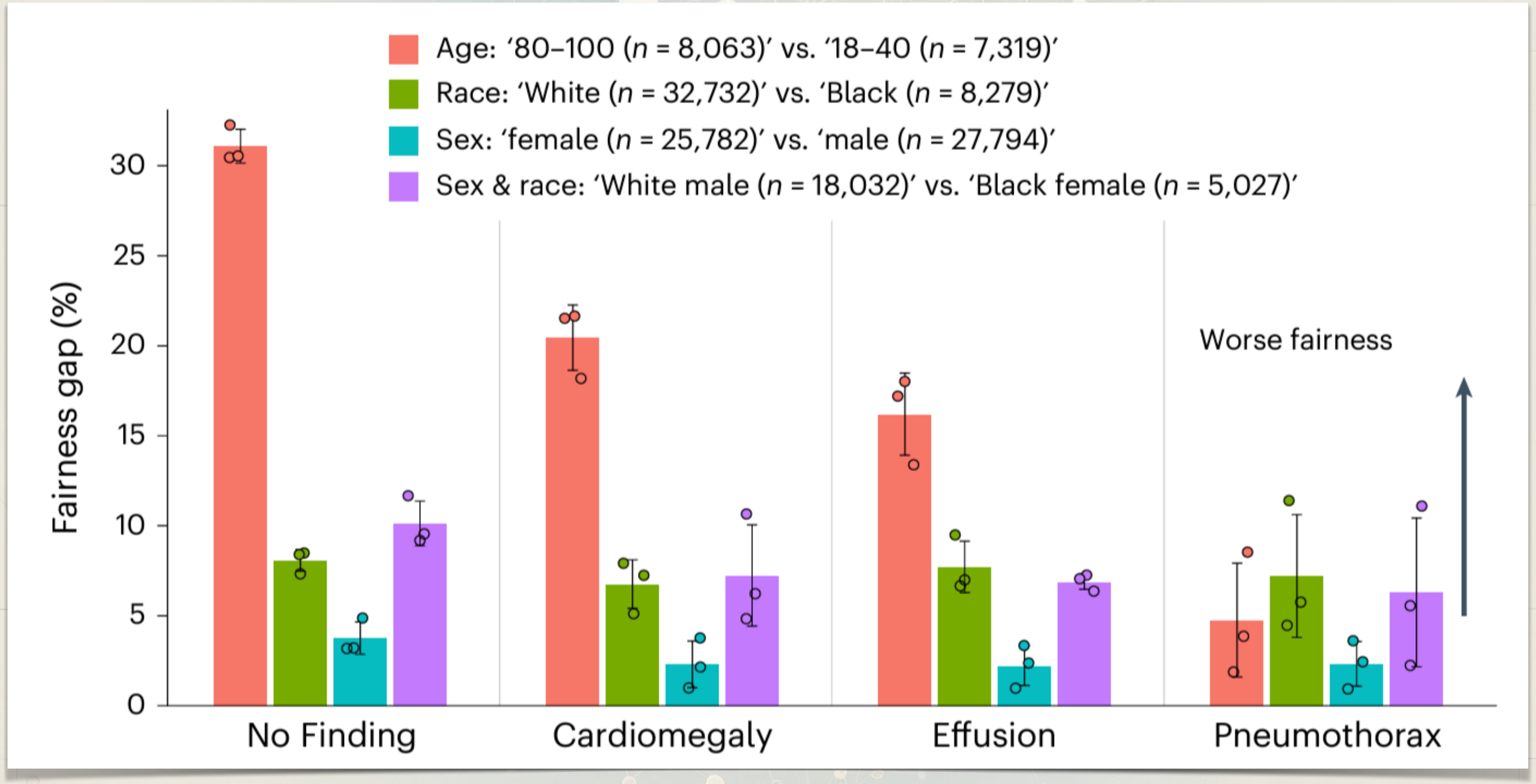
- Classifier $h(x)$
- Rejector $r(x)$
- Features ***X***, labels ***Y*** and human decisions ***M***
- **Constrained L2D:** $\inf_{h,r\in\mathcal{H}\times\mathcal{R}} \mathbb{E}_{X,Y}[\ell_{\text{def}}(h(X), r(X), Y, M)]$ subjected to $\mathbb{E}_{X,Y}[\ell_{\text{c}}(h(X), r(X), X, Y, M)] \leq \delta$
    - Outcome-dependent losses

Algorithmic and Human Bias: COMPAS



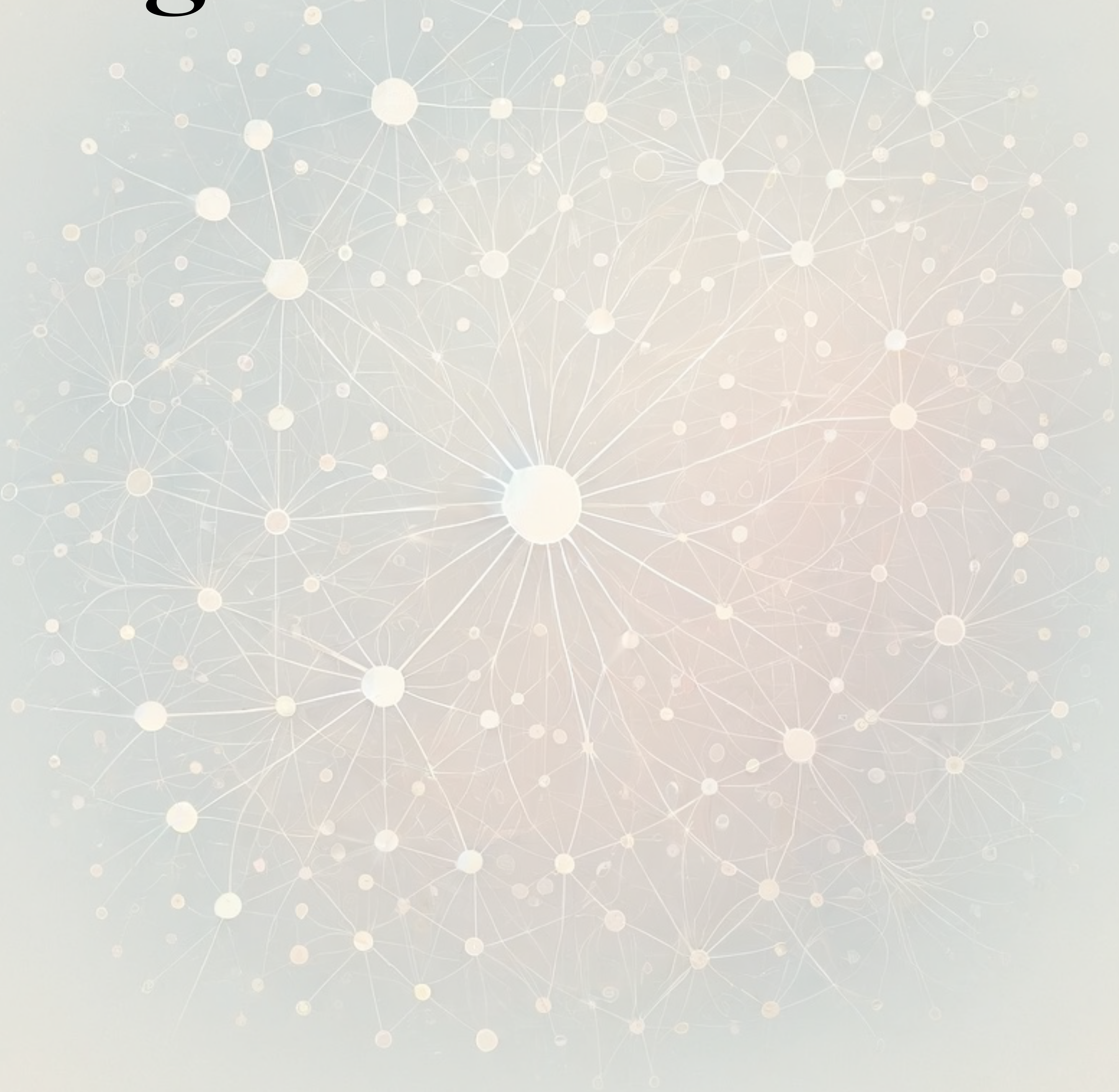
[Dressel et al., Science Advances 2018]

Algorithmic and Human Bias: CheXpert



[Yang et al., Nature Medicine 2024]

Algorithmic Fairness



Algorithmic Fairness

- Demographic Parity (DP): Independence of positive prediction from the sensitive attribute

Algorithmic Fairness

- Demographic Parity (DP): Independence of positive prediction from the sensitive attribute
- Equality of Opportunity (EOp): Independence of false negative from the sensitive attribute

Algorithmic Fairness

- Demographic Parity (DP): Independence of positive prediction from the sensitive attribute
- Equality of Opportunity (EOp): Independence of false negative from the sensitive attribute
- Equalized Odds (EO): Independence of error from the sensitive attribute

Compositionality

We cannot infer independence of a pair of attributes within a sub-universe from the fact of independence within the universe at large. But the converse theorem is also true; a pair of attributes does not necessarily exhibit independence within the universe at large even if it exhibit independence in every sub-universe.

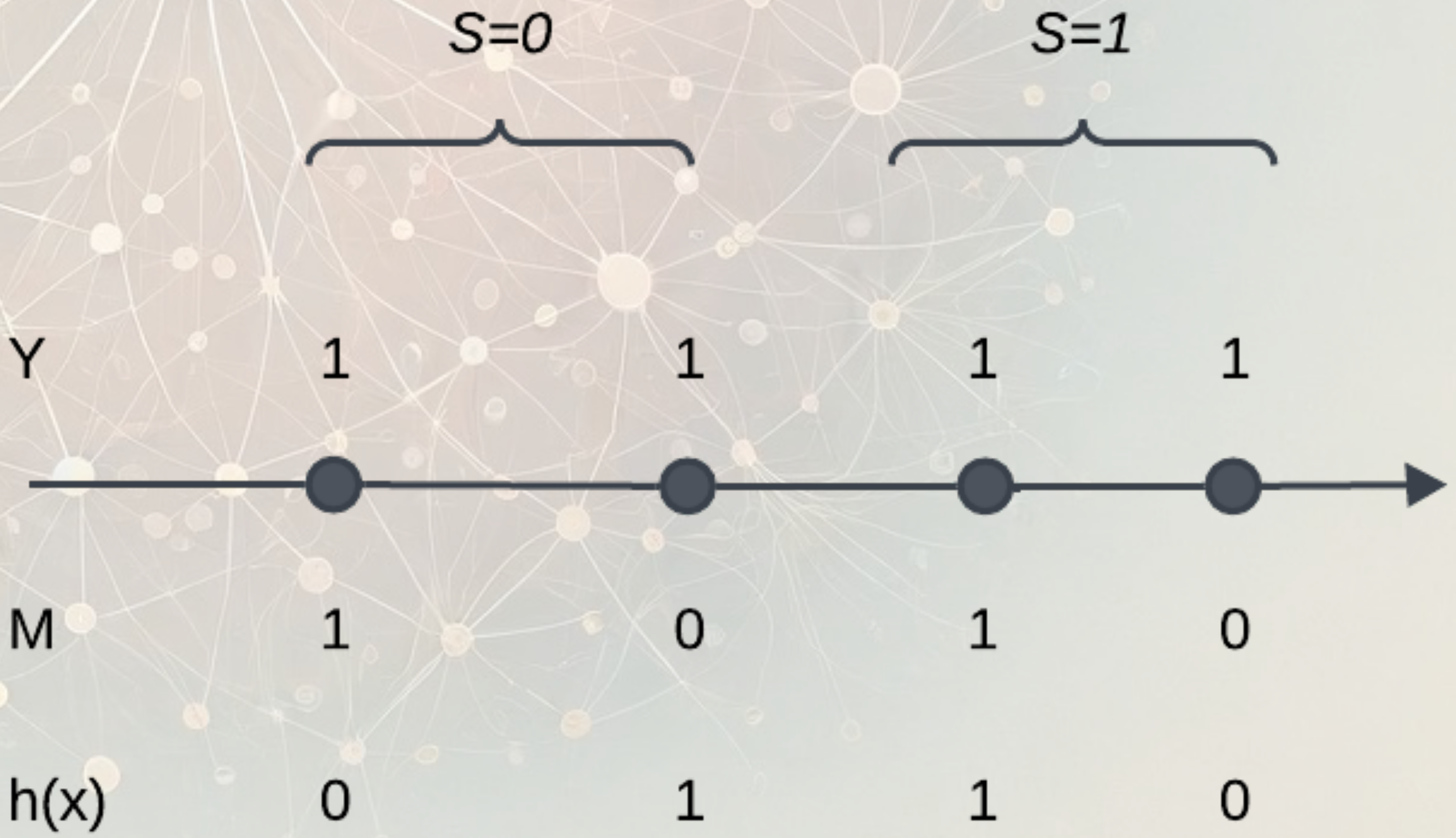
- Udny Yule

Notes on the Theory of Association of Attributes in Statistics 1903

Compositionality

> *We cannot infer independence of a pair of attributes within a sub-universe from the fact of independence within the universe at large. But the converse theorem is also true; a pair of attributes does not necessarily exhibit independence within the universe at large even if it exhibit independence in every sub-universe.*
>
> \- Udny Yule
>
> Notes on the Theory of Association of Attributes in Statistics 1903

L2D Example



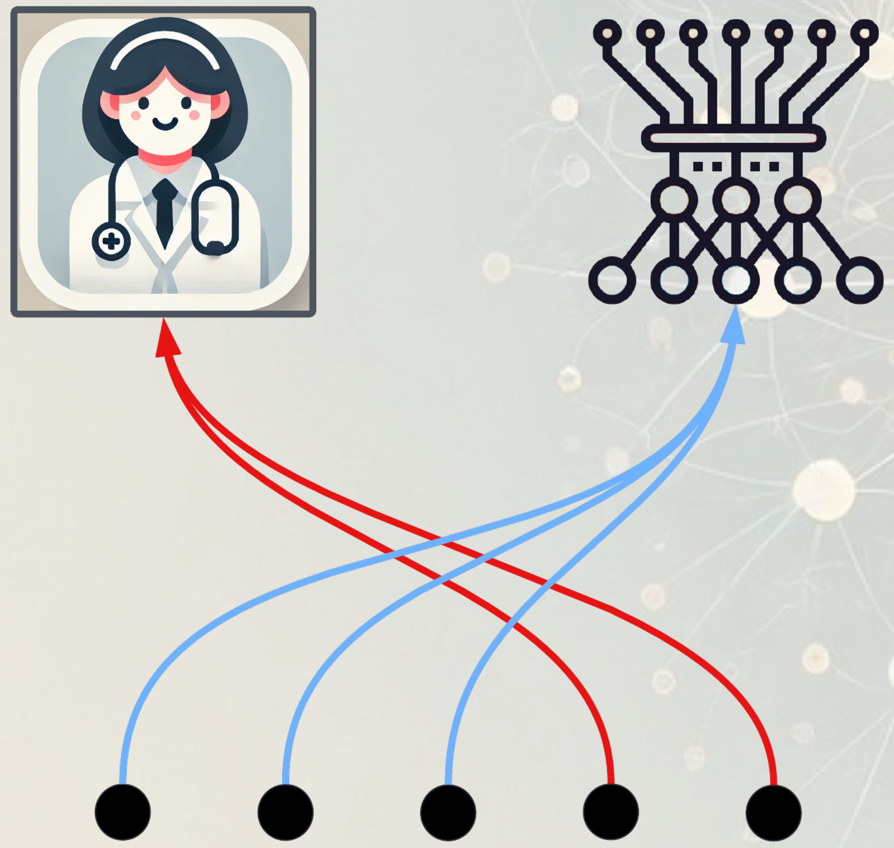
	S=0		S=1	
Y	1	1	1	1
M	1	0	1	0
h(x)	0	1	1	0

Complexity

Theorem: Let the human expert and the classifier induce $0-1$ losses and assume $\mathcal{X}$ to be finite. Finding an optimal deterministic classifier and rejection function for a bounded expert intervention budget is an NP-Hard problem.

Complexity

Theorem: Let the human expert and the classifier induce 0 – 1 losses and assume $\mathscr{X}$ to be finite. Finding an optimal deterministic classifier and rejection function for a bounded expert intervention budget is an NP-Hard problem.

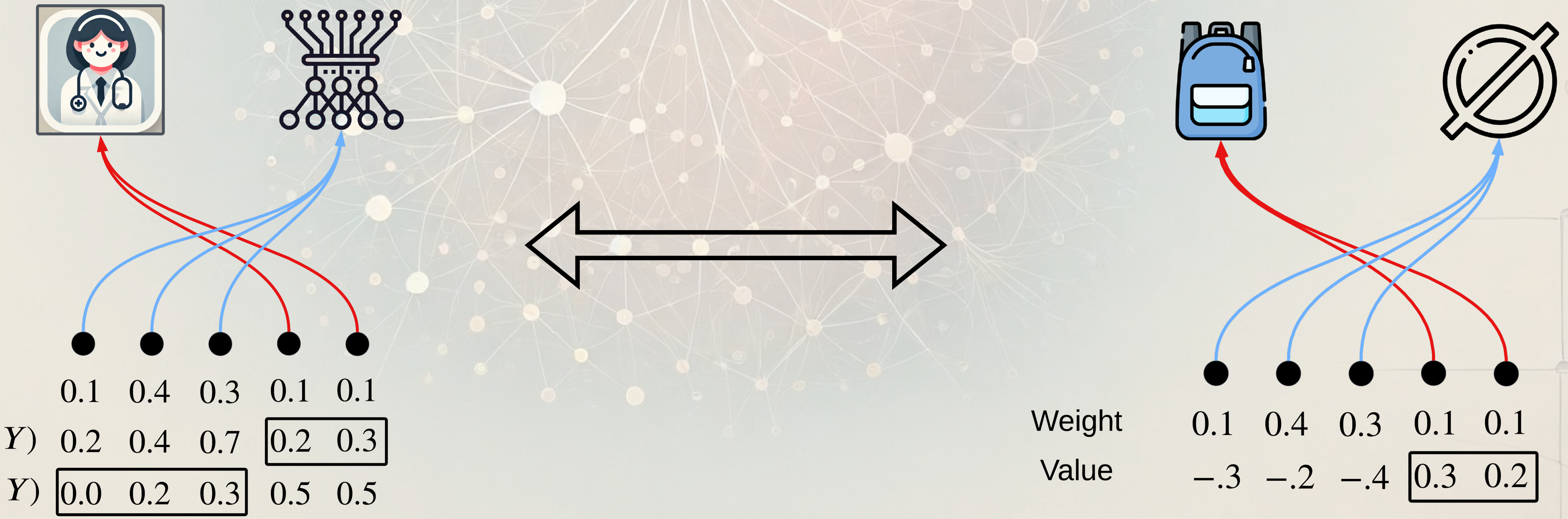


$p(x)$	0.1	0.4	0.3	0.1	0.1
$L_H(M,Y)$	0.2	0.4	0.7	0.2	0.3
$L(h(x),Y)$	0.0	0.2	0.3	0.5	0.5

Complexity

Theorem: Let the human expert and the classifier induce 0 – 1 losses and assume $\mathscr{X}$ to be finite. Finding an optimal deterministic classifier and rejection function for a bounded expert intervention budget is an NP-Hard problem.

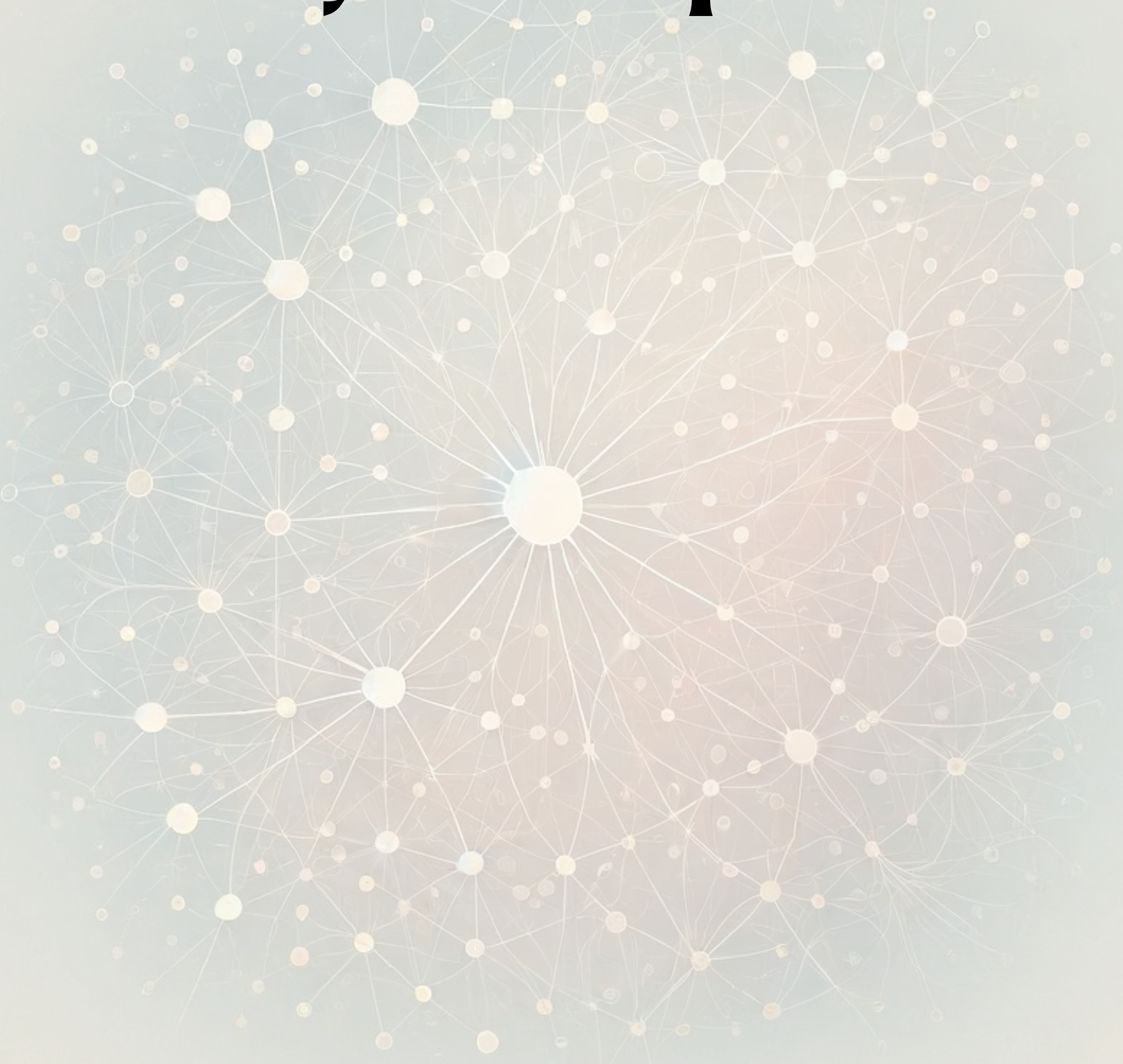
Knapsack Problem



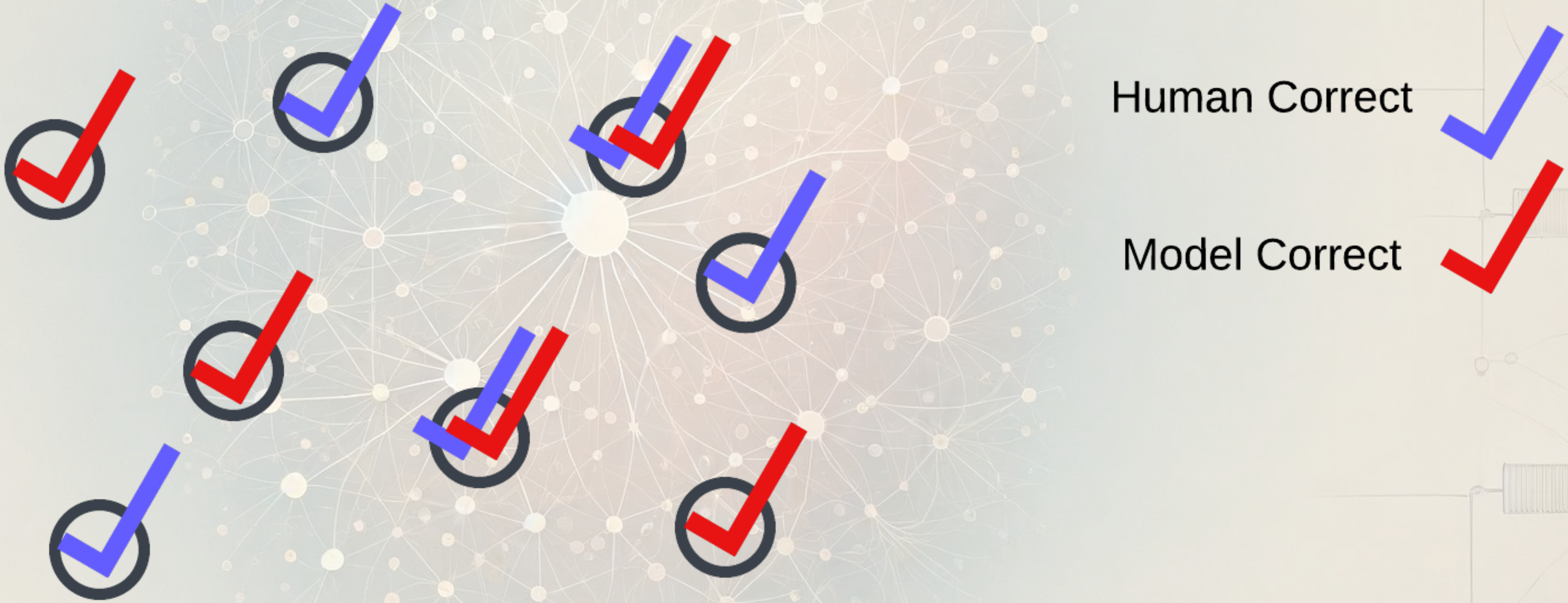
$p(x)$	0.1	0.4	0.3	0.1	0.1
$L_H(M, Y)$	0.2	0.4	0.7	0.2	0.3
$L(h(x), Y)$	0.0	0.2	0.3	0.5	0.5

Weight	0.1	0.4	0.3	0.1	0.1
Value	−.3	−.2	−.4	0.3	0.2

(Im)Possibility of Empirical Solution



(Im)Possibility of Empirical Solution

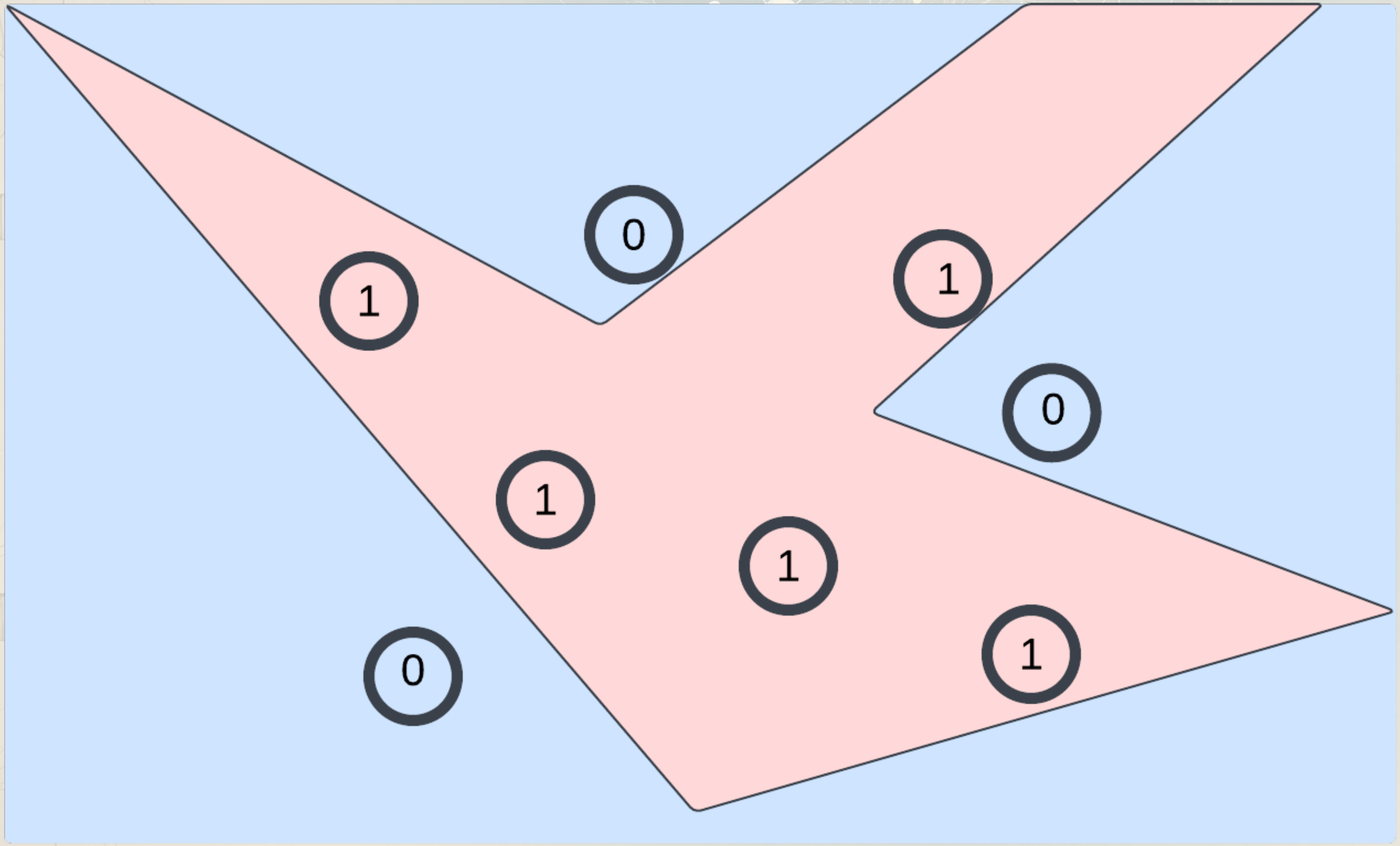


(Im)Possibility of Empirical Solution

Human Correct ✓

Model Correct ✓

(Im)Possibility of Empirical Solution



Human Correct ✓

Model Correct ✓

(Im)Possibility of Empirical Solution

Proposition: For every deterministic deferral rule $\hat{r}$ for empirical distributions and based on the two losses $\mathbb{I}_{m=y}$ and $\mathbb{I}_{h(x)=y}$, there exist two probability measures μ_1 and μ_2 on $\mathscr{X} \times \mathscr{Y} \times \mathscr{M}$ such that the corresponding $(\hat{r}, X)$ for both measures is identically distributed. However, the optimal deferral $r^*_{\mu_1}$and $r^*_{\mu_2}$ for these measures are not interchangeable, that is $L^{\mu_i}_{def}(h, r^*_{\mu_i}) \leq \frac{1}{3}$ while $L^{\mu_i}_{def}(h, r^*_{\mu_j}) = \frac{2}{3}$ for $i = 1{,}2$ and $j \neq i$.

Post-Processing: A Paradigm in ML

Post-Processing: A Paradigm in ML

- Classification: How to solve $\inf_{h \in \mathscr{H}} \mathbb{E}_{X,Y}[\ell(h(X), Y)]$?

Post-Processing: A Paradigm in ML

- Classification: How to solve $\inf_{h \in \mathcal{H}} \mathbb{E}_{X,Y}[\ell(h(X), Y)]$?
    - A1 (e.g., Kernel-SVM): Find $\inf_{f \in \mathcal{F}} \mathbb{E}[\Phi(f(X), Y)]$ for a surrogate function Φ and distance function f

Post-Processing: A Paradigm in ML

- Classification: How to solve $\inf_{h \in \mathcal{H}} \mathbb{E}_{X,Y}[\ell(h(X), Y)]$?
    - A1 (e.g., Kernel-SVM): Find $\inf_{f \in \mathcal{F}} \mathbb{E}[\Phi(f(X), Y)]$ for a surrogate function Φ and distance function f
    - A2 (e.g., Logistic Regression, NNs): Find scores $\boldsymbol{s}^K(x) = [s_1(x), \ldots, s_K(x)]$ related to the loss $\mathbb{E}[\ell(\cdot, \cdot)]$ and find the maximizer

Constrained Classification

Constrained Classification

- $\inf_{h \in \mathscr{H}} \mathbb{E}_{X,Y}[\ell(h(X), Y)]$ subjected to $\mathbb{E}_{X,Y}[\ell_{\mathrm{c}}(h(X), Y)] \leq \delta$

Constrained Classification

- $\inf_{h \in \mathscr{H}} \mathbb{E}_{X,Y}[\ell(h(X), Y)]$ subjected to $\mathbb{E}_{X,Y}[\ell_{\mathrm{c}}(h(X), Y)] \leq \delta$
    - Regularization Method: Find $\inf_{f \in \mathscr{F}} \mathbb{E}_{X,Y}[\Phi(f(X), Y)] + k\mathbb{E}_{X,Y}[\Phi_{\mathrm{c}}(f(X), Y)]$ for a variety of k and for a distance function or score f

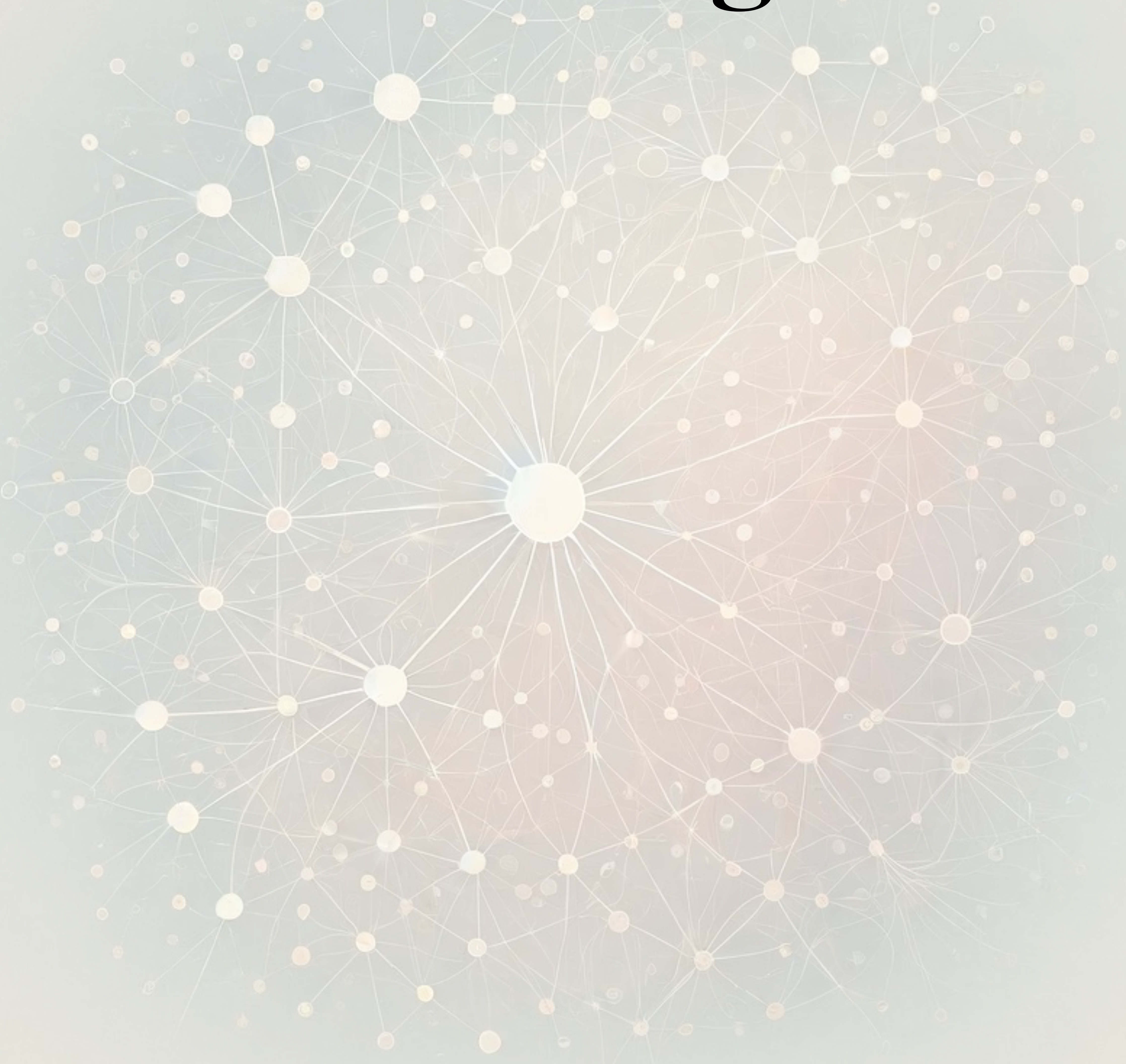
Constrained Classification

- $\inf_{h \in \mathscr{H}} \mathbb{E}_{X,Y}[\ell(h(X), Y)]$ subjected to $\mathbb{E}_{X,Y}[\ell_{\mathrm{c}}(h(X), Y)] \leq \delta$
  - Regularization Method: Find $\inf_{f \in \mathscr{F}} \mathbb{E}_{X,Y}[\Phi(f(X), Y)] + k\mathbb{E}_{X,Y}[\Phi_{\mathrm{c}}(f(X), Y)]$ for a variety of k and for a distance function or score f
  - Post-processing: (e.g., Hardt et al. 2017, Cruz et al. 2023): Find scores s^K related to the loss $\mathbb{E}[\ell(\cdot, \cdot)]$ and threshold differently based on features

Constrained Classification

- $\inf_{h \in \mathscr{H}} \mathbb{E}_{X,Y}[\ell(h(X), Y)]$ subjected to $\mathbb{E}_{X,Y}[\ell_{\mathrm{c}}(h(X), Y)] \leq \delta$
  - Regularization Method: Find $\inf_{f \in \mathscr{F}} \mathbb{E}_{X,Y}[\Phi(f(X), Y)] + k\mathbb{E}_{X,Y}[\Phi_{\mathrm{c}}(f(X), Y)]$ for a variety of k and for a distance function or score f
  - Post-processing: (e.g., Hardt et al. 2017, Cruz et al. 2023): Find scores $\boldsymbol{s}^K$ related to the loss $\mathbb{E}[\ell(\,\cdot\,,\,\cdot\,)]$ and threshold differently based on features
    - Not studied for all types of constraints

Randomized Algorithms



Randomized Algorithms

- $$\begin{aligned} \mu_{\mathscr{A}} \in \operatorname{argmin}_{\mu_{\mathscr{A}}} & \mathbb{E}_{(h,r)\sim\mathscr{A}}\left[\mathbb{E}_{X,Y,M\sim\mu}\left[\ell_{\mathrm{def}}(Y,M,h(X),r(X))\right]\right] \\ \text{s.t.} \quad & \mathbb{E}_{(h,r)\sim\mathscr{A}}\mathbb{E}_{X,Y,M\sim\mu}\left[\Psi_i\left(X,Y,M,h(X),r(X)\right)\right] \leq \delta_i, \end{aligned}$$

Randomized Algorithms

- $$\begin{aligned} \mu_{\mathscr{A}} \in \operatorname{argmin}_{\mu_{\mathscr{A}}} & \mathbb{E}_{(h,r)\sim\mathscr{A}}\left[\mathbb{E}_{X,Y,M\sim\mu}\left[\ell_{\text{def}}(Y, M, h(X), r(X))\right]\right] \\ \text{s.t.} \quad & \mathbb{E}_{(h,r)\sim\mathscr{A}}\mathbb{E}_{X,Y,M\sim\mu}\left[\Psi_i\big(X, Y, M, h(X), r(X)\big)\right] \leq \delta_i, \end{aligned}$$
- $K + 1$ combinations of $h(x)$ and $r(x)$

Randomized Algorithms

- $$\begin{aligned} \mu_{\mathscr{A}} \in \operatorname{argmin}_{\mu_{\mathscr{A}}} & \mathbb{E}_{(h,r)\sim\mathscr{A}}\left[\mathbb{E}_{X,Y,M\sim\mu}\left[\ell_{\text{def}}(Y, M, h(X), r(X))\right]\right] \\ \text{s.t.} \quad & \mathbb{E}_{(h,r)\sim\mathscr{A}}\mathbb{E}_{X,Y,M\sim\mu}\left[\Psi_i\left(X, Y, M, h(X), r(X)\right)\right] \leq \delta_i, \end{aligned}$$

- $K + 1$ combinations of $h(x)$ and $r(x)$

- $\mu_{\mathscr{A}}$ induces a probability $f_i(x)$ over the i-th choice

Randomized Algorithms

- $$\begin{aligned} \mu_{\mathscr{A}} \in \operatorname{argmin}_{\mu_{\mathscr{A}}} & \mathbb{E}_{(h,r)\sim\mathscr{A}}\left[\mathbb{E}_{X,Y,M\sim\mu}\left[\ell_{\text{def}}(Y,M,h(X),r(X))\right]\right] \\ \text{s.t.} \quad & \mathbb{E}_{(h,r)\sim\mathscr{A}}\mathbb{E}_{X,Y,M\sim\mu}\left[\Psi_i\big(X,Y,M,h(X),r(X)\big)\right] \leq \delta_i, \end{aligned}$$
- $K+1$ combinations of $h(x)$ and $r(x)$
- $\mu_{\mathscr{A}}$ induces a probability $f_i(x)$ over the i-th choice
- Linear Functional Programming:

Randomized Algorithms

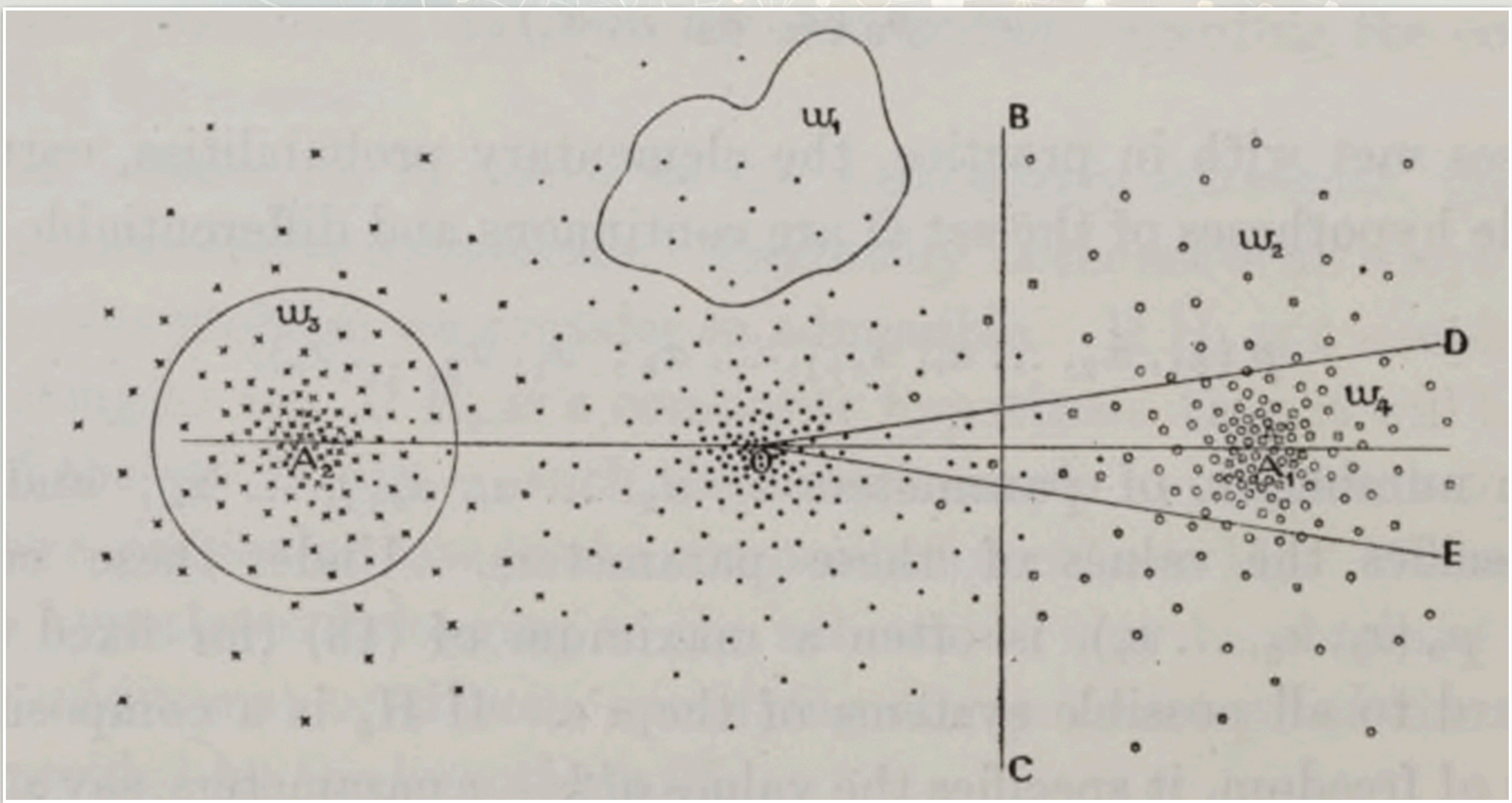
- $$\begin{aligned} \mu_{\mathscr{A}} \in \operatorname{argmin}_{\mu_{\mathscr{A}}} & \mathbb{E}_{(h,r)\sim\mathscr{A}}\left[\mathbb{E}_{X,Y,M\sim\mu}\left[\ell_{\text{def}}(Y,M,h(X),r(X))\right]\right] \\ \text{s.t.} \quad & \mathbb{E}_{(h,r)\sim\mathscr{A}}\mathbb{E}_{X,Y,M\sim\mu}\left[\Psi_i\left(X,Y,M,h(X),r(X)\right)\right] \le \delta_i, \end{aligned}$$
- $K+1$ combinations of $h(x)$ and $r(x)$
- $\mu_{\mathscr{A}}$ induces a probability $f_i(x)$ over the i-th choice
- Linear Functional Programming:
  - $f^* = [f_1^*, \ldots, f_{K+1}^*] \in \operatorname{argmax}_{f \in \Delta_{K+1}^{\mathscr{X}}} \mathbb{E}_X\left[\langle f(X), \psi_{m+1}(X)\rangle\right]$
    s.t. $\quad \mathbb{E}_X\left[\langle f(x), \psi_i(x)\rangle\right] \le \delta_i$ for $i \in \{1, \ldots, m\}$

Randomized Algorithms

- $$\begin{aligned} \mu_{\mathscr{A}} \in \operatorname{argmin}_{\mu_{\mathscr{A}}} & \mathbb{E}_{(h,r)\sim\mathscr{A}}\left[\mathbb{E}_{X,Y,M\sim\mu}\left[\ell_{\text{def}}(Y,M,h(X),r(X))\right]\right] \\ \text{s.t.} \quad & \mathbb{E}_{(h,r)\sim\mathscr{A}}\mathbb{E}_{X,Y,M\sim\mu}\left[\Psi_i\big(X,Y,M,h(X),r(X)\big)\right] \le \delta_i, \end{aligned}$$
- $K + 1$ combinations of $h(x)$ and $r(x)$
- $\mu_{\mathscr{A}}$ induces a probability $f_i(x)$ over the i-th choice
- Linear Functional Programming:
  - $f^* = [f_1^*, \ldots, f_{K+1}^*] \in \operatorname{argmax}_{f\in\Delta_{K+1}^{\mathcal{X}}} \mathbb{E}_X\left[\langle f(X), \psi_{m+1}(X)\rangle\right]$
    s.t. $\quad \mathbb{E}_X\left[\langle f(x), \psi_i(x)\rangle\right] \le \delta_i$ for $i \in \{1, \ldots, m\}$
- Similarly for constrained classification

Hypothesis Testing

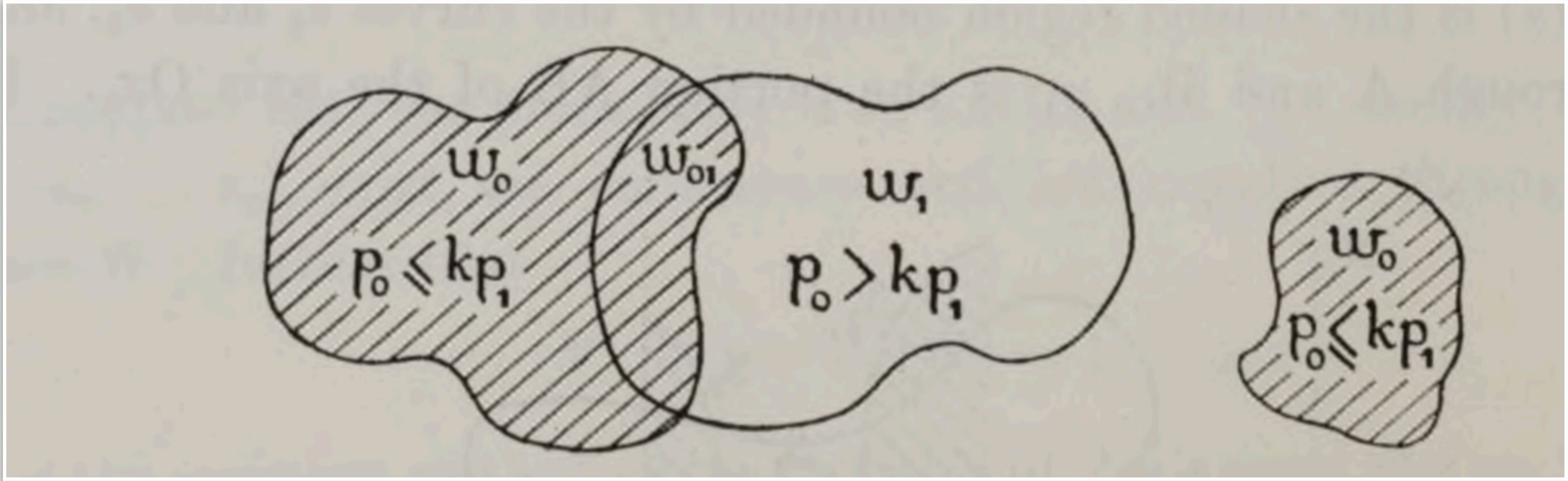
Hypothesis Testing



Neyman and Pearson 1933

Hypothesis Testing

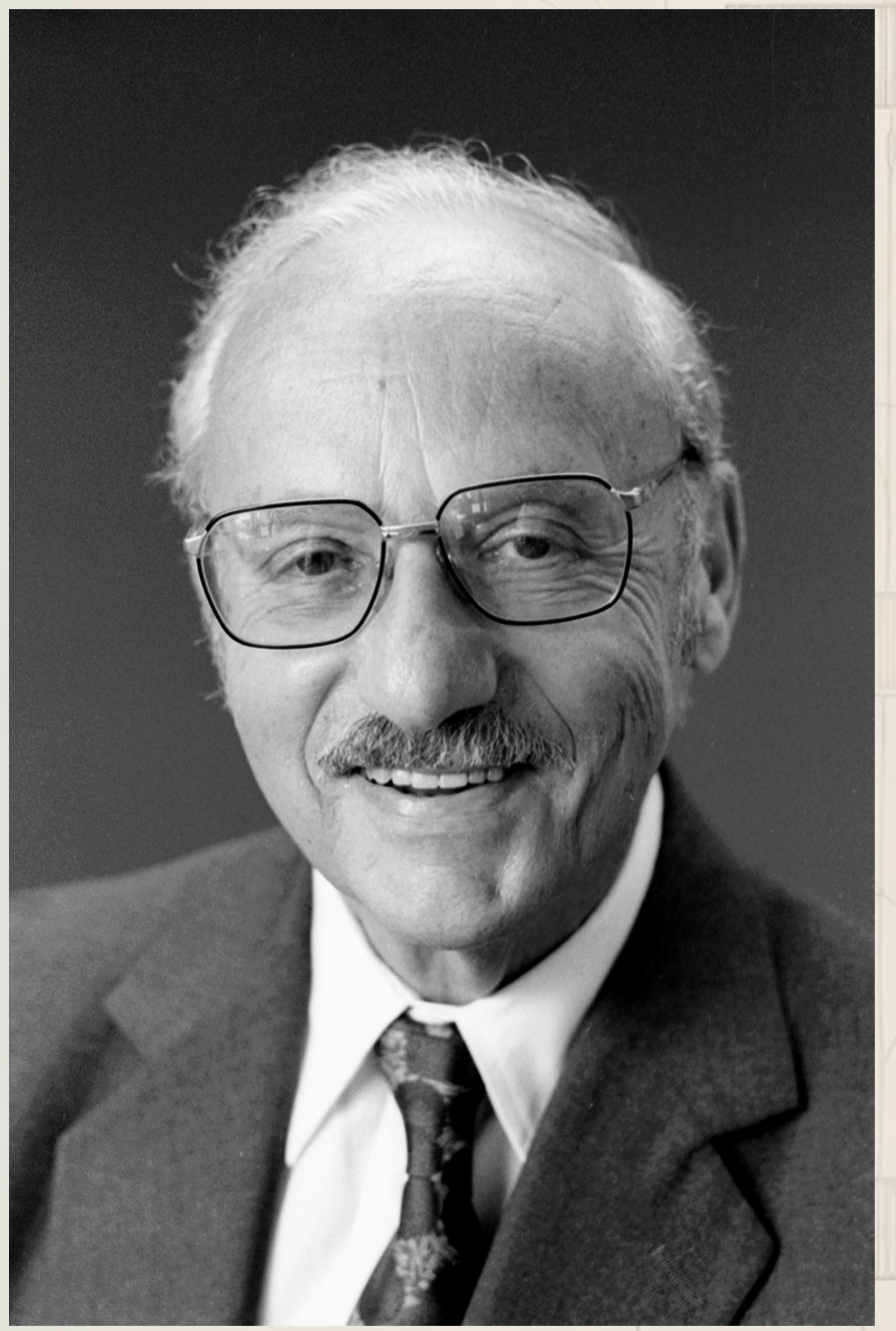
Hypothesis Testing



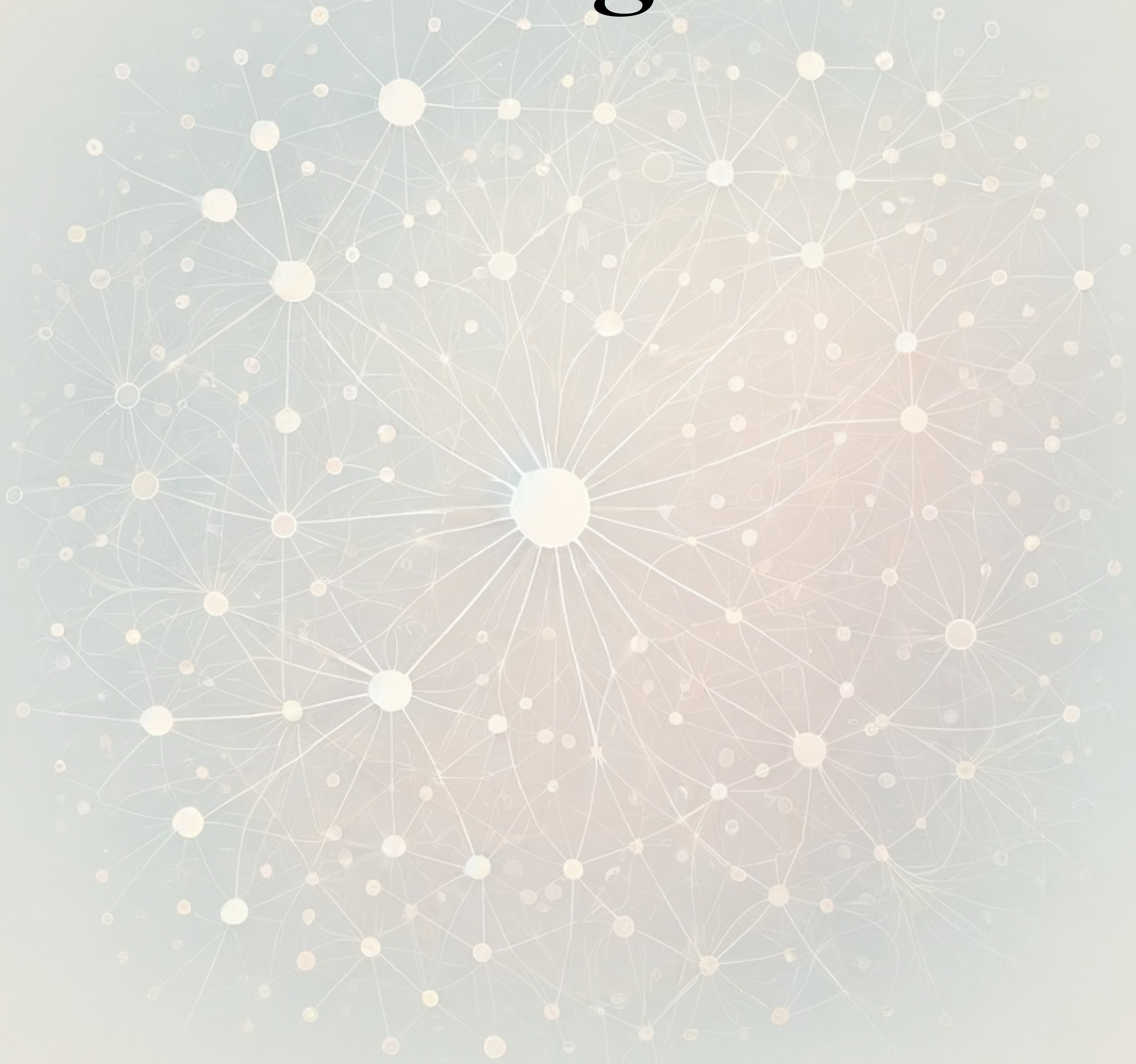
Neyman and Pearson 1933

Hypothesis Testing

1. Does *k* always exist?
2. Is there any other optimal solution?



Hypothesis Testing: A Formal Take



Hypothesis Testing: A Formal Take

- One-sample test: whether X is drawn from a distribution $H_0 : P$ or not

Hypothesis Testing: A Formal Take

- One-sample test: whether X is drawn from a distribution $H_0 : P$ or not
- Known alternative: $H_a : Q$

Hypothesis Testing: A Formal Take

- One-sample test: whether X is drawn from a distribution $H_0 : P$ or not
- Known alternative: $H_a : Q$
- Size (false positive rate) $\mathbb{E}_P[T(X)]$ and power (true negative rate) $\mathbb{E}_Q[T(X)]$

Hypothesis Testing: A Formal Take

- One-sample test: whether X is drawn from a distribution $H_0 : P$ or not
- Known alternative: $H_a : Q$
- Size (false positive rate) $\mathbb{E}_P[T(X)]$ and power (true negative rate) $\mathbb{E}_Q[T(X)]$
- Universally Most Powerful Test (Neyman-Pearson Lemma): Most-powerful test for a size at most α is a likelihood-ratio test, i.e.,

$$T* = \text{argmax}_T \mathbb{E}_Q[T(X)] \text{ s.t. } \mathbb{E}_P[T(X)] \leq \delta$$

where $T*(x) = 1$ where $Q(x) > kP(x)$ and $T*(x) = 0$ where $Q(x) < kP(x)$

Hypothesis Testing: A Formal Take

- One-sample test: whether X is drawn from a distribution $H_0 : P$ or not
- Known alternative: $H_a : Q$
- Size (false positive rate) $\mathbb{E}_P[T(X)]$ and power (true negative rate) $\mathbb{E}_Q[T(X)]$
- Universally Most Powerful Test (Neyman-Pearson Lemma): Most-powerful test for a size at most α is a likelihood-ratio test, i.e.,

  $$T^* = \text{argmax}_T \mathbb{E}_Q[T(X)] \text{ s.t. } \mathbb{E}_P[T(X)] \leq \delta$$

  where $T^*(x) = 1$ where $Q(x) > kP(x)$ and $T^*(x) = 0$ where $Q(x) < kP(x)$
- Generalizations and Applications: Lehmann et al. 2005 (Critical Function), Tian and Feng 2021 (Multiclass), Zeng et al. 2024 (Fairness)

d-dimensional Generalization of Neyman-Pearson Lemma (d-GNP)

d-dimensional Generalization of Neyman-Pearson Lemma (d-GNP)

- $H_1, \ldots, H_d$ where we reject $d - 1$ hypothesis

d-dimensional Generalization of Neyman-Pearson Lemma (d-GNP)

- $H_1, \ldots, H_d$ where we reject $d-1$ hypothesis
- Receive true positive rewards and false negative losses

d-dimensional Generalization of Neyman-Pearson Lemma (d-GNP)

- $H_1, \ldots, H_d$ where we reject $d - 1$ hypothesis
- Receive true positive rewards and false negative losses
- Goal: Maximize sum of rewards, while bounding the sum of losses by α

d-dimensional Generalization of Neyman-Pearson Lemma (d-GNP)

- $H_1, \ldots, H_d$ where we reject $d-1$ hypothesis
- Receive true positive rewards and false negative losses
- Goal: Maximize sum of rewards, while bounding the sum of losses by α
- $f^* = [f_1^*, \ldots, f_d^*] \in \operatorname{argmax}_{f \in \Delta_d^{\mathcal{X}}} \mathbb{E}_X\big[\langle f(X), \psi_2(X) \rangle\big]$
  s.t. $\quad \mathbb{E}_X\big[\langle f(x), \psi_1(x) \rangle\big] \leq \alpha$

d-dimensional Generalization of Neyman-Pearson Lemma (d-GNP)

- $f^* = [f_1^*, \ldots, f_d^*] \in \operatorname{argmax}_{f \in \Delta_d^{\mathcal{X}}} \mathbb{E}_X\left[\langle f(X), \psi_{m+1}(X) \rangle\right]$
  s.t. $\quad \mathbb{E}_X\left[\langle f(x), \psi_i(x) \rangle\right] \leq \alpha_i$

d-dimensional Generalization of Neyman-Pearson Lemma (d-GNP)

- $f^* = [f_1^*, \ldots, f_d^*] \in \mathrm{argmax}_{f \in \Delta_d^{\mathcal{X}}} \mathbb{E}_X\big[\langle f(X), \psi_{m+1}(X)\rangle\big]$
  s.t. $\mathbb{E}_X\big[\langle f(x), \psi_i(x)\rangle\big] \leq \alpha_i$

Theorem (informal): If bounds of constraints are interior-points of all possible pairs of constraints, then $f^*(x) = \mathrm{argmax}_j\Big[\psi_{m+1}(x) - \sum_{i=1}^{m} k_i \psi_i(x)\Big]_j$ when there is a *single maximizer*, and if we know that constraints are achieved tightly. All optimal solutions to the linear functional programming is of form above.

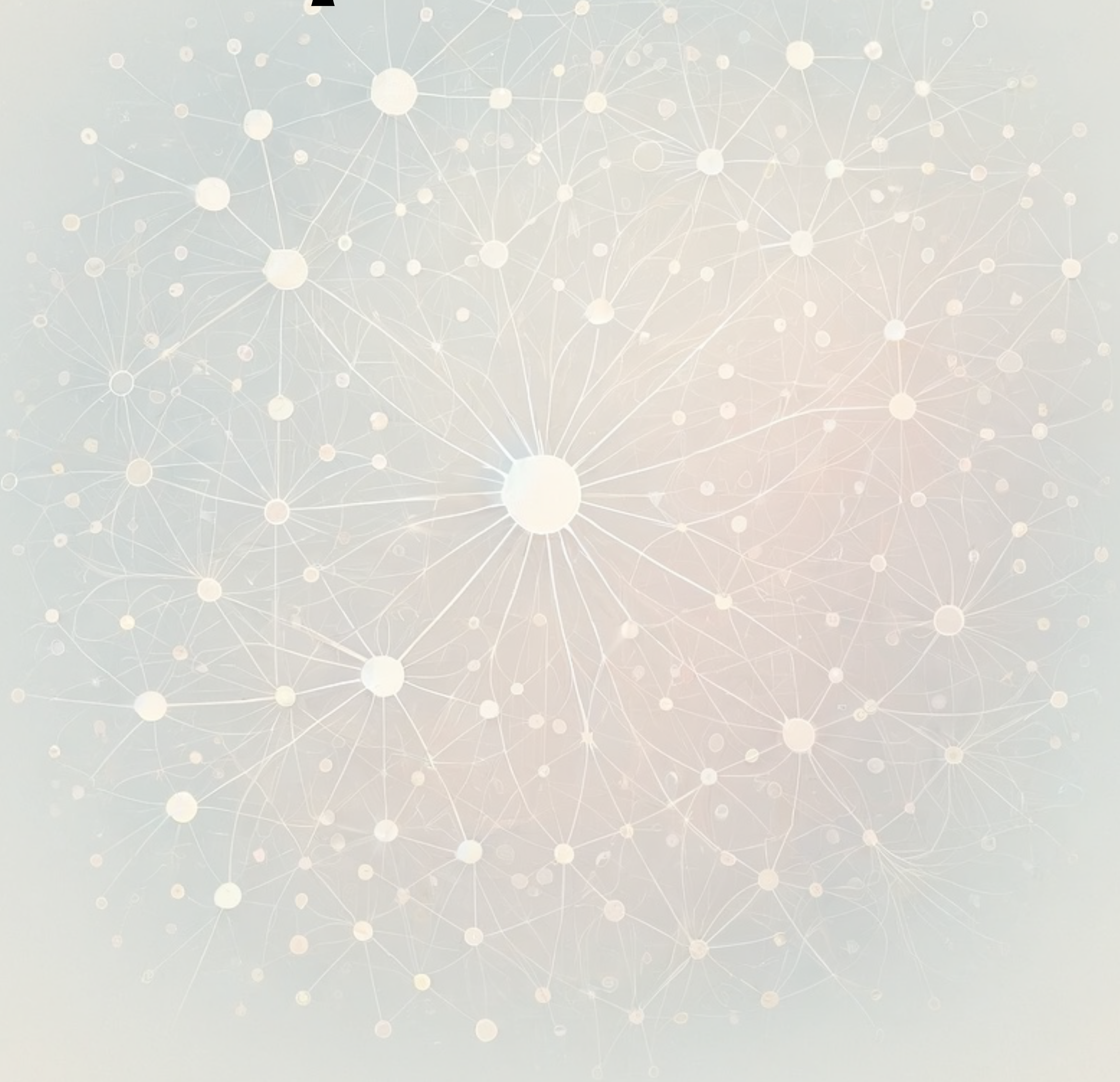
d-dimensional Generalization of Neyman-Pearson Lemma (d-GNP)

- $f^* = [f_1^*, \ldots, f_d^*] \in \text{argmax}_{f \in \Delta_d^{\mathcal{X}}} \mathbb{E}_X\left[\langle f(X), \psi_{m+1}(X)\rangle\right]$
  s.t. $\mathbb{E}_X\left[\langle f(x), \psi_i(x)\rangle\right] \leq \alpha_i$

Theorem (informal): If bounds of constraints are interior-points of all possible pairs of constraints, then $f^*(x) = \text{argmax}_j \left[\psi_{m+1}(x) - \sum_{i=1}^{m} k_i \psi_i(x)\right]_j$ when there is a *single maximizer*, and if we know that constraints are achieved tightly. All optimal solutions to the linear functional programming is of form above.

Theorem (informal): In case of a single constraint, k_1 is the root of a monotone function with known closed-form, and a random predictor is drawn for the cases that we don't have a *single maximizer*

Simplified d-GNP



Simplified d-GNP

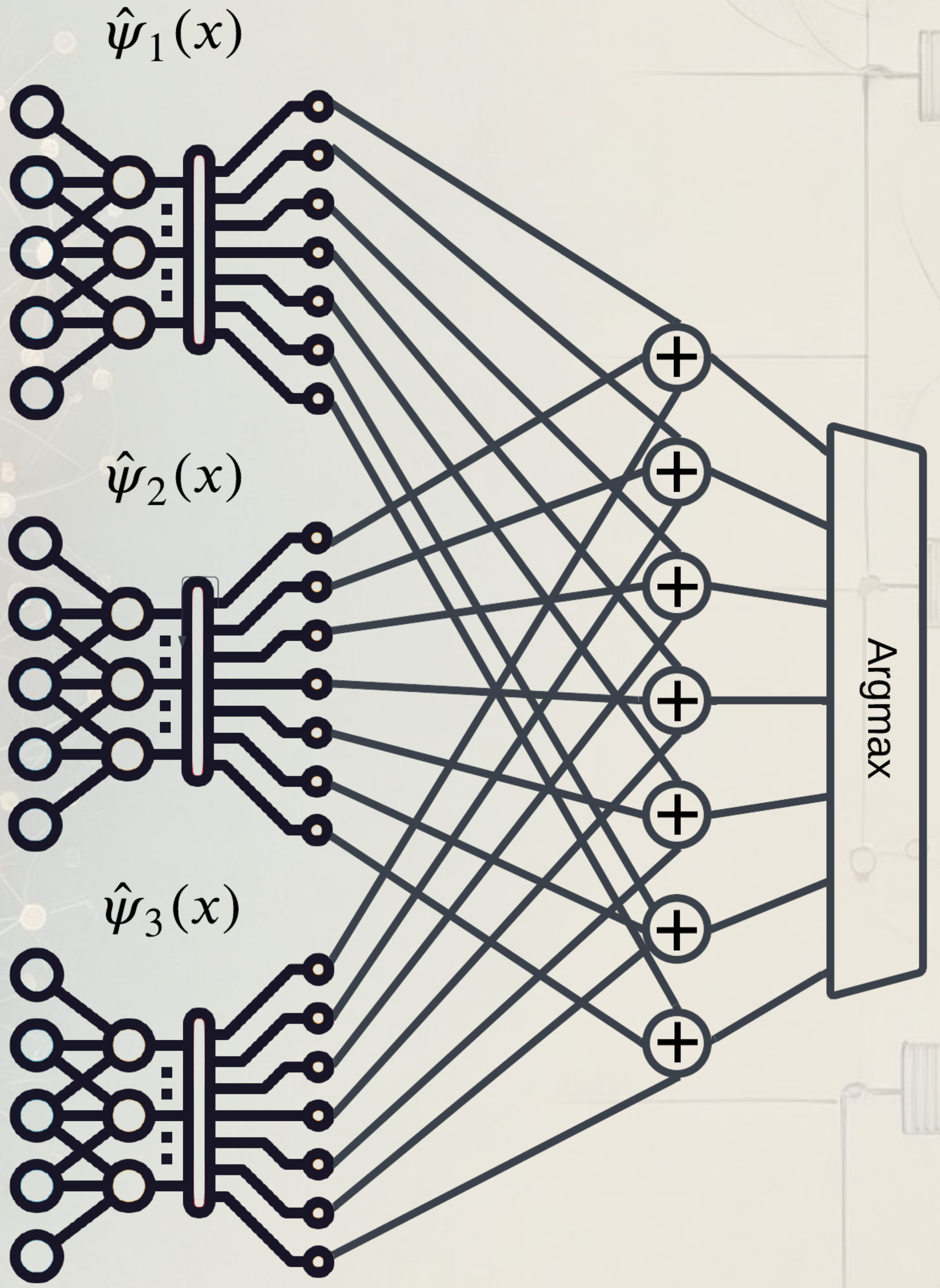
$$
\begin{aligned}
&\mu_{\mathscr{A}} \in \operatorname{argmin}_{\mu_{\mathscr{A}}} \mathbb{E}_{h\sim\mathscr{A}}\left[\mathbb{E}_{X,Y\sim\mu}\left[\Psi_1(Y, h(X))\right]\right] \\
&\text{s.t.} \quad \mathbb{E}_{h\sim\mathscr{A}}\mathbb{E}_{X,Y\sim\mu}\left[\Psi_2\left(X, Y, h(X)\right)\right] \le \delta_2, \\
&\mathbb{E}_{h\sim\mathscr{A}}\mathbb{E}_{X,Y\sim\mu}\left[\Psi_3\left(X, Y, h(X)\right)\right] \le \delta_3,
\end{aligned}
$$

Multi-Objective Learning

Simplified d-GNP

$$\begin{aligned}
&\mu_{\mathscr{A}} \in \operatorname{argmin}_{\mu_{\mathscr{A}}} \mathbb{E}_{h\sim\mathscr{A}}\left[\mathbb{E}_{X,Y\sim\mu}\left[\Psi_1(Y, h(X))\right]\right] \\
&\text{s.t.} \quad \mathbb{E}_{h\sim\mathscr{A}}\mathbb{E}_{X,Y\sim\mu}\left[\Psi_2\left(X, Y, h(X)\right)\right] \le \delta_2, \\
&\mathbb{E}_{h\sim\mathscr{A}}\mathbb{E}_{X,Y\sim\mu}\left[\Psi_3\left(X, Y, h(X)\right)\right] \le \delta_3,
\end{aligned}$$

$$=$$



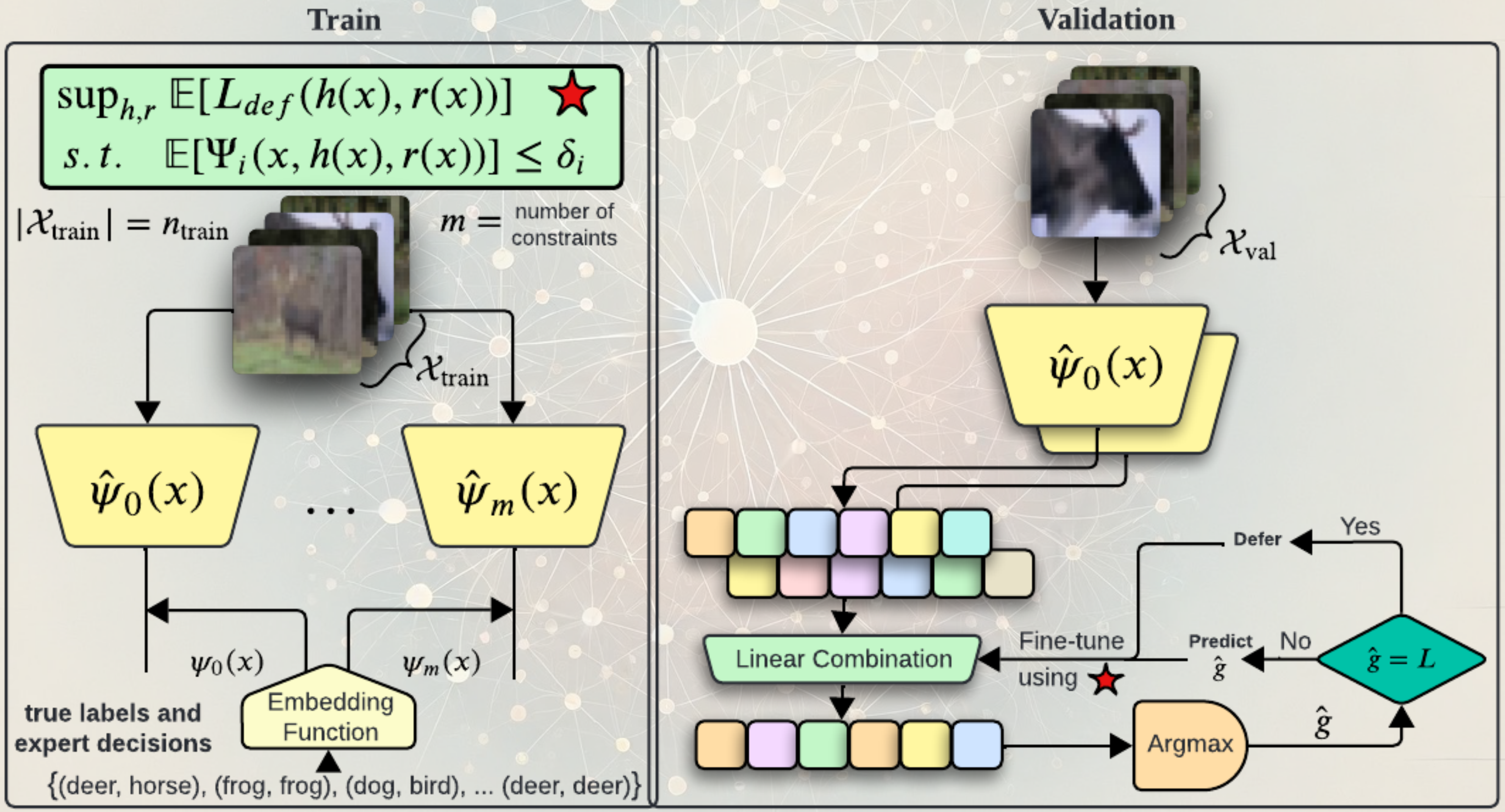
Multi-Objective Learning

Ensembling

Embedding Functions

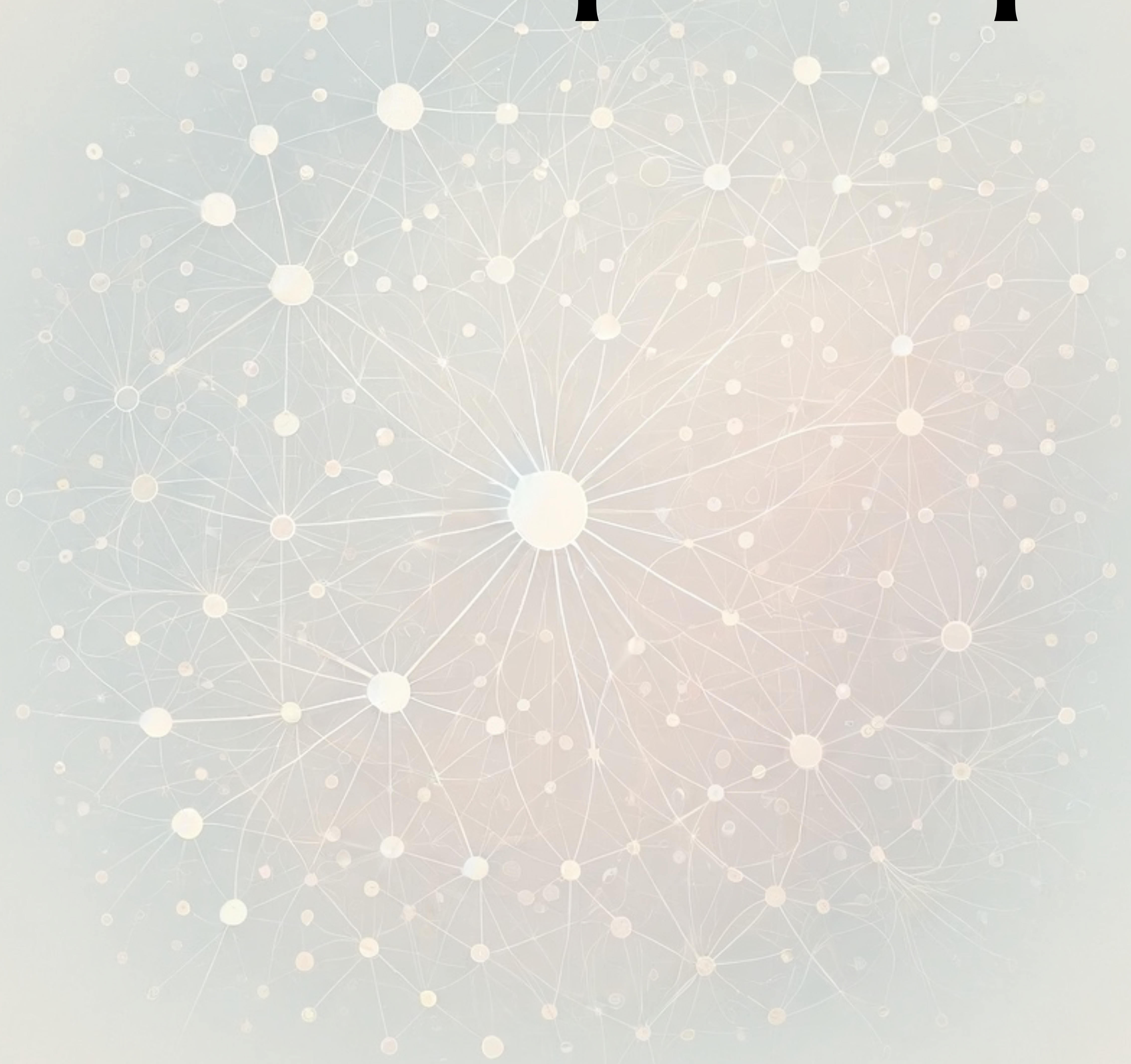
Type of Constraint	Embedding Function $\psi(x)$
Expert Intervention Budget	$[0,\ldots,0,1]$
OOD Detection	$[0,\ldots,0,\frac{f_X^{\text{out}}(x)}{f_X^{\text{in}}(x)}]$
Demographic Parity	$(\frac{\mathbb{1}_{A=1}}{Pr(A=1)}-\frac{\mathbb{1}_{A=0}}{Pr(A=0)})[0,1,\Pr(M=1\,\vert\,x)]$
Equality of Opportunity	$(\frac{\mathbb{1}_{A=1}}{Pr(Y=1,A=1)}-\frac{\mathbb{1}_{A=0}}{Pr(Y=1,A=0)})[0,\Pr(Y=1\,\vert\,x),\Pr(M=1,Y=1\,\vert\,x)]$

d-GNP in Learn-to-Defer



Train
Validation
$\sup_{h,r} \mathbb{E}[L_{def}(h(x), r(x))]$ ★
$s.t. \quad \mathbb{E}[\Psi_i(x, h(x), r(x))] \leq \delta_i$
$|\mathcal{X}_{\text{train}}| = n_{\text{train}}$
$m =$ number of constraints
$\mathcal{X}_{\text{train}}$
$\hat{\psi}_0(x)$
...
$\hat{\psi}_m(x)$
$\psi_0(x)$
$\psi_m(x)$
Embedding Function
true labels and expert decisions
{(deer, horse), (frog, frog), (dog, bird), ... (deer, deer)}
$\mathcal{X}_{\text{val}}$
$\hat{\psi}_0(x)$
Defer
Yes
Linear Combination
Fine-tune using ★
Predict $\hat{g}$
No
$\hat{g} = L$
Argmax
$\hat{g}$

Constraint Sample Complexity



Constraint Sample Complexity

Constraint Statistical Generalization: $O(\sqrt{\log n/n}, \sqrt{\log(1/\epsilon)/n}, \epsilon')$ with probability at least $1 - \epsilon$ and when scores are ϵ'-accurate:

Constraint Sample Complexity

Constraint Statistical Generalization: $O(\sqrt{\log n/n}, \sqrt{\log(1/\epsilon)/n}, \epsilon')$ with probability at least $1-\epsilon$ and when scores are ϵ'-accurate:

1. $\mathbb{E}\left[\langle f(x), \hat{\psi}(x) - \psi(x)\rangle\right] \leq \epsilon'$

Constraint Sample Complexity

Constraint Statistical Generalization: $O(\sqrt{\log n/n}, \sqrt{\log(1/\epsilon)/n}, \epsilon')$ with probability at least $1 - \epsilon$ and when scores are ϵ'-accurate:

1. $\mathbb{E}\left[\langle f(x), \hat{\psi}(x) - \psi(x)\rangle\right] \leq \epsilon'$
2. $\Pr\left(\sup_{k,p} \mathbb{E}_{S^n}\left[\langle f^*_{k,p}(x), \psi(x)\rangle\right] - \mathbb{E}_{\mu}\left[\langle f^*_{k,p}(x), \psi(x)\rangle\right] \leq d_n(\epsilon)\right) \geq 1 - \epsilon$

Constraint Sample Complexity

Constraint Statistical Generalization: $O(\sqrt{\log n/n}, \sqrt{\log(1/\epsilon)/n}, \epsilon')$ with probability at least $1-\epsilon$ and when scores are ϵ'-accurate:

1. $\mathbb{E}\big[\langle f(x), \hat{\psi}(x) - \psi(x)\rangle\big] \le \epsilon'$
2. $\Pr\Big(\sup_{k,p} \mathbb{E}_{S^n}\big[\langle f^*_{k,p}(x), \psi(x)\rangle\big] - \mathbb{E}_{\mu}\big[\langle f^*_{k,p}(x), \psi(x)\rangle\big] \le d_n(\epsilon)\Big) \ge 1-\epsilon$
3. $f^*_{k,p}$ is in a hypothesis class $\mathscr{F}$ with Rademacher complexity at most $\dfrac{4\log_2 en}{n}$

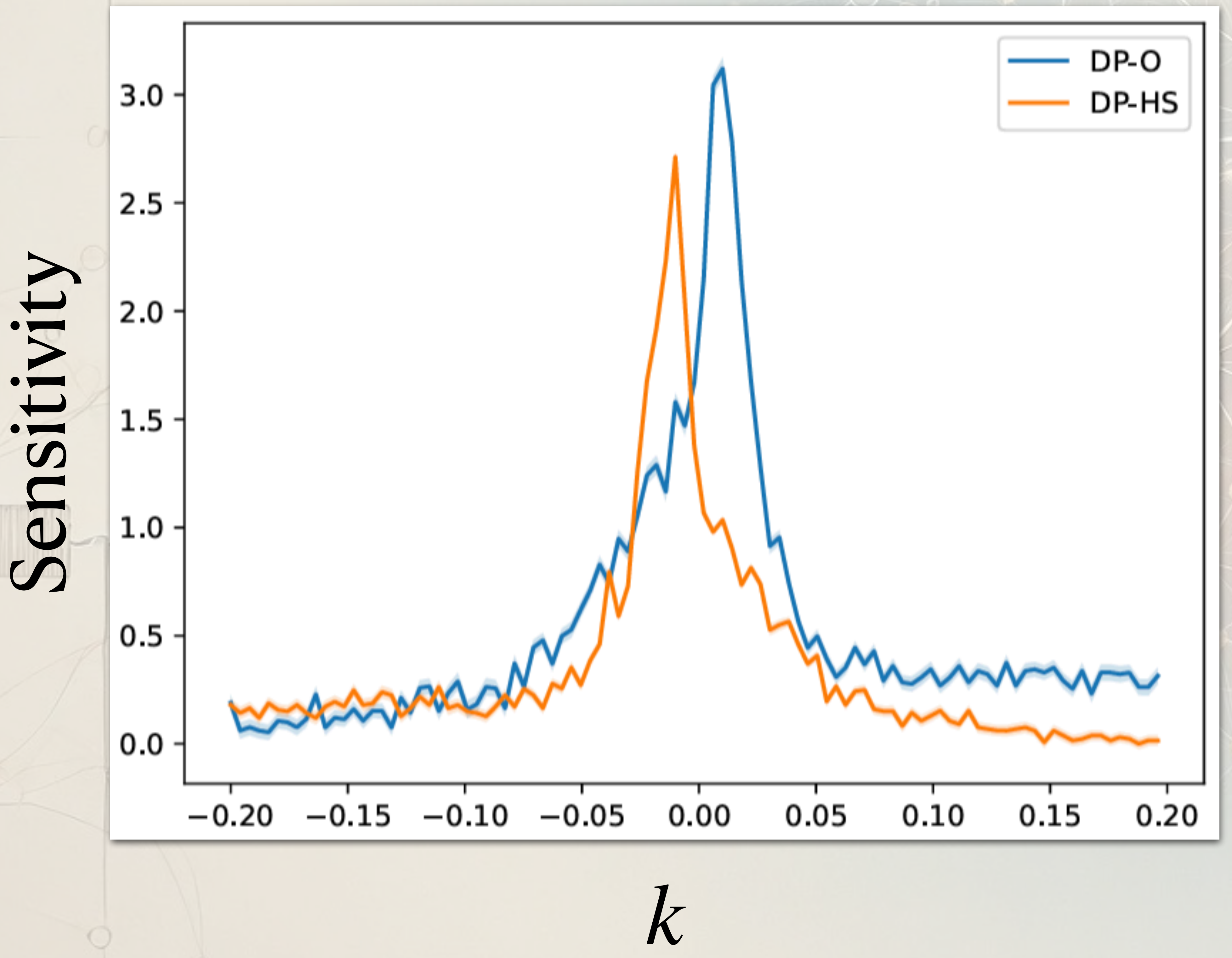
Constraint Sample Complexity

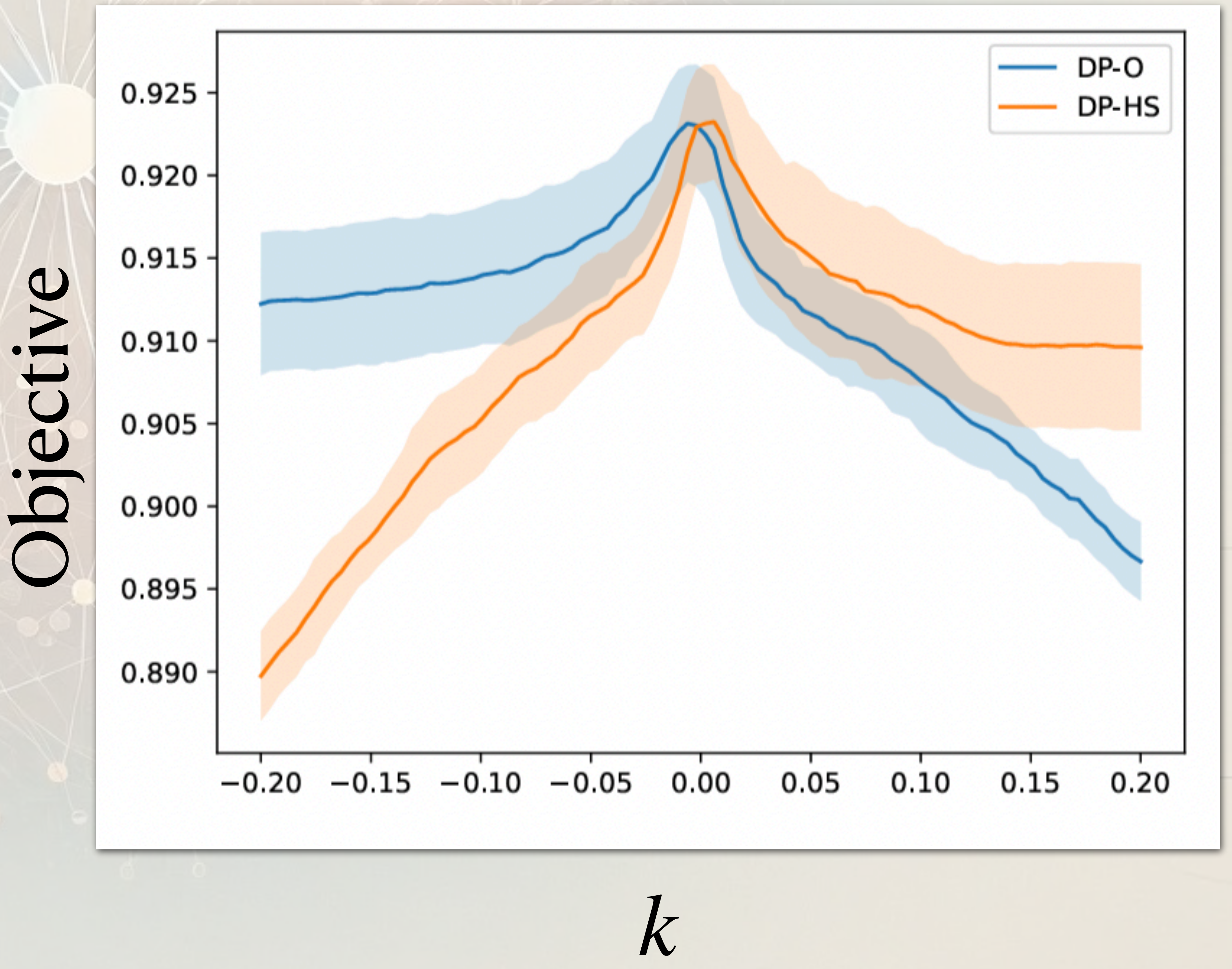
Constraint Statistical Generalization: $O(\sqrt{\log n/n}, \sqrt{\log(1/\epsilon)/n}, \epsilon')$ with probability at least $1-\epsilon$ and when scores are ϵ'-accurate:

1. $\mathbb{E}\left[\langle f(x), \hat{\psi}(x) - \psi(x)\rangle\right] \leq \epsilon'$
2. $\Pr\left(\sup_{k,p} \mathbb{E}_{S^n}\left[\langle f^*_{k,p}(x), \psi(x)\rangle\right] - \mathbb{E}_{\mu}\left[\langle f^*_{k,p}(x), \psi(x)\rangle\right] \leq d_n(\epsilon)\right) \geq 1-\epsilon$
3. $f^*_{k,p}$ is in a hypothesis class $\mathcal{F}$ with Rademacher complexity at most $\dfrac{4\log_2 en}{n}$
4. Using Rademacher generalization inequality, we have
   $d_n(\epsilon) = O\left(\sqrt{\dfrac{\log n}{n}} + \sqrt{\dfrac{\log(\epsilon)}{n}}\right)$

Objective Sample Complexity

- **Objective Statistical Generalization**: $O((\log n/n)^{1/2\gamma}, (\log(1/\epsilon)/n)^{1/2\gamma}, \epsilon')$ where γ measures the sensitivity of the constraint to the change of predictor

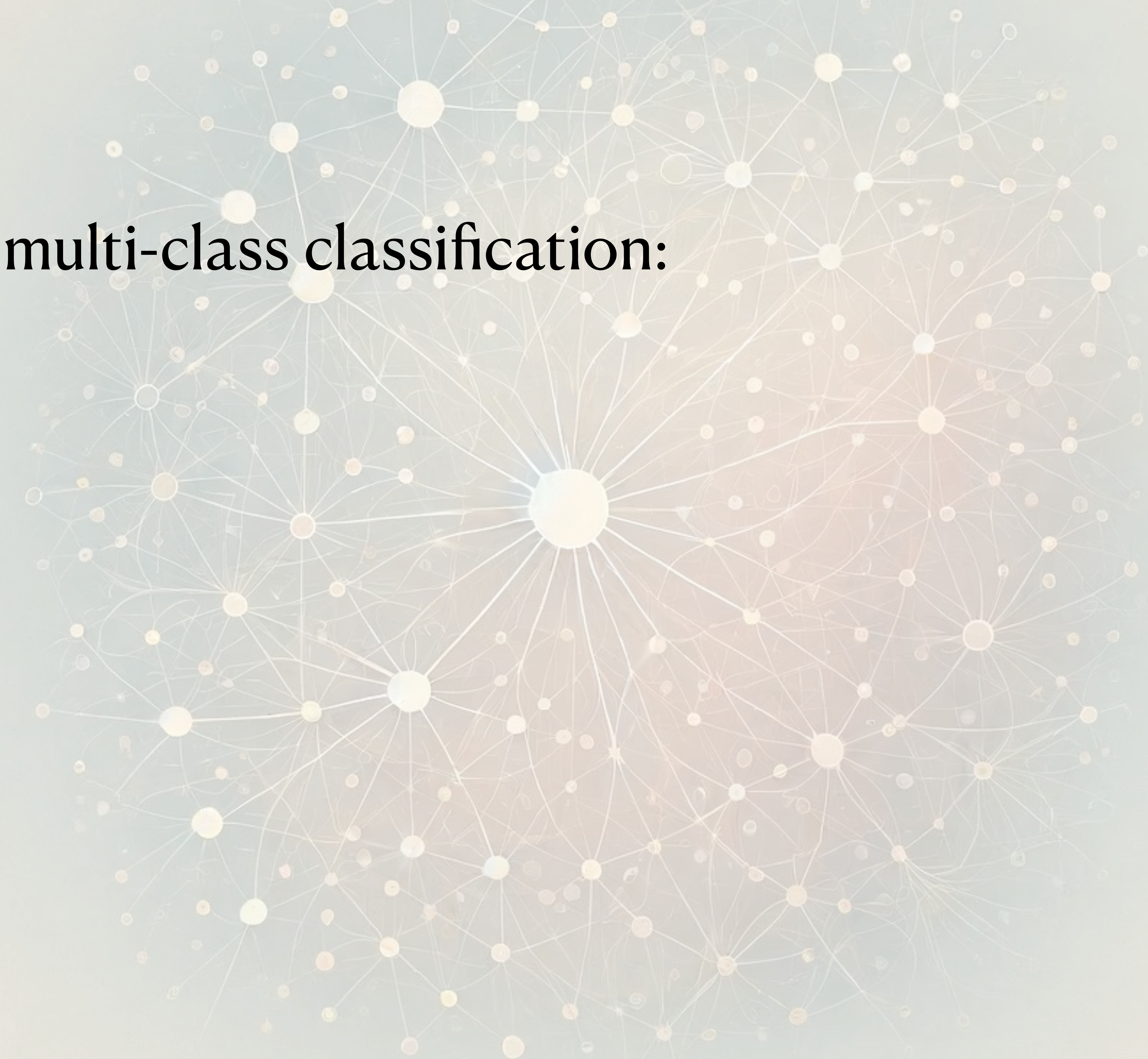




Constrained Classification

Constrained Classification

- Fairness Criteria in multi-class classification:



Constrained Classification

- Fairness Criteria in multi-class classification:
    - Embedding function of accuracy: $\psi_2(x) = [P(Y = 1 \,|\, X = x), \ldots, P(Y = K \,|\, X = x)]$

Constrained Classification

- Fairness Criteria in multi-class classification:
  - Embedding function of accuracy: $\psi_2(x) = [P(Y = 1 | X = x), \ldots, P(Y = K | X = x)]$
  - Embedding function of DP for the first class: $\psi_1(x) = [t(A), 0, \ldots, 0]$

Constrained Classification

- Fairness Criteria in multi-class classification:
  - Embedding function of accuracy: $\psi_2(x) = [P(Y = 1 \mid X = x), \ldots, P(Y = K \mid X = x)]$
  - Embedding function of DP for the first class: $\psi_1(x) = [t(A),0,\ldots,0]$
  - Embedding function of EO for the first class: $\psi_1(x) = [t'(A)P(Y = 1 \mid X = x),0,\ldots,0]$

Constrained Classification

- Fairness Criteria in multi-class classification:
  - Embedding function of accuracy: $\psi_2(x) = [P(Y = 1 \,|\, X = x), \ldots, P(Y = K \,|\, X = x)]$
  - Embedding function of DP for the first class: $\psi_1(x) = [t(A),0,\ldots,0]$
  - Embedding function of EO for the first class: $\psi_1(x) = [t'(A)P(Y = 1 \,|\, X = x),0,\ldots,0]$
  - Bayes DP-classifier: $\mathrm{argmax}[P(Y = 1 \,|\, X = x) - kt(A), \ldots, P(Y = K \,|\, X = x)]$

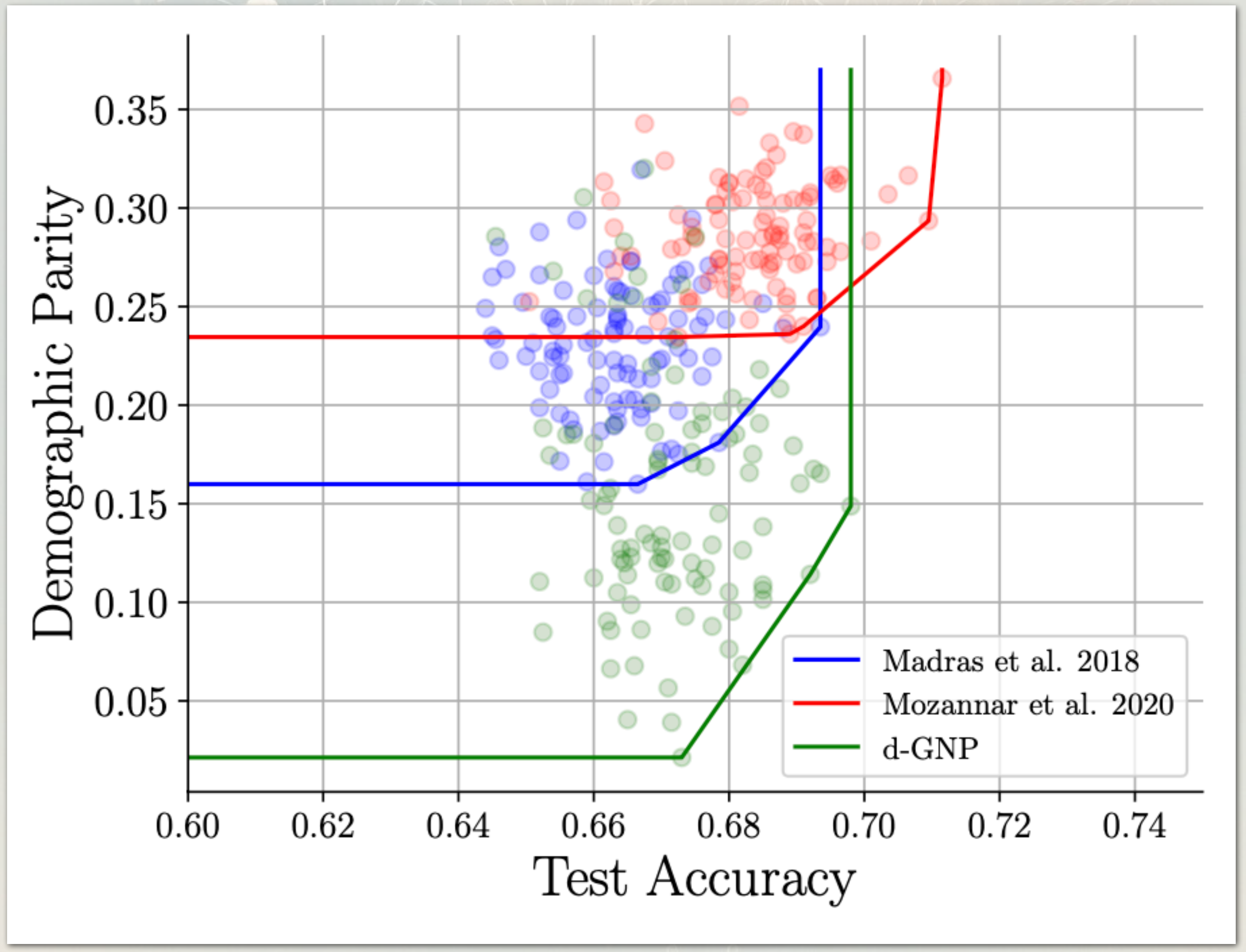
Constrained Classification

- Fairness Criteria in multi-class classification:
  - Embedding function of accuracy: $\psi_2(x) = [P(Y = 1 \mid X = x), \ldots, P(Y = K \mid X = x)]$
  - Embedding function of DP for the first class: $\psi_1(x) = [t(A), 0, \ldots, 0]$
  - Embedding function of EO for the first class: $\psi_1(x) = [t'(A)P(Y = 1 \mid X = x), 0, \ldots, 0]$
  - Bayes DP-classifier: argmax$[P(Y = 1 \mid X = x) - kt(A), \ldots, P(Y = K \mid X = x)]$
  - Bayes EO-classifier: argmax$[P(Y = 1 \mid X = x)(1 - kt'(A)), \ldots, P(Y = K \mid X = x)]$

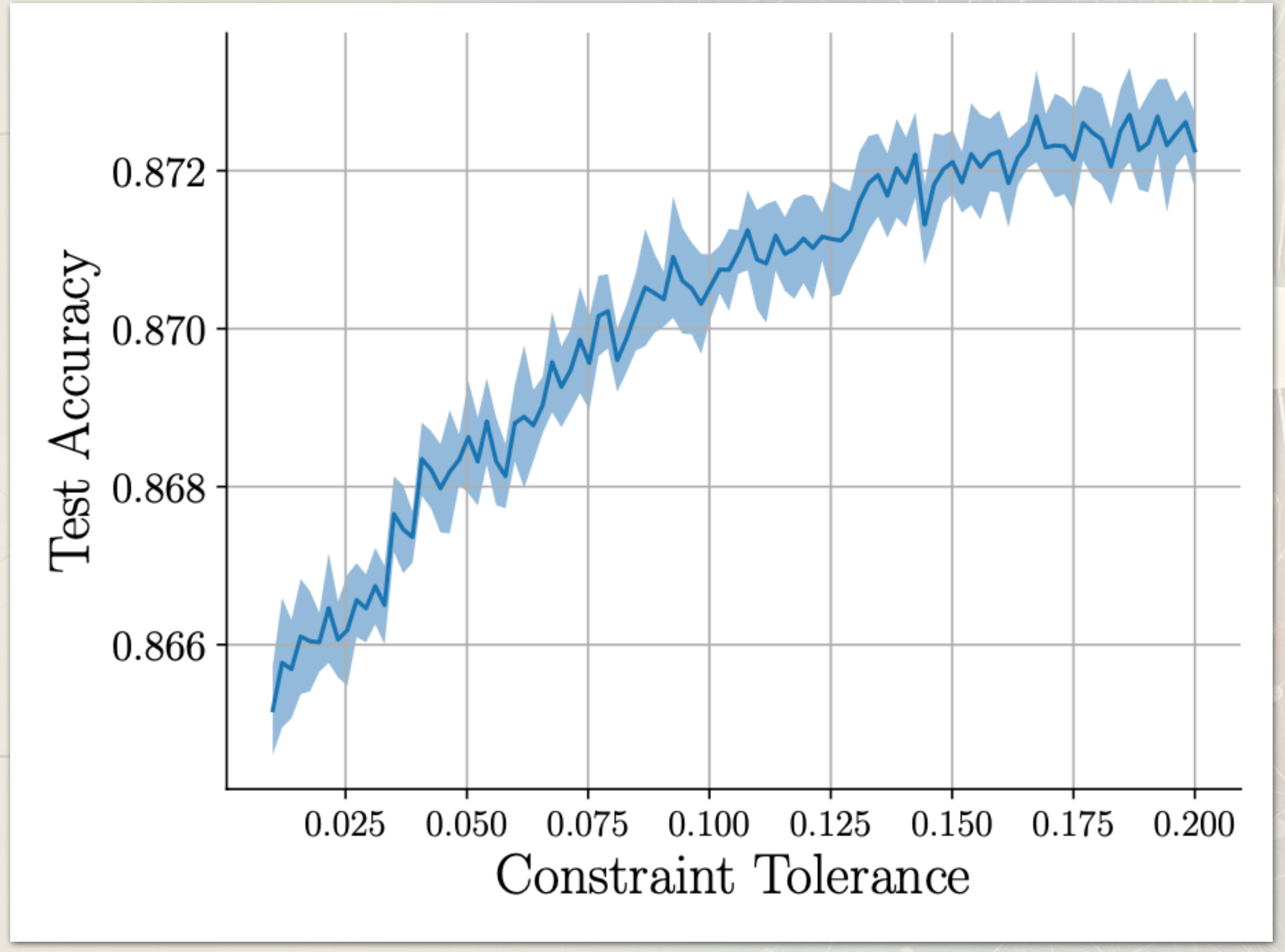
Constrained Classification

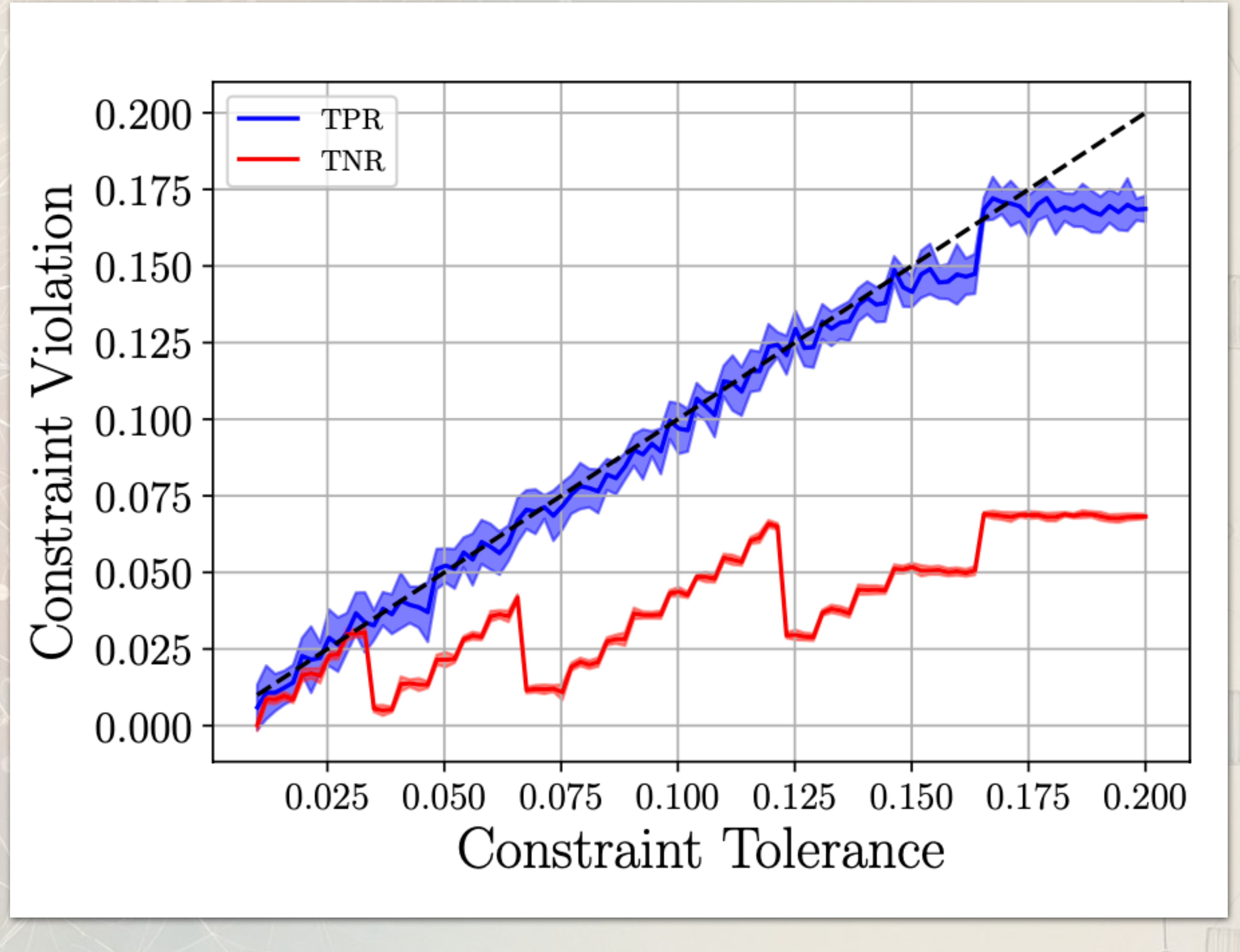
- Fairness Criteria in multi-class classification:
  - Embedding function of accuracy: $\psi_2(x) = [P(Y = 1 | X = x), \ldots, P(Y = K | X = x)]$
  - Embedding function of DP for the first class: $\psi_1(x) = [t(A), 0, \ldots, 0]$
  - Embedding function of EO for the first class: $\psi_1(x) = [t'(A)P(Y = 1 | X = x), 0, \ldots, 0]$
  - Bayes DP-classifier: argmax$[P(Y = 1 | X = x) - kt(A), \ldots, P(Y = K | X = x)]$
  - Bayes EO-classifier: argmax$[P(Y = 1 | X = x)(1 - kt'(A)), \ldots, P(Y = K | X = x)]$
- Binary classification: Different Thresholding

Experiments: COMPAS Dataset



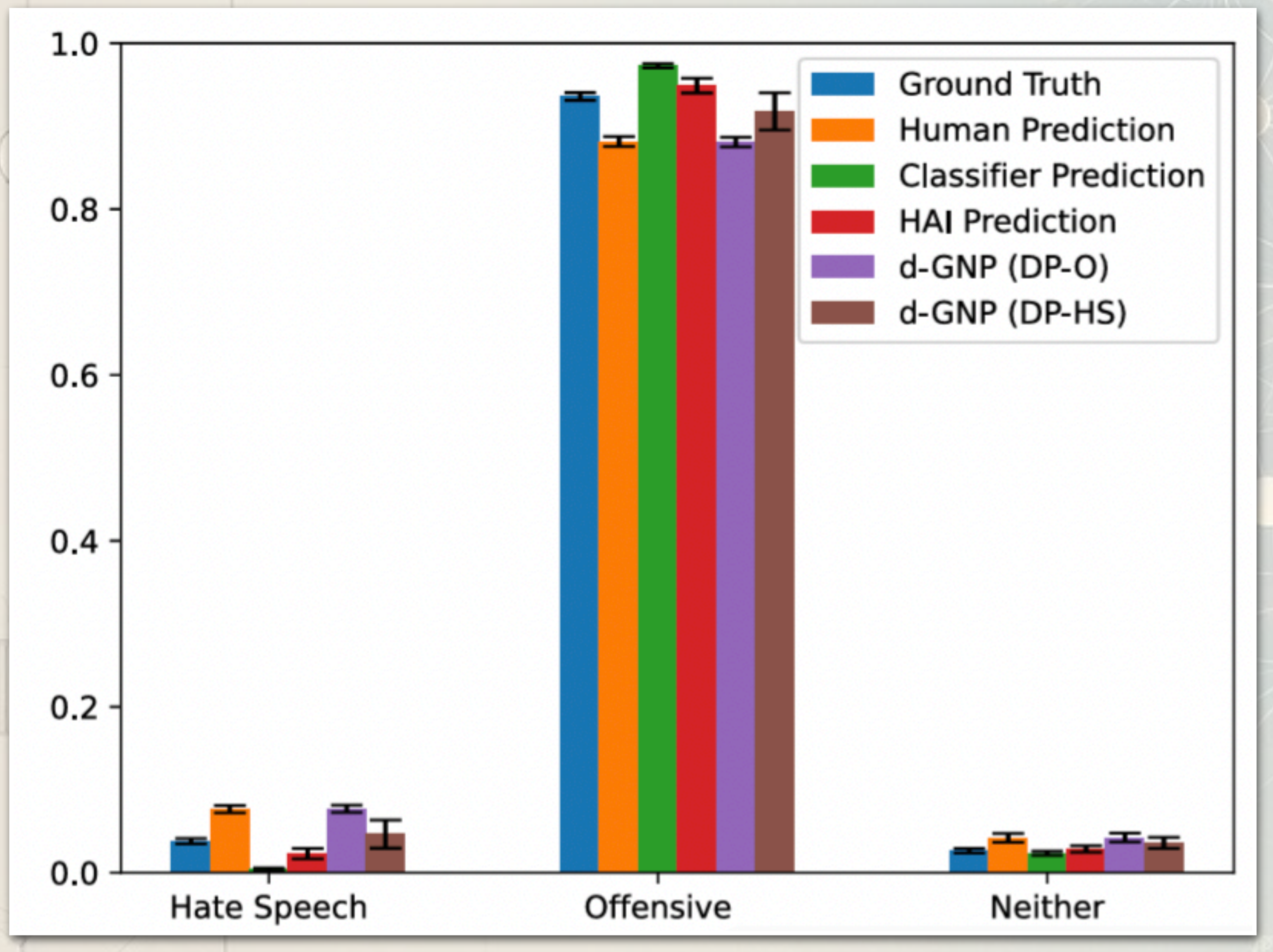
Experiments: American Community Survey



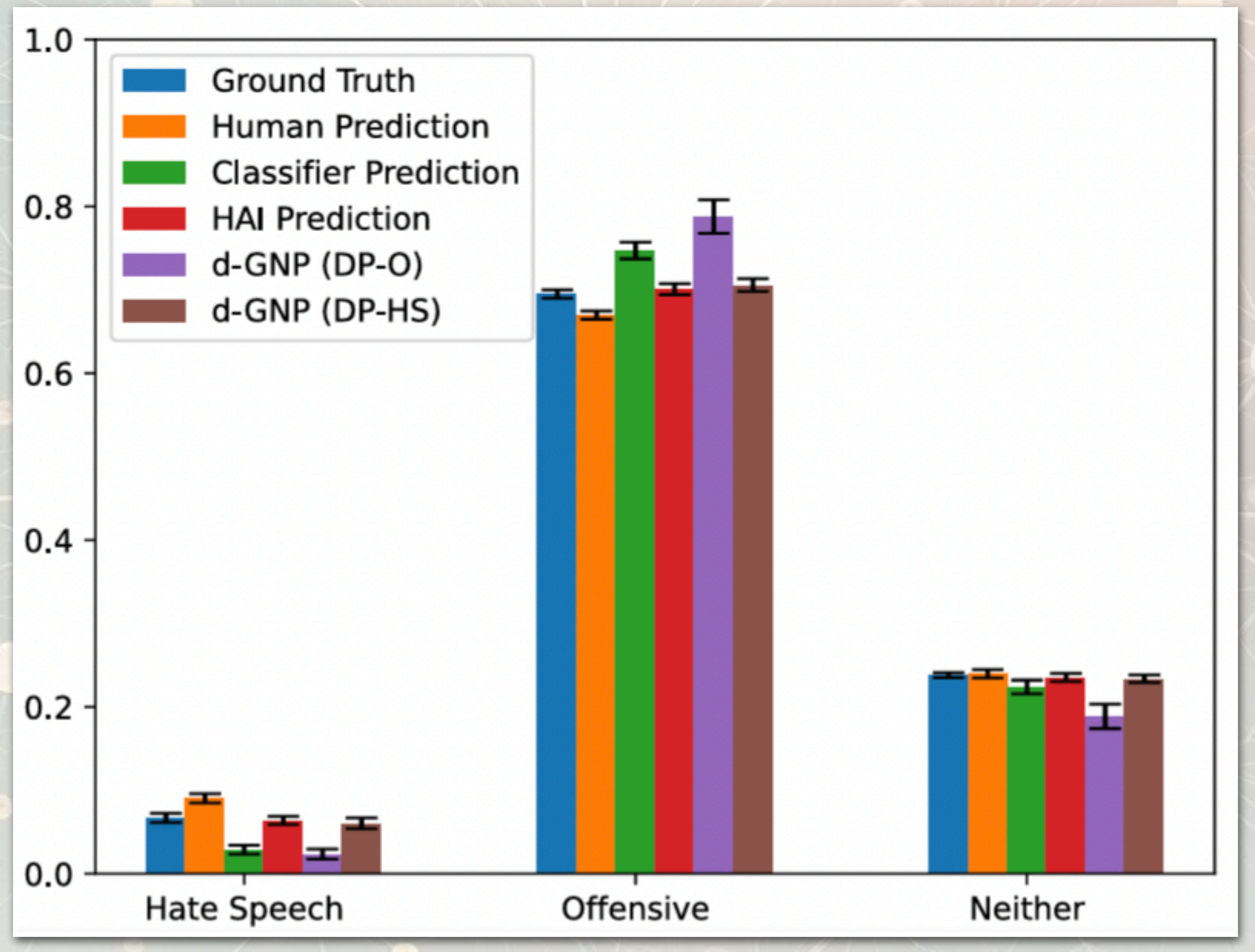


Experiments: Hatespeech Dataset

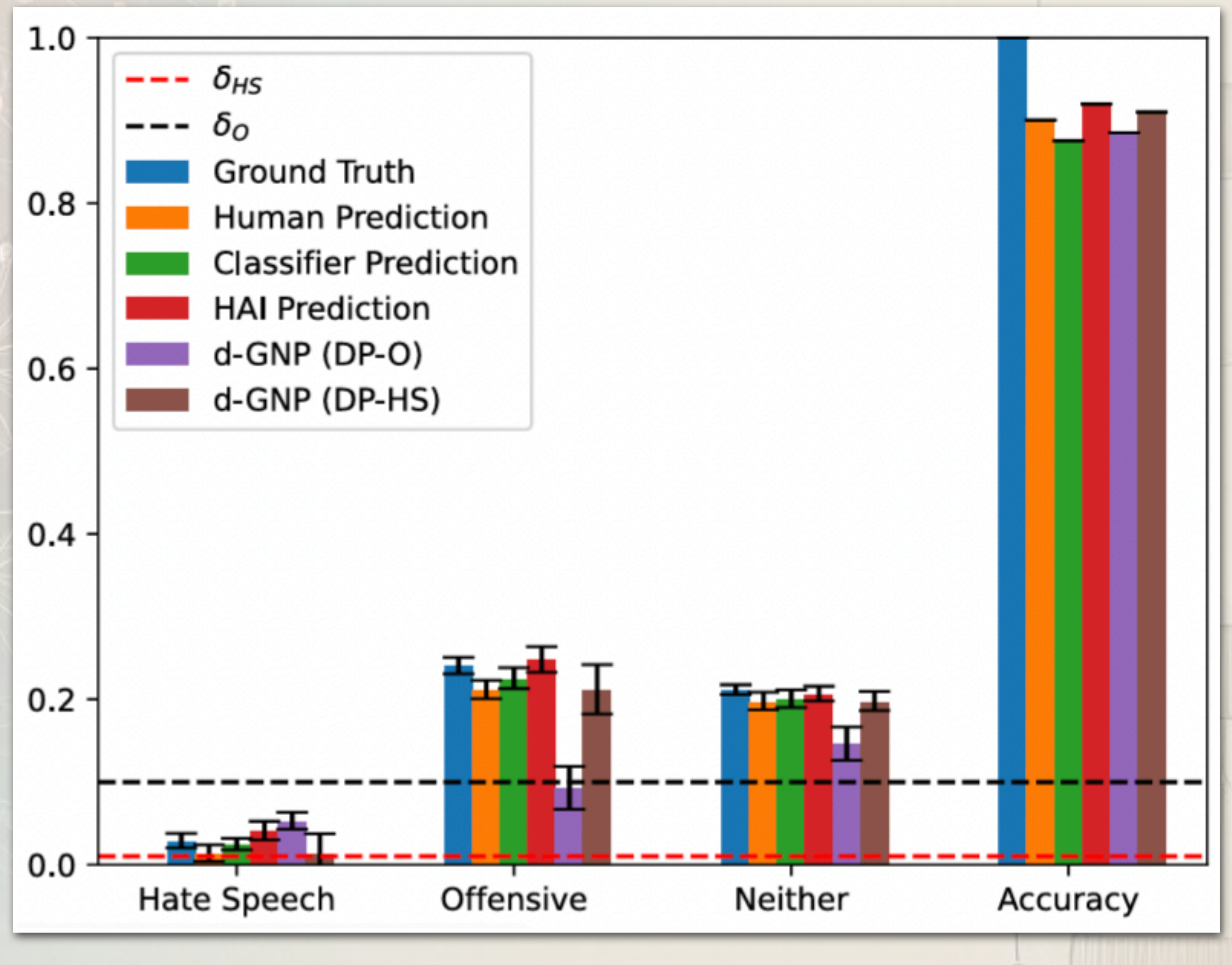
African American



Not African American



Difference



Conclusion

- Constrained Classification and L2D are solvable by a generalization of NP-Lemma
- Find embedding function (scores) of each constraint and loss and maximize a linear combination of them
- No need for regularization, therefore computation efficiency
- Statistical generalization of d-GNP
- Experiments on COMPAS, ACSIncome, and Hatespeech datasets