Towards Trustworthy Machine Learning via Distribution Matching

David I. Inouye
While ML has made great strides in recent years, ML still has many issues:

- Large language models could compound historical bias against minorities.
- Autonomous driving systems could cause loss of life when exposed to unexpected conditions.
- Medical ML systems could recommend fatal treatments based on false counterfactual prediction.
- Scientists may make incorrect scientific conclusions based on black-box models.
The next generation of trustworthy ML will need to exhibit properties beyond accuracy.

- Group Fairness – Are the predictions fair w.r.t. age or race?
- Robustness – Are the predictions accurate even in new environments?
- Causality – Can counterfactual queries be correctly estimated?
- Explainability – Can distribution shifts be explained?
Fair classification example:
Optimize performance given fairness constraint

- Let $f$ be a classification model and let $x, y, d,$ and $\hat{y} \equiv f(x)$ be the input, label, sensitive attribute, and model’s prediction respectively
- Let $p_f(x, y, d, \hat{y})$ denote the joint distribution over all variables
- **Demographic parity (DP) difference** measures the difference between the prediction probability between groups

$$\Delta_{DP}(f) = |p_f(\hat{y}=1|d=1) - p_f(\hat{y}=1|d=2)|$$

- Fair classification w.r.t. DP can be formalized as a task objective subject to a fairness constraint:

$$\min_{f} \mathcal{L}(f) \equiv \mathbb{E}_{p(x,y)}[\ell(f(x), y)]$$

s.t. $\Delta_{DP}(f) \leq \delta$
The fairness constraint can be satisfied via distribution matching

- First, notice that $\Delta_{DP}(f)$ is total variation distance

$$\Delta_{DP}(f) = D_{TV}(p_f(\hat{y}|d=1), p_f(\hat{y}|d=2)) \leq \delta$$

- This is a distribution matching constraint on $p(\hat{y}|d)$!

- In practice, the model is often decomposed into feature extraction followed by a classifier head $f_{cls}$

$$f(x,d) = f_{cls}(g(x,d))$$

- where $z = g(x,d)$ is a latent representation

- The task loss can be written in terms of $g$, i.e., $\mathcal{L}(g) = \min_{f_{cls}} \mathbb{E}_{p(x,y)}[\ell(f_{cls}(g(x,d)), y)]$

- To ensure DP fairness, a sufficient condition is to ensure the latent TV distance is small (due to data processing inequality for $f$-divergences)

$$D_{TV}(p_f(\hat{y}|d=1), p_f(\hat{y}|d=2)) \leq D_{TV}(p_g(z|d=1), p_g(z|d=2)) \leq \delta$$

- This is a distribution matching constraint on $p_g(z|d)$!
Distribution matching for trustworthy ML takes the form of (soft) task constraints

**Definition 1:** Given a task objective $\mathcal{L}(g)$, a distribution matching constraint imposes matching on the latent representation $z := g(x, d, \epsilon)$, where $g \in \mathbb{G}$ is called a matcher and $D(p, q)$ is a divergence:

Hard DM constraint:

$$\min_{g \in \mathbb{G}} \mathcal{L}(g)$$

subject to

$$D\left(p_g(z|d=1), p_g(z|d=2)\right) \leq \delta$$

Soft DM constraint (i.e., regularization):

$$\min_{g \in \mathbb{G}} \mathcal{L}(g) + \lambda D\left(p_g(z|d=1), p_g(z|d=2)\right)$$

Learn a rotation + 1D projection so that $p_g(z_1|d)$ is aligned

Learn 2D $g$ such that $p(z_1, z_2|d)$ is aligned where the red distribution is fixed, i.e., $g(x, 2) = x$
A matcher can have different structures $\mathcal{G}$ depending on the context

- **Translation** matcher, i.e., $g(x, d) = \begin{cases} 
    x, & \text{if } d = 1 \\
    \tilde{g}(x), & \text{otherwise}
\end{cases}$

- **Shared** matcher between domains, i.e., $g(x, d) = \tilde{g}(x)$

- **Invertible** matcher, i.e., $\exists g^{-1}$ s.t. $\forall x$, $g^{-1}(g(x, d), d) = x$
  - **Approximately invertible** via cycle consistency $\exists f$ s.t. $\forall x$, $f(g(x, d), d) \approx x$

- **Stochastic** matcher, i.e., $g(x, d, \epsilon)$, where $\epsilon$ is exogenous noise.
Distribution matching has been known by other names.

- Distribution alignment
- (Domain-)Invariant representation learning
- Adversarial representation learning
- Mutual information minimization
Can we generalize distribution matching to conditional distributions?

- Yes! But we need a notion of **conditional divergence**.

**Definition 2:** Given index sets $\mathcal{A}, \mathcal{B} \subseteq \{1, 2, ..., m\}$, a conditional divergence $D_{\mathcal{A}|\mathcal{B}}(p, q)$ is a function that satisfies two properties:

1. Non-negativity, i.e., $D_{\mathcal{A}|\mathcal{B}}(p, q) \geq 0$.
2. Conditional distribution equality, i.e., $D_{\mathcal{A}|\mathcal{B}}(p, q) = 0 \iff p(z_\mathcal{A}|z_\mathcal{B}) = q(z_\mathcal{A}|z_\mathcal{B}), \forall z_\mathcal{B}$

- Any divergence $D$ can be extended via an expectation over some $\tilde{p}(z_\mathcal{B})$

$$D_{\mathcal{A}|\mathcal{B}}^{\mathcal{E}}(p, q) = \mathbb{E}_{\tilde{p}(z_\mathcal{B})}[D(p(z_\mathcal{A}|z_\mathcal{B}), q(z_\mathcal{A}|z_\mathcal{B}))]$$

- A conditional divergence allows the marginals of $z_\mathcal{B}$ to be different

David I. Inouye, Purdue University
Conditional distribution matching is much less explored but is more general

**Definition 3:** Given a task objective $\mathcal{L}(g)$, a conditional matching constraint imposes matching on latent conditional distributions given a conditional divergence $D_{\mathcal{A}|\mathcal{B}}$:

Hard DM constraint:

$$\min_{g \in \mathcal{G}} \mathcal{L}(g)$$

s.t. $D_{\mathcal{A}|\mathcal{B}}(p_g(z|d=1), p_g(z|d=2)) \leq \delta$

Soft DM constraint (i.e., regularization):

$$\min_{g \in \mathcal{G}} \mathcal{L}(g) + \lambda D_{\mathcal{A}|\mathcal{B}}(p_g(z|d=1), p_g(z|d=2))$$

Learn a rotation + 1D projection so that $p_g(z_2|z_1, d)$ is aligned

Learn 2D $g$ such that $p(z_2|z_1, d)$ is aligned where the red distribution is fixed, i.e., $g(x, 2) = x$
Conditional distribution matching is much less explored but is more general

Definition 3: Given a task objective $\mathcal{L}(g)$, a conditional matching constraint imposes matching on latent conditional distributions given a conditional divergence $D_{\mathcal{A}|\mathcal{B}}$:

Hard DM constraint:

$$\min_{g \in \mathcal{G}} \mathcal{L}(g)$$

s.t. $D_{\mathcal{A}|\mathcal{B}}(p_g(z|d=1), p_g(z|d=2)) \leq \delta$

Soft DM constraint (i.e., regularization):

$$\min_{g \in \mathcal{G}} \mathcal{L}(g) + \lambda D_{\mathcal{A}|\mathcal{B}}(p_g(z|d=1), p_g(z|d=2))$$

Matching the binary class probability given a single latent feature $p(y|\tilde{z}, d)$

- In this case, $z \equiv (y, \tilde{z})$ and $\mathcal{A} = \{1\}, \mathcal{B} = \{2\}$
- For every value of $\tilde{z}$ the positive class probability $p(y = 1|\tilde{z}, d)$ must match
- (This is related to Invariant Risk Minimization, IRM)

Mismatched ☹️

$$p(y = 1|z, d)$$

Partially matched 😊

$$p(y = 1|z, d)$$

Matched 😊

$$p(y = 1|z, d)$$
Many group fairness measures can be described as distribution matching constraints

- Marginal Matching
  - Demographic parity
  - \( p(\hat{y}|d) \)

- Conditional Matching
  - Equalized odds
  - \( p(\hat{y}|y) \)
  - \( p(\hat{y}|y, x, d) \)
  - \( \cdots \)
  - \( p(\hat{y}|y, x_1, \ldots, x_m, d) \)

  - Conditional equalized odds
  - \( p(\hat{y}|y, x_1, \ldots, x_m, d) \)

  - Conditional Demographic Parity
  - \( p(\hat{y}|x_1, d) \)
  - \( p(\hat{y}|x_1, x_2, d) \)
  - \( \cdots \)
  - \( p(\hat{y}|x_1, \ldots, x_m, d) \)

- Sufficiency
  - \( p(y|\hat{y}, d) \)

- In this case, the latent representation is merely the prediction \( (z \equiv \hat{y}) \) and the matcher is merely the predictor \( (g \equiv f) \)
Fair classification is one of the trustworthy ML applications via distribution matching

<table>
<thead>
<tr>
<th>Application</th>
<th>Method</th>
<th>Task Loss</th>
<th>Distribution to Match</th>
<th>Matcher Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fair Classification</td>
<td>Fair VAE</td>
<td>Classification loss</td>
<td>$p_g(z</td>
<td>d)$</td>
</tr>
<tr>
<td></td>
<td>Adversarially Fair</td>
<td></td>
<td></td>
<td>Shared</td>
</tr>
<tr>
<td></td>
<td>Fair Flows</td>
<td></td>
<td></td>
<td>Invertible</td>
</tr>
</tbody>
</table>
Many DG methods can be viewed as matching different latent distributions

<table>
<thead>
<tr>
<th>Application</th>
<th>Method</th>
<th>Task Loss</th>
<th>Distribution to Match</th>
<th>Matcher Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fair Classification</td>
<td>Fair VAE</td>
<td>Classification loss</td>
<td>$p_g(z</td>
<td>d)$</td>
</tr>
<tr>
<td></td>
<td>Adversarially Fair</td>
<td></td>
<td></td>
<td>Shared</td>
</tr>
<tr>
<td></td>
<td>Fair Flows</td>
<td></td>
<td></td>
<td>Invertible</td>
</tr>
<tr>
<td>Domain Generalization</td>
<td>DANN</td>
<td>Classification loss</td>
<td>$p_g(z</td>
<td>d)$</td>
</tr>
<tr>
<td></td>
<td>CDANN</td>
<td></td>
<td>$p_g(z</td>
<td>y,d)$</td>
</tr>
<tr>
<td></td>
<td>IRM</td>
<td></td>
<td>$p_g(y</td>
<td>z,d)$</td>
</tr>
<tr>
<td></td>
<td>Fishr</td>
<td></td>
<td>$p_g(\nabla_{\theta}L_\theta(x)</td>
<td>d)$</td>
</tr>
</tbody>
</table>

[David I. Inouye, Purdue University]
Causal ML methods have different task losses than classification

<table>
<thead>
<tr>
<th>Application</th>
<th>Method</th>
<th>Task Loss</th>
<th>Distribution to Match</th>
<th>Matcher Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fair Classification</td>
<td>Fair VAE</td>
<td>Classification loss</td>
<td>( p_g(z</td>
<td>d) )</td>
</tr>
<tr>
<td></td>
<td>Adversarially Fair</td>
<td></td>
<td></td>
<td>Shared</td>
</tr>
<tr>
<td></td>
<td>Fair Flows</td>
<td></td>
<td></td>
<td>Invertible</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Domain Generalization</td>
<td>DANN</td>
<td>Classification loss</td>
<td>( p_g(z</td>
<td>d) )</td>
</tr>
<tr>
<td></td>
<td>CDANN</td>
<td></td>
<td>( p_g(z</td>
<td>y, d) )</td>
</tr>
<tr>
<td></td>
<td>IRM</td>
<td></td>
<td>( p_g(y</td>
<td>z, d) )</td>
</tr>
<tr>
<td></td>
<td>Fishr</td>
<td></td>
<td>( p_g(\nabla_\theta \mathcal{L}_\theta(x)</td>
<td>d) )</td>
</tr>
<tr>
<td>Causality</td>
<td>CATE</td>
<td>Factual risk</td>
<td>( p_g(z</td>
<td>d) )</td>
</tr>
<tr>
<td></td>
<td>ICP</td>
<td>n/a</td>
<td>( p_g(y</td>
<td>z_{Pa(y)}, d) )</td>
</tr>
<tr>
<td></td>
<td>Domain</td>
<td>NLL</td>
<td>( p_g(z_i</td>
<td>z_{&lt;i}, d) )</td>
</tr>
<tr>
<td></td>
<td>Counterfactuals</td>
<td>( \forall i ) not intervened</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Bai et al., 2023] [Kulinski et al., 2023] [Kulinski & Inouye, 2023]
Explaining distribution shifts can be cast as finding an interpretable matcher

<table>
<thead>
<tr>
<th>Application</th>
<th>Method</th>
<th>Task Loss</th>
<th>Distribution to Match</th>
<th>Matcher Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fair Classification</td>
<td>Fair VAE</td>
<td>Classification loss</td>
<td>$p_g(z</td>
<td>d)$</td>
</tr>
<tr>
<td></td>
<td>Adversarially Fair</td>
<td></td>
<td></td>
<td>Shared</td>
</tr>
<tr>
<td></td>
<td>Fair Flows</td>
<td></td>
<td></td>
<td>Invertible</td>
</tr>
<tr>
<td>Domain Generalization</td>
<td>DANN</td>
<td>Classification loss</td>
<td>$p_g(z</td>
<td>d)$</td>
</tr>
<tr>
<td></td>
<td>CDANN</td>
<td></td>
<td>$p_g(z</td>
<td>y,d)$</td>
</tr>
<tr>
<td></td>
<td>IRM</td>
<td></td>
<td>$p_g(y</td>
<td>z,d)$</td>
</tr>
<tr>
<td></td>
<td>Fishr</td>
<td></td>
<td>$p_g(∇_θ L_θ(x)</td>
<td>d)$</td>
</tr>
<tr>
<td>Causality</td>
<td>CATE</td>
<td>Factual risk</td>
<td>$p_g(z</td>
<td>d)$</td>
</tr>
<tr>
<td></td>
<td>ICP</td>
<td>n/a</td>
<td>$p_g(y</td>
<td>z_{Pa(y)},d)$</td>
</tr>
<tr>
<td></td>
<td>Domain Counterfactuals</td>
<td>NLL</td>
<td>$p_g(z_i</td>
<td>z_{&lt;i},d)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\forall i$ not intervened</td>
<td></td>
</tr>
<tr>
<td>Dist. Shift</td>
<td>Sparse transport</td>
<td>Reg. Transport Cost</td>
<td>$p_g(z</td>
<td>d)$</td>
</tr>
<tr>
<td>Explanations</td>
<td>Interpretable transport</td>
<td>Transport Cost</td>
<td>$p_g(z</td>
<td>d)$</td>
</tr>
</tbody>
</table>

David I. Inouye, Purdue University
Some analogies to summarize
Classification is to current ML as scaling data or models? is to trustworthy ML

No, doesn’t solve fundamental issues
Classification is to current ML as application-specific design? is to trustworthy ML

No, it is not as broadly applicable as classification is to many different problems.
Classification is to current ML as distribution matching is to trustworthy ML.

In contrast to scaling, it is fundamentally different from classification.

In contrast to application-specific design, it is broadly applicable to many tasks.
Why could distribution matching be an enabling tool for trustworthy ML?

Two analogies: One fun, the other more technical
Classification is to Tigger as distribution matching is to **Eeyore**

Optimism – Pessimism
Classification is to task objective as
distribution matching is to **task constraints**

Positive goal – Negative constraint
Performance – Safety
Why could distribution matching be broadly applicable?
Class labels are to classification as domain labels are to distribution matching.

Data-driven
Easier to elicit from experts than formal definitions
Generic algorithms
What is distribution matching?
Divergence maximization is to classification as divergence minimization is to distribution matching
Distribution matching is representation learning with the opposite objective of classification.

Original Space

Representation Learning Objective

Classification

\[
\max_{g \in \mathcal{G}} D\left( p(g(x)|d=1), p(g(x)|d=2) \right)
\]

where \( g: \mathbb{R}^2 \rightarrow \mathbb{R} \) and \( D \) is a distribution divergence (e.g., KL, JSD, \( W_2 \))

Distribution matching

\[
\min_{g \in \mathcal{G}} D\left( p(g(x)|d=1), p(g(x)|d=2) \right)
\]

Optimal solution

\[
p(g^*(x)|d=1) = p(g^*(x)|d=2)
\]

Latent Space

David I. Inouye, Purdue University
Yet, prior distribution matching research lacks a unified scientific framework.

• Much prior work focuses specific applications (e.g., fairness or domain adaptation)
  • This stems from the fact that DM is rarely useful by itself
  • But DM is a much broader tool

• Other works only consider one algorithm (e.g., adversarial)
  • But there are diverse non-adversarial approaches

• DM has rarely been investigated in its own right
I aim to **unify distribution matching** under a common framework for trustworthy ML problems.

---

**Fundamentals** (Already covered)

**Applications** (Already covered)

**Algorithms**

**Evaluation**
Algorithms:
How do we enforce distribution matching in practice?
Matching algorithms fall into three different categories depending on how they estimate the theoretic divergence

• Adversarial matching (GAN)
  • First and continues to be the most popular approach to matching
  • Easy to implement, just add a discriminator for the domain
  • No restriction on model architectures
  • Challenging to optimize in practice
  • Hard to evaluate solution

• Likelihood-based algorithms (flows ⭐ [Cho et al. 2022] and VAEs ⭐ [Gong et al. 2023])
  • Less well-known
  • Non-adversarial so more stable to optimize
  • Flow-based algorithms requires invertible model
  • VAE-based algorithms usually enforce fixed prior distribution

• Other algorithms
  • Optimal transport algorithms
  • Statistical conditional independence tests
  • Iterative matching ⭐ [Zhou et al., 2022a,b]


David I. Inouye, Purdue University
Many matching algorithms form variational approximation of divergences

**Adversarial / Lower Bound**

\[
\min_g \max_h \bar{D}(p_g(z|d=1), p_g(z|d=2); h)
\]

- Variational maximization problem over critic \(h(z)\) forms a lower bound on a divergence

- Different choices of approximation yield lower bounds on JSD, Wasserstein, \(f\)-divergences

**Likelihood / Upper Bound**

\[
\min_g \min_q \bar{D}(p_g(z|d=1), p_g(z|d=2); q)
\]

- Variational minimization problem over variational distribution \(q\) forms an upper bound on a divergence

- We generalize flow-based methods for upper bounds ★ [Cho et al., 2022]

- We revisit VAE-based methods for upper bounds ★ [Gong et al., 2023]
Alignment Upper Bound (AUB) forms an upper bound on JS divergence via invertible models

- A variational upper bound of JSD:

\[
\overline{D}_{AUB}(g) = \min_{q(z)} \sum_{d=1}^{k} \mathbb{E}_p(x|d)[-\log|J_g(x,d)|q(g(x,d))]
\]

- \(q(z)\) is a prior density model shared among domains
- \(g(x,d)\) is invertible w.r.t \(x\) and \(|J_g(x,d)|\) is the determinant Jacobian of \(g\) w.r.t. \(x\)

- **Bound gap** is exactly \(KL\left(p_g(z), q(z)\right)\)

- **Any \(q\)** provides an upper bound on JSD + const

AUB optimization provides a cooperative alternative to adversarial matching

AUB cooperative matching problem

\[
\min_{g} \left( \min_{q(z)} \sum_{j=1}^{k} \mathbb{E}_{p(d)} \left[ \log J_{g}(x, d) | q(g(x, d)) \right] \right)
\]

- Minimizing \( g \) makes distributions closer to current \( q \) (left)
- Minimizing \( q(z) \) tightens bound by getting closer to the latent mixture \( p(z) = \sum_{d} p(d)p(z|d) \) (right)

The invertibility assumption can be relaxed via a decoder and reconstruction loss

<table>
<thead>
<tr>
<th>Model</th>
<th>Jensen-Shannon Divergence Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow</td>
<td>$\min_{q(z)} \mathbb{E}_p[- \log(</td>
</tr>
<tr>
<td>$\beta$-VAE ($\beta \leq 1$)</td>
<td>$\min_{q(z)} \mathbb{E}_p \left[ - \log \left( \frac{q(x</td>
</tr>
</tbody>
</table>

- The Jacobian term can be relaxed by using the ratio of decoder to encoder
  \[ |J_g(x, d)| \iff \frac{q(x|z, d)}{p_g(z|x, d)} \]

- To encourage approximate invertibility of $g$, we can include a reconstruction regularization which simplifies to a $\beta$-VAE objective with $\beta \leq 1$
Adding noise to JSD can reduce vanishing gradient and local minimum problems

<table>
<thead>
<tr>
<th>Model</th>
<th>Jensen-Shannon Divergence Upper Bound</th>
<th>Noisy JSD Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow: $z = g(x, d)$</td>
<td>$\min_{q(z)} \mathbb{E}_p[-\log(</td>
<td>J_g(x, d)</td>
</tr>
<tr>
<td>$\beta$-VAE ($\beta \leq 1$)</td>
<td>$\min_{q(z)} \mathbb{E}_p\left[-\log\left(\frac{q(x</td>
<td>z, d)}{p_g(z</td>
</tr>
</tbody>
</table>

Iterative matching flows iteratively solve 1D matching problems to create deep matcher

1. Find 1D projection that is maximally mismatched

$$\max_{\theta} \ W_2(p(\theta^T x|d=1), p(\theta^T x|d=2))$$

2. Match along this 1D projection by mapping to barycenter distribution

$$\min_{\tilde{g}} \ E_{p(\bar{x} = \theta^T x, d)} [\|\tilde{g}(\bar{x}, d) - \bar{x}\|^2]$$

s.t. $$D(p(\tilde{g}(\bar{x}, 1)|d=1), p(\tilde{g}(\bar{x}, 2)|d=2)) = 0$$

3. Update global matcher (add one layer) and repeat

$$g(x, d) = \tilde{g}(\theta^T x, d)\theta + x^\perp$$

$$g_{\text{global}}^{\text{new}} = g \circ g_{\text{global}}^{\text{old}}$$

$$x^{\text{new}} = g(x)$$

A projection-pursuit type of algorithm
<table>
<thead>
<tr>
<th>Application</th>
<th>Method</th>
<th>Task Loss</th>
<th>Distribution to align</th>
<th>Aligner Structure</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fair Classification</strong></td>
<td>Fair VAE</td>
<td>ERM</td>
<td>$p_g(z</td>
<td>d)$</td>
<td>Stochastic</td>
</tr>
<tr>
<td></td>
<td>Adversarially Fair</td>
<td></td>
<td></td>
<td>Shared</td>
<td>Adversarial</td>
</tr>
<tr>
<td></td>
<td>Fair Flows</td>
<td></td>
<td></td>
<td>Invertible</td>
<td>Flow-based</td>
</tr>
<tr>
<td><strong>Domain Generalization</strong></td>
<td>DANN</td>
<td>ERM</td>
<td>$p_g(z</td>
<td>d)$</td>
<td>Shared</td>
</tr>
<tr>
<td></td>
<td>CDANN</td>
<td></td>
<td>$p_g(z</td>
<td>y,d)$</td>
<td>Shared</td>
</tr>
<tr>
<td></td>
<td>IRM</td>
<td></td>
<td>$p_g(y</td>
<td>z,d)$</td>
<td>Shared</td>
</tr>
<tr>
<td></td>
<td>Fishr</td>
<td></td>
<td>$p_g(\nabla_\theta \mathcal{L}_\theta(x)</td>
<td>d)$</td>
<td>Implicit</td>
</tr>
<tr>
<td><strong>Causality</strong></td>
<td>CATE</td>
<td>Factual risk</td>
<td>$p_g(z</td>
<td>d)$</td>
<td>Invertible</td>
</tr>
<tr>
<td></td>
<td>ICP</td>
<td>n/a</td>
<td>$p_g(y</td>
<td>z_{PA(y)},d)$</td>
<td>Permutation</td>
</tr>
<tr>
<td></td>
<td>Domain Counterfactuals</td>
<td>NLL</td>
<td>$p_g(z_i</td>
<td>z_{&lt;i},d)$</td>
<td>Shared</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\forall i$ not intervened</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Dist. Shift Explanations</strong></td>
<td>Sparse transport</td>
<td>Reg. Transport Cost</td>
<td>$p_g(z</td>
<td>d)$</td>
<td>Sparse</td>
</tr>
<tr>
<td></td>
<td>Interpretable transport</td>
<td>Transport Cost</td>
<td>$p_g(z</td>
<td>d)$</td>
<td>Sparse or cluster</td>
</tr>
</tbody>
</table>

[David I. Inouye, Purdue University]
Classification is to complexity **upper bounds** as
distribution matching is to complexity **lower bounds**

$P \neq NP$? - Easy to construct exponential alg. but nearly impossible to prove lower bound.
Classification - Constructing one classifier that works is like proving a complexity upper bound
DM - Ensuring that no classifier can work is like proving a complexity lower bound
Evaluation:
How do we evaluate DM?
Distribution matching evaluation requires comparing sets in high dimensions

• Classification metric - Empirical average over point-wise distances/losses \( \ell(\hat{y}, y) \)

\[
L_{\text{cls}}(f, p(x, y)) \approx \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)
\]

• Distribution matching objective – Must compute set-wise distance

\[
L_{\text{DM}}(g, p(x, d)) = D(p(g(x)|d_1), p(g(x)|d_2)) \approx \tilde{D} \left( \{z_i^{(1)}\}_{i=1}^{n_1}, \{z_i^{(2)}\}_{i=1}^{n_2} \right) = ?
\]

• Where \( \tilde{D} \) is a set-wise distance function that approximates the true divergence \( D \) given only samples
Current DM evaluation methods are diverse with varied strengths and weaknesses

1. Qualitative
   - Easy to inspect images or 2D visualization
   - Subjective and unsystematic

2. Two-sample test statistics
   - Examples: Demographic parity in fairness, FID in image generation models
   - Often simple to compute
   - Usually, necessary but not sufficient condition for match
     (e.g., two distributions can have the same mean and covariance but be quite different)

3. Empirical optimal transport
   - Wasserstein divergence is well-defined for empirical distributions (i.e., comparing samples directly)
   - Can be computed efficiently for empirical distributions via Sinkhorn algorithm or for 1D distributions via sorting
   - Strongly depends on the geometry of latent space
   - May not scale to high dimensions since non-parametric

4. Variational bounds on divergences
   - Lower bounds – Inner maximization of adversarial methods
   - Upper bounds – Inner minimization of likelihood-based methods
   - May scale better in high dimensions since parametric
   - Looseness of bounds is hard to estimate or quantify
DM metrics can be unified under these four categories

<table>
<thead>
<tr>
<th>Distribution Matching Metric</th>
<th>Category</th>
<th>Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compare images</td>
<td>Qualitative</td>
<td></td>
</tr>
<tr>
<td>t-SNE plot</td>
<td>Qualitative</td>
<td></td>
</tr>
<tr>
<td>Demographic parity in fairness</td>
<td>Two-sample statistic</td>
<td></td>
</tr>
<tr>
<td>Frechet Inception Distance (FID)</td>
<td>Two-sample statistic</td>
<td></td>
</tr>
<tr>
<td>Maximum Mean Discrepancy (MMD)</td>
<td>Two-sample statistic</td>
<td></td>
</tr>
<tr>
<td>Entropic-regularized discrete OT</td>
<td>Empirical OT</td>
<td></td>
</tr>
<tr>
<td>Sliced Wasserstein distance</td>
<td>Empirical OT</td>
<td></td>
</tr>
<tr>
<td>$f$-divergence adversarial loss</td>
<td>Variational</td>
<td>Lower</td>
</tr>
<tr>
<td>Wasserstein adversarial loss</td>
<td>Variational</td>
<td>Lower</td>
</tr>
<tr>
<td>Flow-based likelihood loss</td>
<td>Variational</td>
<td>Upper</td>
</tr>
<tr>
<td>VAE-based likelihood loss</td>
<td>Variational</td>
<td>Upper</td>
</tr>
</tbody>
</table>
Classifier metrics are to comparing points as distribution matching metrics are to **comparing sets**.

**Easy** - Classification accuracy is an average over **point-wise** distances.

**Hard** - DM evaluation must compute a **set-wise** distance in high dimensional space.
Conjecture: Variational bounds hold the most promise long-term.

Optimization-based metrics can leverage advances in (1) model architectures, (2) computational power, and (2) optimization, while other metrics do not benefit from these advancements.
Future research opportunities in all areas of distribution matching

<table>
<thead>
<tr>
<th>Matching fundamentals</th>
<th>Matching applications</th>
<th>Matching algorithms</th>
<th>Matching evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Conditional matching in particular</td>
<td>• Causal representation learning</td>
<td>• Stable and scalable non-adversarial methods</td>
<td>• More application-agnostic measures</td>
</tr>
<tr>
<td></td>
<td>• Domain generalization</td>
<td></td>
<td>• Rigorous evaluation protocols</td>
</tr>
</tbody>
</table>
Thanks for listening! Any questions?

(Many thanks to helpful feedback from previous audiences.)

<table>
<thead>
<tr>
<th>Application</th>
<th>Method</th>
<th>Task Loss</th>
<th>Distribution to align</th>
<th>Aligner Structure</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fair Classific.</td>
<td>Fair VAE</td>
<td>ERM</td>
<td>$p_g(z</td>
<td>d)$</td>
<td>Stochastic</td>
</tr>
<tr>
<td></td>
<td>Adversarially Fair</td>
<td></td>
<td></td>
<td>Shared</td>
<td>Adversarial</td>
</tr>
<tr>
<td></td>
<td>Fair Flows</td>
<td></td>
<td></td>
<td>Invertible</td>
<td>Flow-based</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Domain Generaliz.</td>
<td>DANN</td>
<td>ERM</td>
<td>$p_g(z</td>
<td>d)$</td>
<td>Shared</td>
</tr>
<tr>
<td>[Bai et al., 2023]</td>
<td>CDANN</td>
<td></td>
<td>$p_g(z</td>
<td>y, d)$</td>
<td>Shared</td>
</tr>
<tr>
<td></td>
<td>IRM</td>
<td></td>
<td>$p_g(y</td>
<td>z, d)$</td>
<td>Shared</td>
</tr>
<tr>
<td></td>
<td>Fishr</td>
<td></td>
<td>$p_g(\nabla_\theta L_\theta(x)</td>
<td>d)$</td>
<td>Implicit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Causality</td>
<td>CATE</td>
<td>Factual risk</td>
<td>$p_g(z</td>
<td>d)$</td>
<td>Invertible</td>
</tr>
<tr>
<td></td>
<td>ICP</td>
<td>n/a</td>
<td>$p_g(y</td>
<td>z_{\pi\alpha(y)}, d)$</td>
<td>Permutation</td>
</tr>
<tr>
<td></td>
<td>Domain</td>
<td>NLL</td>
<td>$p_g(z_i</td>
<td>z_{&lt;i}, d)$</td>
<td>Shared</td>
</tr>
<tr>
<td></td>
<td>Counterfactuals</td>
<td></td>
<td>$\forall i$ not intervened</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dist. Shift</td>
<td>Sparse transport</td>
<td>Reg. Transport</td>
<td>$p_g(z</td>
<td>d)$</td>
<td>Sparse</td>
</tr>
<tr>
<td>Explanations</td>
<td></td>
<td>Cost Transport</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Interpretable transport</td>
<td>Transport Cost</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This was work supported in part by NSF (IIS-2212097), ARL (W911NF-2020-221), and ONR (N00014-23-C-1016). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsor.