New Applications of Deep Generative Models

Jiaming Song
Stanford University

11/19/2018
Deep Generative Models
Deep Generative Models

• Images (BigGAN, Glow)
Deep Generative Models

- Images (BigGAN, Glow)
Deep Generative Models

• Images (BigGAN, Glow)

• Audio (WaveNet)
Topics

- Imitation Learning
  - Distribution matching view of imitation learning
  - Multi-Agent Generative Adversarial Imitation Learning

- Fair Representation Learning
  - Information-theoretic notions on latent variable generative models
  - Learning Controllable Fair Representations
Multi-Agent Generative Adversarial Imitation Learning

Jiaming Song
w/ Hongyu Ren, Dorsa Sadigh and Stefano Ermon

Stanford University
Reinforcement Learning

- Goal: Learn policies
Reinforcement Learning

- Goal: Learn policies
- High-dimensional, raw observations
Reinforcement Learning

- Goal: Learn policies
- High-dimensional, raw observations
Reinforcement Learning

- Goal: Learn policies
- High-dimensional, raw observations
Reinforcement Learning

- Goal: Learn policies
- High-dimensional, raw observations
Reinforcement Learning

- Goal: Learn policies
- High-dimensional, raw observations

RL needs cost signal
Imitation

(Ng and Russell, 2000), (Abbeel and Ng, 2004; Syed and Schapire, 2007), (Ratliff et al., 2006), (Ziebart et al., 2008), (Kolter et al., 2008), (Finn et al., 2016), etc.
Imitation

Input: expert behavior generated by $\pi_E$

$$\{(s_0^i, a_0^i, s_1^i, a_1^i, \ldots)\}_{i=1}^n \sim \pi_E$$

(Ng and Russell, 2000), (Abbeel and Ng, 2004; Syed and Schapire, 2007), (Ratliff et al., 2006), (Ziebart et al., 2008), (Kolter et al., 2008), (Finn et al., 2016), etc.
Imitation

Input: expert behavior generated by $\pi_E$

$$\left\{ (s_0^i, a_0^i, s_1^i, a_1^i, \ldots) \right\}_{i=1}^n \sim \pi_E$$

Goal: learn cost function (reward) or policy

(Ng and Russell, 2000), (Abbeel and Ng, 2004; Syed and Schapire, 2007), (Ratliff et al., 2006), (Ziebart et al., 2008), (Kolter et al., 2008), (Finn et al., 2016), etc.
Problem setup

\[ RL(r) = \arg \max_{\pi \in \Pi} H(\pi) + \mathbb{E}_\pi [r(s, a)] \]
Problem setup

$$RL(r) = \arg \max_{\pi \in \Pi} H(\pi) + \mathbb{E}_{\pi}[r(s, a)]$$
Problem setup

\[ RL(r) = \arg \max_{\pi \in \Pi} H(\pi) + E_{\pi}[r(s, a)] \]
Problem setup

\[ RL(r) = \arg \max_{\pi} H(\pi) + \mathbb{E}_\pi[r(s, a)] \]

(Ziebart et al., 2010; Rust 1987)
Problem setup

\[ \text{RL}(r) = \arg \max_{\pi \in \Pi} \mathcal{H}(\pi) + \mathbb{E}_{\pi}[r(s, a)] \]

\[ \text{IRL}(\pi_E) = \arg \max_{\pi \in \mathbb{R}^{S \times A}} \mathbb{E}_{\pi_E}[r(s, a)] - \left( \max_{\pi \in \Pi} \mathcal{H}(\pi) + \mathbb{E}_{\pi}[r(s, a)] \right) \]

(Ziebart et al., 2010; Rust 1987)
Problem setup

\[ RL(r) = \arg \max_{\pi \in \Pi} H(\pi) + \mathbb{E}_{\pi}[r(s, a)] \]

**Reward Function**

**Reinforcement Learning (RL)**

**Environment (MDP)**

**Inverse Reinforcement Learning (IRL)**

**Optimal policy**

\[ IRL(\pi_E) = \arg \max_{\pi \in \mathbb{R}^{S \times A}} \mathbb{E}_{\pi_E}[r(s, a)] - \left( \max_{\pi \in \Pi} H(\pi) + \mathbb{E}_{\pi}[r(s, a)] \right) \]

- **Expert has high reward**
- **Everything else have small reward**

(Ziebart et al., 2010; Rust 1987)
Problem setup

\[
\text{IRL}_\psi(\pi_E) = \arg \max_{r \in \mathbb{R}^S \times A} -\psi(r) + \mathbb{E}_{\pi_E} [r(s, a)] - \left( \max_{\pi \in \Pi} H(\pi) + \mathbb{E}_{\pi} [r(s, a)] \right)
\]
Problem setup

\[
\text{IRL}_\psi(\pi_E) = \arg\max_{r \in \mathbb{R}^S \times \mathcal{A}} -\psi(r) + \mathbb{E}_{\pi_E}[r(s, a)] - \left( \max_{\pi \in \Pi} H(\pi) + \mathbb{E}_\pi[r(s, a)] \right)
\]

Convex reward regularizer
Problem setup

\[
\text{IRL}_\psi(\pi_E) = \arg \max_{r \in \mathbb{R}^{S \times A}} -\psi(r) + \mathbb{E}_{\pi_E}[r(s, a)] - \left( \max_{\pi \in \Pi} H(\pi) + \mathbb{E}_\pi[r(s, a)] \right)
\]

Convex reward regularizer

(similar wrt $\psi$)
Combining RL and IRL

Reinforcement Learning (RL) \rightarrow \text{Optimal policy} \approx (\text{similar w.r.t. } \psi)

\psi\text{-regularized Inverse Reinforcement Learning (IRL)} \leftrightarrow \text{Expert’s Trajectories } s_0, s_1, s_2, \ldots

**Theorem:** $\psi$-regularized inverse reinforcement learning, implicitly, seeks a policy whose occupancy measure is close to the expert’s, as measured by $\psi^*$ (convex conjugate of $\psi$)
Combining RL and IRL

Theorem: $\psi$-regularized inverse reinforcement learning, implicitly, seeks a policy whose occupancy measure is close to the expert's, as measured by $\psi^*$ (convex conjugate of $\psi$)

$$RL \circ IRL_\psi(\pi_E) = \arg\min_{\pi \in \Pi} -H(\pi) + \psi^*(\rho_\pi - \rho_{\pi_E})$$
Takeaway

**Theorem:** $\psi$-regularized inverse reinforcement learning, implicitly, seeks a policy whose occupancy measure is close to the expert’s, as measured by $\psi^*$

- Typical IRL definition: finding a reward function $r$ such that the expert policy is uniquely optimal w.r.t. $r$

- Alternative view: IRL as a procedure that tries to induce a policy that matches the expert’s occupancy measure (**generative model**)
  - Generalizes existing frameworks
Generative Adversarial Imitation Learning

- Use this regularizer

\[
\psi_{GA}(c) \triangleq \begin{cases} 
    \mathbb{E}_{\pi_E} [g(c(s,a))] & \text{if } c < 0 \\
    +\infty & \text{otherwise}
\end{cases}
\]

where \( g(x) = \begin{cases} 
    -x - \log(1 - e^x) & \text{if } x < 0 \\
    +\infty & \text{otherwise}
\end{cases} \)
Generative Adversarial Imitation Learning

- $\psi^* = \text{optimal negative log-loss of the binary classification problem of distinguishing between state-action pairs of } \pi \text{ and } \pi_E$

\[
\psi^*_G \left( \rho_\pi - \rho_{\pi_E} \right) = \max_{D \in (0,1)^S \times A} \mathbb{E}_{\pi_E} \left[ \log(D(s,a)) \right] + \mathbb{E}_\pi \left[ \log(1 - D(s,a)) \right]
\]
Results

Input: driving demonstrations (TORCS Simulator)

Output policy:

From raw visual inputs

Li et al, 2017. InfoGAIL: Interpretable Imitation Learning from Visual Demonstrations
Multi-agent environments

What are the goals of these agents?
Problem setup

Cost Functions
\[ r_i(s,a_i) \]
\[ \ldots \]
\[ r_N(s,a_N) \]

MA Reinforcement Learning (MARL)

Environment (Markov Game)
Problem setup

Cost Functions
\[ r_1(s, a_1), \ldots, r_N(s, a_N) \]

MA Reinforcement Learning (MARL)

Environment (Markov Game)

Agent 1

\[ \ldots \]

Agent K
Problem setup

Cost Functions
\[ r_1(s, a_1) \]
\[ \ldots \]
\[ r_N(s, a_N) \]

MA Reinforcement Learning (MARL)

Environment (Markov Game)

Agent 1

... Agent K

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>0.0</td>
<td>10.10</td>
</tr>
<tr>
<td>L</td>
<td>10.10</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Drive on Left

Drive on Right
Problem setup

Reward Functions → MA Reinforcement Learning (MARL) → Agent Policies

Environment (Markov Game)

≈
(similar wrt ψ)
Problem setup

Reward Functions → MA Reinforcement Learning (MARL) → Agent Policies

Environment (Markov Game) ≈ (similar wrt $\psi$)

Reward Functions ← Inverse Reinforcement Learning (MAIRL) ← Expert's Trajectories $(s_0,a_0^1,...,a_0^N)$ $(s_1,a_1^1,...,a_1^N)$ ...
Problem setup

Can we design MAIRL that match occupancy measures?
Problem setup

For single agent IRL:

$$RL \circ IRL_{\psi}(\pi_E) = \arg\min_{\pi \in \Pi} -H(\pi) + \psi^*(\rho_\pi - \rho_{\pi_E})$$

Given MARL operator, can we design MAIRL such that $MARL \circ MAIRL(\pi_E)$ recovers occupancy measure?
Multi-Agent Reinforcement Learning

Markov Games: extension of MDPs for multi-agents.
- Agents optimal policy depends on other agents!
- Need alternative notions of optimality

Nash Equilibrium
- No agent can achieve higher reward by unilaterally changing its policy

\[ \forall i \in [1, N], \forall \hat{\pi}_i \neq \pi_i, E_{\pi_i, \pi_{-i}}[r_i] \geq E_{\hat{\pi}_i, \pi_{-i}}[r_i] \]
Multi-Agent Reinforcement Learning

Finding a Nash Equilibrium can be formalized into:

$$\min_{\pi \in \Pi, v \in \mathbb{R}^{S \times N}} f_r(\pi, v) = \sum_{i=1}^{N} \left( \sum_{s \in S} v_i(s) - \mathbb{E}_{a_i \sim \pi_i(\cdot|s)} q_i(s, a_i) \right)$$

s.t. $v_i(s) \geq q_i(s, a_i) \triangleq \mathbb{E}_{\pi_{-i}} \left[ r_i(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) v_i(s') \right]$
Multi-Agent Reinforcement Learning

Finding a Nash Equilibrium can be formalized into:

\[
\min_{\pi \in \Pi, \nu \in \mathbb{R}^{S \times N}} f_r(\pi, \nu) = \sum_{i=1}^{N} \left( \sum_{s \in S} v_i(s) - \mathbb{E}_{a_i \sim \pi_i(\cdot|s)} q_i(s, a_i) \right)
\]

s.t. \( v_i(s) \geq q_i(s, a_i) \triangleq \mathbb{E}_{\pi_{-i}} \left[ r_i(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) v_i(s') \right] \)
Multi-Agent Reinforcement Learning

Finding a Nash Equilibrium can be formalized into:

\[
\min_{\pi \in \Pi, \nu \in \mathbb{R}^{S \times N}} f_r(\pi, \nu) = \sum_{i=1}^{N} \left( \sum_{s \in S} v_i(s) - \mathbb{E}_{a_i \sim \pi_i(\cdot|s)} q_i(s, a_i) \right)
\]

s.t. \( v_i(s) \geq q_i(s, a_i) \triangleq \mathbb{E}_{\pi_{-i}} \left[ r_i(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) v_i(s') \right] \)

Reward by deviating from current policy

“Bellman Equation”
Multi-Agent Reinforcement Learning

Finding a Nash Equilibrium can be formalized into:

\[
\min_{\pi \in \Pi, \nu \in \mathbb{R}^{S \times N}} f_r(\pi, \nu) = \sum_{i=1}^{N} \left( \sum_{s \in S} \nu_i(s) - \mathbb{E}_{a_i \sim \pi_i(\cdot | s)} q_i(s, a_i) \right)
\]

\[
\text{s.t. } v_i(s) \geq q_i(s, a_i) \triangleq \mathbb{E}_{\pi_{-i}} \left[ r_i(s, a) + \gamma \sum_{s' \in S} P(s' | s, a) v_i(s') \right]
\]

Reward by deviating from current policy

Objective: find value function via Value Iteration
Constraints: policy has to satisfy Nash Equilibrium
Multi-Agent Inverse Reinforcement Learning

- Assume expert is (unique) Nash under the proposed reward function.
- Expert is the (unique) global optimizer for the primal problem
- For the Lagrangian

\[ g(\pi) = \max_{\lambda \geq 0} f_r(\pi, v) + \sum_{i=1}^{N} \sum_{s, a_i} \lambda_{s, a_i} (q_i(s, a_i) - v_i(s)) \]

Then \( g(\pi_E) = 0, \quad g(\pi) = \infty \)

Not useful for comparing distances between policies!
Multi-Agent Inverse Reinforcement Learning

\[ IRL(\pi_E) = \arg \max_{\pi \in \Pi} \mathbb{E}_{\pi} \left[ r(s, a) \right] - \left( \max_{\pi \in \Pi} H(\pi) + \mathbb{E}_{\pi} \left[ r(s, a) \right] \right) \]

- \( g(.) \) is unsuited for computing the “distance” between policies
- We construct a “smooth lower bound” of \( g \), by choosing specific Lagrange multipliers
Step 1: Equivalent Constraints
Step 1: Equivalent Constraints

Expand the 1-step constraint to k-step constraint
Step 1: Equivalent Constraints

Expand the 1-step constraint to k-step constraint
Recall TD learning: difference with 1-step return
Step 1: Equivalent Constraints

Expand the 1-step constraint to k-step constraint
Recall TD learning: difference with 1-step return

$$r_t + \gamma \mathbb{E}_\pi [V(s_{t+1})]$$
Step 1: Equivalent Constraints

Expand the 1-step constraint to k-step constraint
Recall TD learning: difference with 1-step return

\[ r_t + \gamma \mathbb{E}_\pi[V(s_{t+1})] \]

k-step TD target:
Step 1: Equivalent Constraints

Expand the 1-step constraint to k-step constraint

Recall TD learning: difference with 1-step return

\[ r_t + \gamma \mathbb{E}_\pi [V(s_{t+1})] \]

k-step TD target:

\[ \sum_{j=0}^{k-1} \gamma^j r_{t+j} + \gamma^k \mathbb{E}_\pi [V(s_{t+k})] \]
Step 1: Equivalent Constraints

Define $Q_i^{(k)}(\{s^j, a_i^j\}_{j=0}^{k-1}, a_i^k)$ as discounted return
- For agent $i$
- When agent $i$ took and visited $\{s^j, a_i^j\}_{j=0}^{k-1}$
- And other agents act according to their policies

Change constraints to

$$v_i(s^{(0)}) \geq Q_i^{(k)}(\{s^{(j)}, a_i^{(j)}\}_{j=0}^{k-1}, a_i^{(k)})$$

Still ensures Nash Equilibrium!
Step 2: Find the Lagrange Multipliers
Step 2: Find the Lagrange Multipliers

Consider the Lagrangian

$$\max_{\lambda \geq 0} \min_{\pi} L^{(t+1)}_\pi (\pi, \lambda) \triangleq \sum_{i=1}^{N} \sum_{\tau_i \in T^i_t} \lambda(\tau_i) \left( Q^{(t)}_i (\tau_i; \pi, r) - v_i(s^{(0)}; \pi, r) \right)$$
Step 2: Find the Lagrange Multipliers

Consider the Lagrangian

$$\max_{\lambda \geq 0} \min_{\pi} L_{\pi}^{(t+1)}(\pi, \lambda) \triangleq \sum_{i=1}^{N} \sum_{\tau_i \in T_i^t} \lambda(\tau_i) \left( Q_i^{(t)}(\tau_i; \pi, r) - v_i(s^{(0)}; \pi, r) \right)$$

Length $t$ trajectories
Step 2: Find the Lagrange Multipliers

Consider the Lagrangian

$$\max_{\lambda \geq 0} \min_{\pi} L_x^{(t+1)}(\pi, \lambda) \triangleq \sum_{i=1}^{N} \sum_{\tau_i \in T_i^t} \lambda(\tau_i) \left( Q_i^{(t)}(\tau_i; \pi, r) - v_i(s^{(0)}; \pi, r) \right)$$

For any two policies $\pi^*$ and $\pi$

Let $\lambda^*_\pi(\tau_i)$ be the probability of generating the sequence with $(\pi_i, \pi^*_i)$, then
Step 2: Find the Lagrange Multipliers

Consider the Lagrangian

$$\max_{\lambda \geq 0} \min_{\pi} \quad L^{(t+1)}_x (\pi, \lambda) \triangleq \sum_{i=1}^{N} \sum_{\tau_i \in T_i^t} \lambda(\tau_i) \left( Q_i^{(t)}(\tau_i; \pi, r) - v_i(s^{(0)}; \pi, r) \right)$$

Length \( t \) trajectories

For any two policies \( \pi^* \) and \( \pi_i \)

Let \( \lambda^*_\pi(\tau_i) \) be the probability of generating the sequence with \( (\pi_i, \pi^*_\pi) \), then

$$\lim_{t \to \infty} L^{(t+1)}_x (\pi^*, \lambda^*_\pi) = \sum_{i=1}^{N} \left( \mathbb{E}_{\pi_i, \pi^*_\pi} [r_i(s, a)] - \mathbb{E}_{\pi^*_i, \pi^*_\pi} [r_i(s, a)] \right)$$
MAIRL Operator

This motivates the following MAIRL operator

$$\arg\max_r -\psi(r) + \sum_{i=1}^N (\mathbb{E}_{\pi_i} [r_i]) - \left( \max_{\pi} \sum_{i=1}^N (\beta H_i(\pi_i) + \mathbb{E}_{\pi_i,\pi_{-i}} [r_i]) \right)$$

Strictly generalizes the IRL operator when $N=1$!
MAIRL Operator

This motivates the following MAIRL operator

$$\arg \max_{\mathbf{r}} -\psi(\mathbf{r}) + \sum_{i=1}^{N} (\mathbb{E}_{\pi^{i}}[r^{i}_{i}]) \quad \left( \max_{\pi} \sum_{i=1}^{N} (\beta H_{i}(\pi^{i}) + \mathbb{E}_{\pi^{i},\pi_{-i}}[r^{i}_{-i}]) \right)$$

Expert has high reward

Strictly generalizes the IRL operator when $N=1$!
MAIRL Operator

This motivates the following MAIRL operator

$$\arg \max_r -\psi(r) + \sum_{i=1}^N (\mathbb{E}_{\pi_E}[r_i]) - \left( \max_{\pi} \sum_{i=1}^N (\beta H_i(\pi_i) + \mathbb{E}_{\pi_i, \pi_{E-i}}[r_i]) \right)$$

- **Expert has high reward**
- **Everything else have small reward**

Strictly generalizes the IRL operator when $N=1$!
Multi-Agent Imitation Learning

Assume that reward function is additively separable

$$\psi(r) = \sum_{i=1}^{N} \psi_i(r_i)$$

Then

$$\text{MARL} \circ \text{MAIRL}_{\psi}(\pi_E) = \arg\min_{\pi \in \Pi} \sum_{i=1}^{N} -\beta H_i(\pi_i) + \psi_i^*(\rho_{\pi_i, E-i} - \rho_{\pi_E})$$
MAGAIL

Sample from expert $(s,a_1,a_2,...,a_N)$

Sample from model $(s,a_1,a_2,...,a_N)$

Black box simulator

Generator $G$

Policy Agent 1

Policy Agent N

Song, Ren, Sadigh, Ermon, Multi-Agent Generative Adversarial Imitation Learning
MAGAIL

Incorporate knowledge on reward structure via regularizers
MAGAIL

Incorporate knowledge on reward structure via regularizers

Centralized

Decentralized

Zero-sum
MAGAIL

Incorporate knowledge on reward structure via regularizers

Higher rewards

Centralized

Decentralized

Zero-sum
MAGAIL

Incorporate knowledge on reward structure via regularizers

Higher rewards

Centralized

Decentralized

Zero-sum
Experiments

Table 1: Average agent rewards in competitive tasks. We compare behavior cloning (BC), GAIL (G), Centralized (C), Decentralized (D), and Zero-Sum (ZS) methods. Best marked in bold (high vs. low rewards is preferable depending on the agent vs. adversary role).

<table>
<thead>
<tr>
<th>Task</th>
<th>Predator-Prey</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Agent</strong></td>
<td><strong>Behavior Cloning</strong></td>
<td><strong>G</strong></td>
<td><strong>C</strong></td>
<td><strong>D</strong></td>
<td><strong>ZS</strong></td>
</tr>
<tr>
<td><strong>Adversary</strong></td>
<td><strong>BC</strong></td>
<td><strong>G</strong></td>
<td><strong>C</strong></td>
<td><strong>D</strong></td>
<td><strong>ZS</strong></td>
</tr>
<tr>
<td><strong>Rewards</strong></td>
<td><strong>-93.20</strong></td>
<td><strong>-93.71</strong></td>
<td><strong>-93.75</strong></td>
<td><strong>-95.22</strong></td>
<td><strong>-95.48</strong></td>
</tr>
<tr>
<td><strong>Task</strong></td>
<td><strong>Keep-Away</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Agent</strong></td>
<td><strong>Behavior Cloning</strong></td>
<td><strong>G</strong></td>
<td><strong>C</strong></td>
<td><strong>D</strong></td>
<td><strong>ZS</strong></td>
</tr>
<tr>
<td><strong>Adversary</strong></td>
<td><strong>BC</strong></td>
<td><strong>G</strong></td>
<td><strong>C</strong></td>
<td><strong>D</strong></td>
<td><strong>ZS</strong></td>
</tr>
<tr>
<td><strong>Rewards</strong></td>
<td><strong>24.22</strong></td>
<td><strong>24.04</strong></td>
<td><strong>23.28</strong></td>
<td><strong>23.56</strong></td>
<td><strong>23.19</strong></td>
</tr>
</tbody>
</table>
Suboptimal demos

Expert
Conclusions
Conclusions

- IRL is a dual of an occupancy measure matching problem (generative modeling)
Conclusions

• IRL is a dual of an occupancy measure matching problem (generative modeling)
• Multi-agent cases are complicated by alternative optimality notions (e.g. Nash Equilibrium)
Conclusions

- IRL is a dual of an occupancy measure matching problem (generative modeling)
- Multi-agent cases are complicated by alternative optimality notions (e.g. Nash Equilibrium)
- Under certain cases, we can link multi-agent imitation learning to occupancy measure matching
Conclusions

• IRL is a dual of an occupancy measure matching problem (generative modeling)
• Multi-agent cases are complicated by alternative optimality notions (e.g. Nash Equilibrium)
• Under certain cases, we can link multi-agent imitation learning to occupancy measure matching
• Limitations exist (e.g. zero-sum)
Learning Fair and Controllable Representations

Jiaming Song

w/ Ria Kalluri, Aditya Grover, Shengjia Zhao, Stefano Ermon

Stanford University
Problem setup

- X: features of an individual
Problem setup

- $X$: features of an individual
- $U$: sensitive attribute (e.g. gender)
Problem setup

- $X$: features of an individual
- $U$: sensitive attribute (e.g. gender)
- $C = c(X, U)$ predicted values
Problem setup

- X: features of an individual
- U: sensitive attribute (e.g. gender)
- C = c(X, U) predicted values
- Y: target variable (labels)
Problem setup

- $X$: features of an individual
- $U$: sensitive attribute (e.g. gender)
- $C = c(X, U)$ predicted values
- $Y$: target variable (labels)
- $Z$: representation (used for downstream tasks)
Problem setup

- $X$: features of an individual
- $U$: sensitive attribute (e.g. gender)
- $C = c(X, U)$ predicted values
- $Y$: target variable (labels)
- $Z$: representation (used for downstream tasks)

Make accurate predictions while protecting $U$. 
Problem setup

- $X$: features of an individual
- $U$: sensitive attribute (e.g. gender)
- $C = c(X, U)$ predicted values
- $Y$: target variable (labels)
- $Z$: representation (used for downstream tasks)

Make accurate predictions while protecting $U$.
- “Give loan according to credit but fair to race”
Examples of Fairness Notions
Examples of Fairness Notions

- Demographic parity
  - $C$ and $U$ are independent.
  - $I(C; U) = 0$
Examples of Fairness Notions

- Demographic parity
  - C and U are independent.
  - I(C; U) = 0

- Equalized odds
  - C and U are independent conditional on Y
  - I(C; U|Y) = 0
Examples of Fairness Notions

• Demographic parity
  • C and U are independent.
  • \( I(C; U) = 0 \)

• Equalized odds
  • C and U are independent conditional on Y
  • \( I(C; U|Y) = 0 \)

• Equalized opportunity
  • C and U are independent conditional on \( Y = 1 \)
  • \( I(C; U|Y=1) = 0 \)
Examples of Fairness Notions

• Demographic parity
  • C and U are independent.
  • $I(C; U) = 0$

• Equalized odds
  • C and U are independent conditional on Y
  • $I(C; U|Y) = 0$

• Equalized opportunity
  • C and U are independent conditional on $Y = 1$
  • $I(C; U|Y=1) = 0$

Not all notions can be satisfied at once!
Representation Learning
Representation Learning

- Learn a representation $Z$, and use $Z$ to predict $Y$
Representation Learning

- Learn a representation $Z$, and use $Z$ to predict $Y$
- “Data Preprocessing”
Learning Fair and Expressive Representations

$X, U \rightarrow Z \rightarrow C$

$max \ I(X; Z|U) \quad min \ I(U; Z)$
Optimization Problem
Optimization Problem

- Maximize “expressiveness”
Optimization Problem

- Maximize “expressiveness”
- Under “fairness” constraints
Optimization Problem

- Maximize “expressiveness”
- Under “fairness” constraints
- Prediction model
Optimization Problem

- Maximize "expressiveness"
- Under "fairness" constraints
- Prediction model $q_{\phi}(z|x, u)$
Optimization Problem

• Maximize “expressiveness”
• Under “fairness” constraints
• Prediction model $q_{\phi}(z|x, u)$
• Data distribution
Optimization Problem

- Maximize “expressiveness”
- Under “fairness” constraints
- Prediction model \( q_\phi(z|x, u) \)
- Data distribution \( q(x, u) \)
Optimization Problem

- Maximize "expressiveness"
- Under "fairness" constraints
- Prediction model $q_\phi(z|x, u)$
- Data distribution $q(x, u)$
- Constrained optimization problem
  - e.g. demographic parity
Optimization Problem

- Maximize “expressiveness”
- Under “fairness” constraints
- Prediction model $q_\phi(z|x, u)$
- Data distribution $q(x, u)$
- Constrained optimization problem
  - e.g. demographic parity

$$\max I_q(x; z | u)$$
$$\text{s.t. } I_q(z; u) < \epsilon$$
Optimization Problem

• Maximize “expressiveness”
• Under “fairness” constraints
• Prediction model $q_\phi(z|x, u)$
• Data distribution $q(x, u)$
• Constrained optimization problem
  • e.g. demographic parity

$$\max I_q(x; z|u)$$
$$\text{s.t. } I_q(z; u) < \epsilon$$

Quantity determined by user (hyperparameter)
Tractable Bounds for Mutual Information
Tractable Bounds for Mutual Information

These quantities are not “tractable”!
Tractable Bounds for Mutual Information

These quantities are not "tractable"

\[ I_q(x; z|u) = E_{q(x,z,u)}[\log q(x, z|u) - \log q(x|u) - \log q(z|u)] \]
Tractable Bounds for Mutual Information

These quantities are not “tractable”!

$$I_q(x; z|u) = E_{q_{\phi}(x,z,u)}[\log q_{\phi}(x, z|u) - \log q(x|u) - \log q_{\phi}(z|u)]$$

$$I_q(z; u) = E_{q_{\phi}(z,u)}[\log q_{\phi}(z|u) - \log q_{\phi}(z)]$$
Tractable Bounds for Mutual Information

These quantities are not "tractable"!

\[ I_q(x; z|u) = \mathbb{E}_{q_{\phi}(x, z, u)}[\log q_{\phi}(x, z|u) - \log q(x|u) - \log q_{\phi}(z|u)] \]

\[ I_q(z; u) = \mathbb{E}_{q_{\phi}(z, u)}[\log q_{\phi}(z|u) - \log q_{\phi}(z)] \]

Only \( q_{\phi}(z|x, u) \) has tractable log density!
\[
\max I_q(x; z|u)
\]

- Introduce a parametrized distribution

Fixing a Broken ELBO, Alemi et al. ICML 2018.
$$\max I_q(x; z|u)$$

- Introduce a parametrized distribution $p_\theta(x|z, u)$
\[
\max I_q(x; z|u)
\]

- Introduce a parametrized distribution \( p_\theta(x|z, u) \)

\[
I_q(x; z|u) = E_{q_\phi(x, z, u)}[\log p_\theta(x|z, u)] + H_q(x|u) + E_{q_\phi(z, u)}D_{KL}(q_\phi(x|z, u) || p_\theta(x|z, u))
\]

Fixing a Broken ELBO, Alemi et al. ICML 2018.
\[
\max I_q(x; z|u)
\]

- Introduce a parametrized distribution \( p_\theta(x|z, u) \)

\[
I_q(x; z|u) = E_{q_\phi(x,z,u)}[\log p_\theta(x|z, u)] + H_q(x|u) + E_{q_\phi(z,u)} D_{KL}(q_\phi(x|z, u) \| p_\theta(x|z, u))
\]

"Constant Entropy"

Fixing a Broken ELBO, Alemi et al. ICML 2018.
$\max I_q(x; z|u)$

- Introduce a parametrized distribution $p_\theta(x|z, u)$

$I_q(x; z|u) = E_{q_\phi(x,z,u)}[\log p_\theta(x|z,u)] + H_q(x|u) + E_{q_\phi(z,u)} D_{KL}(q_\phi(x|z,u) || p_\theta(x|z,u))$ 

"Constant Entropy"  

KL divergence $> 0$

Fixing a Broken ELBO, Alemi et al. ICML 2018.
\[
\max I_q(x; z|u)
\]

- Introduce a parametrized distribution \( p_\theta(x|z, u) \)

\[
I_q(x; z|u) = E_{q_\phi(x, z, u)}[\log p_\theta(x|z, u)] + H_q(x|u) + E_{q_\phi(z, u)} D_{KL}(q_\phi(x|z, u)||p_\theta(x|z, u))
\]

"Reconstruction Error"  
"Distortion"  
"Constant Entropy"  
KL divergence > 0

Fixing a Broken ELBO, Alemi et al. ICML 2018.
\[
\max I_q(x; z|u)
\]

- Introduce a parametrized distribution \( p_\theta(x|z, u) \)

\[
I_q(x; z|u) = E_{q_\phi(x,z,u)}[\log p_\theta(x|z, u)] + H_q(x|u) + E_{q_\phi(z,u)} D_{KL}(q_\phi(x,z,u) \parallel p_\theta(x|z, u))
\]

"Reconstruction Error"  "Distortion"  "Constant Entropy"  KL divergence > 0

Use "Reconstruction Error" as lower bound (up to constant)!

Fixing a Broken ELBO, Alemi et al. ICML 2018.
\[ \min I_q(u, z) \]
\[ \min I_q(u, z) \]

\[ I_q(z; u) \leq I_q(z; x, u) = \mathbb{E}_{q(x; u)} D_{KL}(q_\phi(z | x, u) || p(z)) - D_{KL}(q_\phi(z) || p(z)) \]
\[ \min I_q(u, z) \]

\[
I_q(z; u) \leq I_q(z; x, u) = \mathbb{E}_{q(x, u)} D_{KL}(q(z|x, u)||p(z)) - D_{KL}(q_\phi(z)||p(z))
\]

Add additional variable

A Lagrangian Perspective of Latent Variable
Generative Models, Zhao, Song, Ermon. UAI 2018.
\[ \min I_q(u, z) \]

\[ I_q(z; u) \leq I_q(z; x, u) = \mathbb{E}_{q(x, u)} D_{KL}(q_\phi(z|x, u) || p(z)) - D_{KL}(q_\phi(z) || p(z)) \]

Add additional variable

KL divergence > 0
$\min I_q(u, z)$

$I_q(z; u) \leq I_q(z; x, u) = \mathbb{E}_{q(x, u)} D_{KL}(q_{\phi}(z|x, u) \| p(z)) - D_{KL}(q_{\phi}(z) \| p(z))$

Add additional variable    "Rate"    KL divergence $> 0$

Upper Bound
\[
\min \ I_q(u, z)
\]
\[
I_q(z; u) \leq I_q(z; x, u) = \mathbb{E}_{q(x; u)} D_{\text{KL}}(q_\phi(z|x, u)\|p(z)) - D_{\text{KL}}(q_\phi(z)\|p(z))
\]

Add additional variable

"Rate"

Upper Bound

KL divergence > 0

\[
I_q(x; z|u)
\]

A Lagrangian Perspective of Latent Variable Generative Models, Zhao, Song, Ermon. UAI 2018.
\[ \min I_q(u, z) \]

\[ I_q(z; u) \leq I_q(z; x, u) = \mathbb{E}_{q(x, u)} D_{KL}(q_\phi(z|x, u)||p(z)) - D_{KL}(q_\phi(z)||p(z)) \]

Add additional variable  
"Rate"  
Upper Bound  
KL divergence > 0

\[ \mathbb{E}_{q_\phi(x, z, u)} \left[ \log p_\theta(x|z, u) \right] \]

A Lagrangian Perspective of Latent Variable Generative Models, Zhao, Song, Ermon. UAI 2018.
\[
\min I_q(u, z)
\]

\[
I_q(z; u) \leq I_q(z; x, u) = \mathbb{E}_{q(x, u)} D_{KL}(q_{\phi}(z| x, u) \| p(z)) - D_{KL}(q_{\phi}(z) \| p(z))
\]

Add additional variable  
“Rate”  
Upper Bound  
KL divergence > 0

\[
\mathbb{E}_{q_{\phi}(x, z, u)} [\log p_{\theta}(x| z, u)]
\]

\[
I_q(x; z| u)
\]

\[
\mathbb{E}_{q(x, u)} D_{KL}(q_{\phi}(z| x, u) \| p(z))
\]

A Lagrangian Perspective of Latent Variable Generative Models, Zhao, Song, Ermon. UAI 2018.
\[
\min I_q(u, z)
\]

\[
I_q(z; u) \leq I_q(z; x, u) = \mathbb{E}_{q(x,u)} D_{KL}(q_{\phi}(z|x, u) \| p(z)) - D_{KL}(q_{\phi}(z) \| p(z))
\]

Add additional variable

"Rate"

Upper Bound

KL divergence $> 0$

A Lagrangian Perspective of Latent Variable
Generative Models, Zhao, Song, Ermon. UAI 2018.
\[ \min I_q(u, z) \]
\[ \min I_q(u, z) \]

The "rate" upper bound is not tight!

Tighter version with some \( p(u) \):
\[
\min I_q(u, z)
\]

The "rate" upper bound is not tight!

Tighter version with some \( p(u) \):

\[
I_q(z; u) = E_{q_\phi(z)} D_{KL}(q_\phi(u|z)||p(u)) - D_{KL}(q(u)||p(u))
\]
\[
\min I_q(u, z)
\]

The “rate” upper bound is not tight!

Tighter version with some \( p(u) \):

\[
I_q(z; u) = \mathbb{E}_{q_\phi(z)} D_{KL}(q_\phi(u|z)||p(u)) - D_{KL}(q(u)||p(u))
\]

Upper bound, but we don’t know \( q_\phi(u|z) \)
\[
\min I_q(u, z)
\]

The “rate” upper bound is not tight!

Tighter version with some \( p(u) \):

\[
I_q(z; u) = \mathbb{E}_{q_\phi(z)} D_{KL}(q_\phi(u|z) \| p(u)) - D_{KL}(q(u) \| p(u))
\]

Upper bound, but we don’t know \( q_\phi(u|z) \)

Approximate \( q_\phi(u|z) \) with \( p_\psi(u|z) \) by maximum likelihood!
“Adversarial Training”
“Adversarial Training”

Approximate $q_\phi(u|z)$ with $p_\psi(u|z)$ by maximum likelihood!
“Adversarial Training”

Approximate $q_\phi(u|z)$ with $p_\psi(u|z)$ by maximum likelihood!

$$\min_{\psi} \mathbb{E}_{q_\phi(z)} D_{KL}(q_\phi(u|z) \| p_\psi(u|z))$$
“Adversarial Training”

Approximate $q_\phi(u|z)$ with $p_\psi(u|z)$ by maximum likelihood!

$$\min_\psi \mathbb{E}_{q_\phi(z)} D_{KL}(q_\phi(u|z) \| p_\psi(u|z))$$

Replace q with p, we have
"Adversarial Training"

Approximate \( q_{\phi}(u|z) \) with \( p_{\psi}(u|z) \) by maximum likelihood!

\[
\min_{\psi} \mathbb{E}_{q_{\phi}(z)} D_{KL}(q_{\phi}(u|z) \| p_{\psi}(u|z))
\]

Replace \( q \) with \( p \), we have

\[
\mathbb{E}_{q_{\phi}(z,u)} [\log p_{\psi}(u|z) - \log p(u)]
= \mathbb{E}_{q_{\phi}(z)} [D_{KL}(q_{\phi}(u|z) \| p(u)) - D_{KL}(q_{\phi}(u|z) \| p_{\psi}(u|z))]
\leq \mathbb{E}_{q_{\phi}(z)} D_{KL}(q_{\phi}(u|z) \| p(u))
\]
“Adversarial Training”

Approximate $q_\phi(u|z)$ with $p_\psi(u|z)$ by maximum likelihood!

$$\min_{\psi} \mathbb{E}_{q_\phi(z)} D_{KL}(q_\phi(u|z) \| p_\psi(u|z))$$

Replace $q$ with $p$, we have

$$\mathbb{E}_{q_\phi(z,u)} [\log p_\psi(u|z) - \log p(u)]$$

$$= \mathbb{E}_{q_\phi(z)} [D_{KL}(q_\phi(u|z) \| p(u)) - D_{KL}(q_\phi(u|z) \| p_\psi(u|z))]$$

$$\leq \mathbb{E}_{q_\phi(z)} D_{KL}(q_\phi(u|z) \| p(u)) \quad \text{“Maximum likelihood”}$$
“Adversarial Training”

Approximate $q_\phi(u|z)$ with $p_\psi(u|z)$ by maximum likelihood!

$$\min_{\psi} \mathbb{E}_{q_\phi(z)} D_{\text{KL}}(q_\phi(u|z) \parallel p_\psi(u|z))$$

Replace $q$ with $p$, we have

$$\mathbb{E}_{q_\phi(z,u)} [\log p_\psi(u|z) - \log p(u)]$$

"Lower bound to upper bound"

$$= \mathbb{E}_{q_\phi(z)} [D_{\text{KL}}(q_\phi(u|z) \parallel p(u)) - D_{\text{KL}}(q_\phi(u|z) \parallel p_\psi(u|z))]$$

$$\leq \mathbb{E}_{q_\phi(z)} D_{\text{KL}}(q_\phi(u|z) \parallel p(u))$$

"Maximum likelihood"
“Adversarial Training”

Approximate $q_\phi(u|z)$ with $p_\psi(u|z)$ by maximum likelihood!

$$\min_{\psi} \mathbb{E}_{q_\phi(z)} D_{KL}(q_\phi(u|z) \parallel p_\psi(u|z))$$

Replace $q$ with $p$, we have

$$\mathbb{E}_{q_\phi(z,u)} [\log p_\psi(u|z) - \log p(u)]$$

“Lower bound to upper bound”

$$= \mathbb{E}_{q_\phi(z)} [D_{KL}(q_\phi(u|z) \parallel p(u)) - D_{KL}(q_\phi(u|z) \parallel p_\psi(u|z))]$$

“Maximum likelihood”

$$\leq \mathbb{E}_{q_\phi(z)} D_{KL}(q_\phi(u|z) \parallel p(u))$$

Maximum likelihood --> Gap with upper bound = 0!
“Adversarial Training” Intuition
“Adversarial Training” Intuition

Classifier: predict $u$ from $z$. 
“Adversarial Training” Intuition

Classifier: predict $u$ from $z$.
Representation: prevent classifier to predict $u$. 
“Adversarial Training” Intuition

Classifier: predict $u$ from $z$.
Representation: prevent classifier to predict $u$.

- Adversarial training but not a GAN!
"Adversarial Training" Intuition

Classifier: predict u from z.
Representation: prevent classifier to predict u.

- Adversarial training but not a GAN!
- Allows for any type of u (as opposed to binary)
“Tractable” Objective
“Tractable” Objective

\[
\min_{\theta, \phi} \max_{\psi \in \Psi} \quad \mathcal{L}_r = -\mathbb{E}_{q_{\phi}(x, z, u)}[\log p_{\theta}(x|z, u)]
\]

s.t. \[
C_1 = \mathbb{E}_{q(x, u)} D_{KL}(q_{\phi}(z|x, u) || p(z)) < \epsilon_1
\]
\[
C_2 = \mathbb{E}_{q_{\phi}(z, u)}[\log p_{\psi}(u|z) - \log p(u)] < \epsilon_2
\]
Dual Formulation
Dual Formulation

$$\arg\min_{\theta, \phi} \max_\psi \mathcal{L}_{r} + \lambda_1 (C_1 - \varepsilon_1) + \lambda_2 (C_2 - \varepsilon_2)$$
Dual Formulation

\[
\arg \min_{\theta, \phi} \max_{\psi} \mathcal{L}_r + \lambda_1 (C_1 - \epsilon_1) + \lambda_2 (C_2 - \epsilon_2)
\]

Existing works optimize this objective with fixed lambda!
Dual Formulation

\[ \arg\min_{\theta, \phi} \max_{\psi} \mathcal{L}_r + \lambda_1 (C_1 - \epsilon_1) + \lambda_2 (C_2 - \epsilon_2) \]

Existing works optimize this objective with fixed lambda!
- Variational Fair Autoencoder: \( \lambda_1 = 1 \); MMD on two groups of \( z \).
Dual Formulation

\[ \arg \min_{\theta, \phi} \max_{\psi} \mathcal{L}_r + \lambda_1 (C_1 - \epsilon_1) + \lambda_2 (C_2 - \epsilon_2) \]

Existing works optimize this objective with fixed lambda!
- Variational Fair Autoencoder: $\lambda_1 = 1$; MMD on two groups of $z$.
- Adversarial Censoring: $\lambda_1 = 0$; GAN on two groups of $x$. 
Dual Formulation

\[ \arg \min_{\theta, \phi} \max_{\psi} \mathcal{L}_r + \lambda_1 (C_1 - \epsilon_1) + \lambda_2 (C_2 - \epsilon_2) \]

Existing works optimize this objective with fixed lambda!
- Variational Fair Autoencoder: \( \lambda_1 = 1 \); MMD on two groups of \( z \).
- Adversarial Censoring: \( \lambda_1 = 0 \); GAN on two groups of \( x \).
- Fair and Transferrable Representations: \( \lambda_1 = 0 \); GAN on two groups of \( z \).
Dual Optimization
Dual Optimization

Straightforward to use dual optimization
Dual Optimization

Straightforward to use dual optimization

\[
\max_{\lambda_1, \lambda_2 \geq 0} \min_{\theta, \phi, \psi} \max \mathcal{L} = \mathcal{L}_r + \lambda_1^\top (C_1 - \epsilon_1) + \lambda_2^\top (C_2 - \epsilon_2)
\]
Dual Optimization

Straightforward to use dual optimization

$$\max_{\lambda_1, \lambda_2 \geq 0} \min_{\theta, \phi, \psi} \max L = L_r + \lambda_1^T (C_1 - \epsilon_1) + \lambda_2^T (C_2 - \epsilon_2)$$

If constraint is violated, increase its weight
Dual Optimization

Straightforward to use dual optimization

\[
\max_{\lambda_1, \lambda_2 \geq 0} \min_{\theta, \phi, \psi} \max_{\lambda_1, \lambda_2 \geq 0} \mathcal{L} = \mathcal{L}_r + \lambda_1^\top (C_1 - \epsilon_1) + \lambda_2^\top (C_2 - \epsilon_2)
\]

If constraint is violated, increase its weight

• Find the trade-off between fairness and expressiveness!
Dual Optimization

Straightforward to use dual optimization

\[
\max_{\lambda_1, \lambda_2 \geq 0} \min_{\theta, \phi, \psi} \max L = L_r + \lambda_1^T (C_1 - \epsilon_1) + \lambda_2^T (C_2 - \epsilon_2)
\]

If constraint is violated, increase its weight

- Find the trade-off between fairness and expressiveness!
- Particularly useful with multiple fairness notions!
Dual Optimization

Straightforward to use dual optimization

$$\max_{\lambda_1, \lambda_2 \geq 0} \min_{\theta, \phi} \max_{\psi} \mathcal{L} = \mathcal{L}_r + \lambda_1^T (C_1 - \epsilon_1) + \lambda_2^T (C_2 - \epsilon_2)$$

If constraint is violated, increase its weight

- Find the trade-off between fairness and expressiveness!
- Particularly useful with multiple fairness notions!
- “Find solution that is reasonable under multiple notions”
Dual Optimization

Straightforward to use dual optimization

\[
\max_{\lambda_1, \lambda_2 \geq 0} \min_{\theta, \phi, \psi} \max \mathcal{L} = \mathcal{L}_r + \lambda_1^\top (C_1 - \epsilon_1) + \lambda_2^\top (C_2 - \epsilon_2)
\]

If constraint is violated, increase its weight

- Find the trade-off between fairness and expressiveness!
- Particularly useful with multiple fairness notions!
- “Find solution that is reasonable under multiple notions”
- Allows user to control level of “unfairness” directly
Experiments
Experiments

Dual optimization can find feasible solutions!
Learning Better Representations
Learning Better Representations

- Search for most expressive representations under certain fairness constraints.
Learning Better Representations

- Search for most expressive representations under certain fairness constraints.
- MIFR: fixed multipliers, grid search over 5x5 hyperparameters, find most "expressive" feasible solution.
Learning Better Representations

- Search for most expressive representations under certain fairness constraints.
- MIFR: fixed multipliers, grid search over 5x5 hyperparameters, find most “expressive” feasible solution.
- L-MIFR: train Lagrange multipliers on-the-fly, run once.
Learning Better Representations

• Search for most expressive representations under certain fairness constraints.
• MIFR: fixed multipliers, grid search over 5x5 hyperparameters, find most “expressive” feasible solution.
• L-MIFR: train Lagrange multipliers on-the-fly, run once.
Learning Better Representations
Learning Better Representations

- Search for most expressive representations under multiple fairness constraints.
  - Demographic parity, Equalized odds, Equalized opportunity
Learning Better Representations

• Search for most expressive representations under multiple fairness constraints.
  • Demographic parity, Equalized odds, Equalized opportunity
• MIFR: greedy hyperparameter search, 12 runs.
Learning Better Representations

• Search for most expressive representations under multiple fairness constraints.
  • Demographic parity, Equalized odds, Equalized opportunity

• MIFR: greedy hyperparameter search, 12 runs.

• L-MIFR: train Lagrange multipliers on-the-fly, run once.
Learning Better Representations

• Search for most expressive representations under multiple fairness constraints.
  • Demographic parity, Equalized odds, Equalized opportunity
• MIFR: greedy hyperparameter search, 12 runs.
• L-MIFR: train Lagrange multipliers on-the-fly, run once.

|        | $I_q(x; z|u)$ | $C_1$ | $I_{DP}$ | $I_{EO}$ | $I_{EOpp}$ |
|--------|--------------|-------|----------|----------|------------|
| MIFR   | 9.34         | 9.39  | 0.09     | 0.10     | 0.07       |
| L-MIFR | 9.94         | 9.95  | 0.08     | 0.09     | 0.04       |
Learning Better Representations

- Search for most expressive representations under multiple fairness constraints.
  - Demographic parity, Equalized odds, Equalized opportunity
- MIFR: greedy hyperparameter search, 12 runs.
- L-MIFR: train Lagrange multipliers on-the-fly, run once.

|       | $I_q(x; z|u)$ | $C_1$ | $I_{DP}$ | $I_{EO}$ | $I_{EOpp}$ |
|-------|--------------|-------|----------|----------|------------|
| MIFR  | 9.34         | 9.39  | 0.09     | 0.10     | 0.07       |
| L-MIFR| 9.94         | 9.95  | 0.08     | 0.09     | 0.04       |

L-MIFR learns better representations!
Summary
Summary

- Information-theoretic view of fair representation learning
Summary

- Information-theoretic view of fair representation learning
- Connection with VAEs and GANs
Summary

- Information-theoretic view of fair representation learning
- Connection with VAEs and GANs
- Dual optimization avoids tedious hyperparameter tuning and learns trade-offs
Summary

• Information-theoretic view of fair representation learning
• Connection with VAEs and GANs
• Dual optimization avoids tedious hyperparameter tuning and learns trade-offs
• Better representation with less compute