InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets

Chen et. al *(NeurIPS 2016)*

Motivation

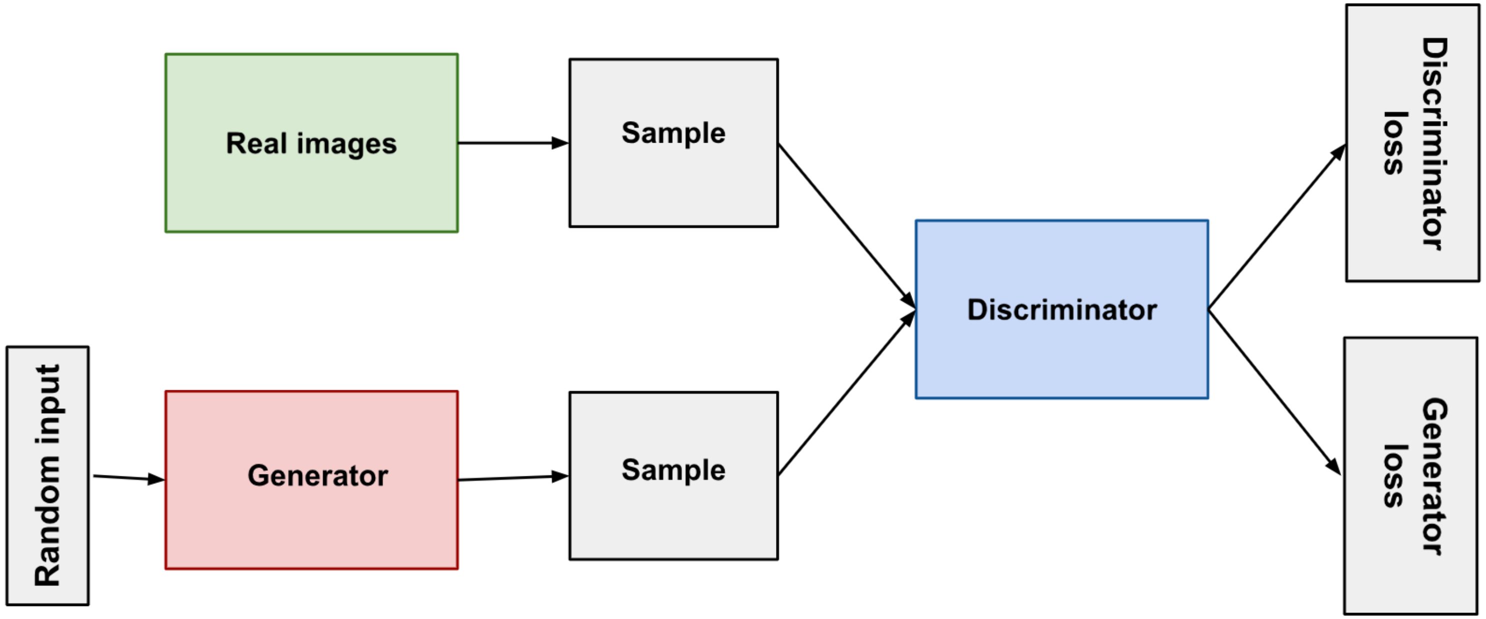
- Unsupervised representation learning for disentangled representations
    - Unsupervised Learning – extracting value from unlabeled data
    - Representation Learning – use unlabeled data to learn a representation that exposes semantic features as easily decodable factors
        - i.e., discover information that highlights meaningful features that are easier to understand and process
    - Disentangled Representation – explicit representation of the salient attributes of a data instance
        - e.g., (For a dataset of faces) facial expressions, eye color, hairstyle, etc.
- Generative modeling should be able to synthesize the observed data via automatic learning of disentangled representations

Overview

- Modify GANs (Generative Adversarial Networks) to encourage the learning of interpretable and meaningful representations.
- Maximize the mutual information between a subset of a GAN’s noise variables and the observations

Background: GANs

- Goal: Learn a generator distribution $P_G(x)$ that matches the real distribution of the data $P_{data}(x)$



Background: GANs

Define a Generator network as G_{θ_g} and a discriminator network as D_{θ_d}.

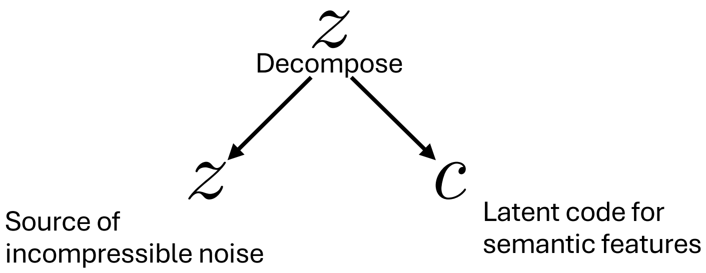
1. Define prior on input noise variable $p_z(z)$
2. Define mapping from noise to dataspace as $G_{\theta_g}(z)$
3. Generate samples as $x \sim P_G$ by transforming a noise variable $z \sim P_{noise}(z)$ into generated sampled $G_{\theta_g}(z)$
4. Train generator G_{θ_g} by maximizing the discriminator D_{θ_d} that can distinguish between samples from P_{data} and P_G
   (a) Train D_{θ_d} to maximize the probability of assigning correct label to both training examples and samples from G_{θ_g}
   (b) Train G_{θ_g} to minimize $\log(1 - D_{\theta_d}(G_{\theta_g}(z)))$

The full optimization procedure is then given by

$$\min_{G_{\theta_g}} \max_{D_{\theta_d}} V(D_{\theta_d}, G_{\theta_g}) = \mathbb{E}_{x \sim P_{data}}[\log D_{\theta_d}(x)] + \mathbb{E}_{z \sim noise}[\log(1 - D_{\theta_d}(G_{\theta_g}(z)))]$$

Mutual Information for Inducing Latent Codes

- In GANs, there are no restrictions on the noise variable z, meaning that is is highly possible the generator will use z in an entangled manner
- Ideally, the model should allocate some random variable to represent the categorical identity of the data and the sematic properties of the data in an unsupervised manner
- To this end, we decompose z into two parts
  1. z, treated as source of incompressible noise
  2. c, which is the latent code to target structured semantic features of data



z

Decompose

z

Source of incompressible noise

c

Latent code for semantic features

Mutual Information for Inducing Latent Codes

- Denote the set of structured latent codes as $\{c_1, c_2, \ldots, c_L\}$ and assume a factorized distribution

$$P(c_1, c_2, \ldots, c_L) = \prod_{i=1}^{L} P(c_i)$$

- Provide G_{θ_g} with both z and c so $G_{\theta_g}(z)$ becomes $G_{\theta_g}(z, c)$
- In standard GANs, there is no restriction to the distribution the generator can learn so the GAN can choose to model $P_G(x|c)$ as $P_G(x)$
  - I.e., the generator disregards the latent code
- To enforce the GAN to make use of the latent code we use an information theoretic regularization
  - Idea: There should be high mutual information between c and the generator distribution $G_{\theta_g}(z, c)$

A Brief History of Information Theory

- Entropy

$$H(X) = -\sum_{x} p(x) \log p(x) = \mathbb{E}\left[-\log p(x)\right]$$

- Conditional Entropy

$$H(Y|X) = \sum_{x,y} p(x,y) \log p(y|x)$$

- Mutual Information

$$I(X;Y) = H(X) - H(Y|X) = H(Y) - H(Y|X)$$

  - $I(X;Y)$ is the reduction of uncertainty in X when Y is observed
  - Equals 0 when X and Y are independent
  - If X and Y are related by a deterministic, invertible function, then maximal mutual information is attained

Modified Cost Function

- Based upon mutual information intuition, for any given $x \sim P_G(x)$, we want $P_G(c|x)$ to have small entropy
    - I.e, the information in the latent code c should not be made irrelevant during generation
- Thus, the GAN cost is reformulated as

$$\min_{G_{\theta_g}} \max_{D_{\theta_d}} V_I(D_{\theta_d}, G_{\theta_g}) = V(D_{\theta_d}, G_{\theta_g}) - \lambda I(c; G_{\theta_g}(x, c))$$

Mutual Information Maximization

- In practice, $I(c; G_{\theta_g}(z, c))$ is hard to maximize as we need access to the posterior p$(c|x)$.
- We instead define an auxiliary distribution q$(c|x)$ to approximate p$(c|x)$

$$
\begin{aligned}
I(c; G_{\theta_g}(z,c)) &= H(c) - H(c|G_{\theta_g}(z,c)) \\
&= H(c) + \sum_{c,x \sim G_{\theta_g}(z,c)} p(c,x) \log p(c|x) \\
&= H(c) + \sum_{c,x} p(x) p(c|x) \log p(c|x) \\
&= H(c) + \mathbb{E}_{x \sim G_{\theta_g}(z,c)} \left[\sum_{c} p(c|x) \log p(c|x) \right] \\
&= H(c) + \mathbb{E}_{x \sim G_{\theta_g}(z,c)} [\mathbb{E}_{c' \sim p(c|x)} [\log p(c'|x)]] \\
&= H(c) + \mathbb{E}_{x \sim G_{\theta_g}(z,c)} [D_{KL}(p(\cdot|x) || q(\cdot|x)) + \mathbb{E}_{c' \sim p(c|x)} [\log q(c'|x)]] \\
&\geq H(c) + \mathbb{E}_{x \sim G_{\theta_g}(z,c)} [\mathbb{E}_{c' \sim p(c|x)} [\log q(c'|x)]]
\end{aligned}
$$

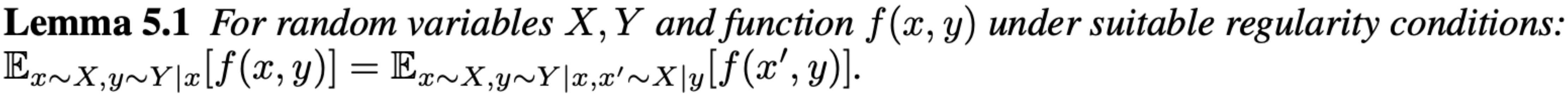
- This technique of lower bounding mutual information is known as Variational Information Maximization

Mutual Information Maximization

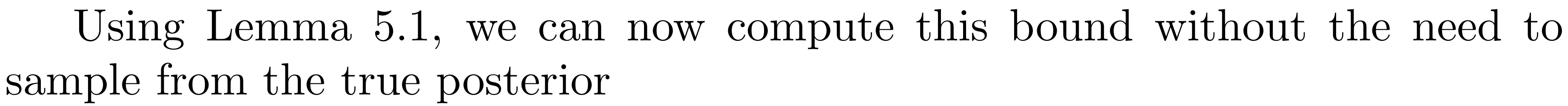
$$I(c; G_{\theta_g}(z, c)) \geq H(c) + \mathbb{E}_{x \sim G_{\theta_g}(z,c)}[\mathbb{E}_{c' \sim p(c|x)}[\log q(c'|x)]]$$

- In implementation we fix the latent code distribution so that $H(c)$ can be treated as a constant
- As can be seen, we no longer need to compute the posterior $p(c|x)$, but we must still sample from it which is bothersome.

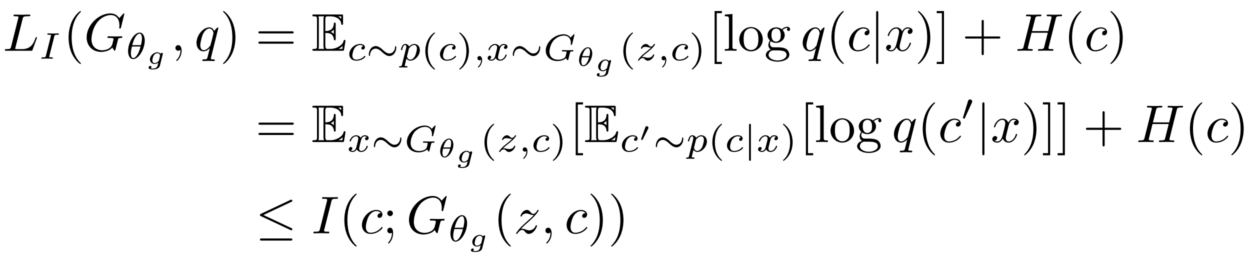
Mutual Information Maximization



Lemma 5.1 *For random variables X, Y and function $f(x, y)$ under suitable regularity conditions:* $\mathbb{E}_{x\sim X, y\sim Y|x}[f(x, y)] = \mathbb{E}_{x\sim X, y\sim Y|x, x'\sim X|y}[f(x', y)]$.



Using Lemma 5.1, we can now compute this bound without the need to sample from the true posterior



$$
\begin{aligned}
L_I(G_{\theta_g}, q) &= \mathbb{E}_{c\sim p(c), x\sim G_{\theta_g}(z,c)}[\log q(c|x)] + H(c) \\
&= \mathbb{E}_{x\sim G_{\theta_g}(z,c)}[\mathbb{E}_{c'\sim p(c|x)}[\log q(c'|x)]] + H(c) \\
&\leq I(c; G_{\theta_g}(z, c))
\end{aligned}
$$

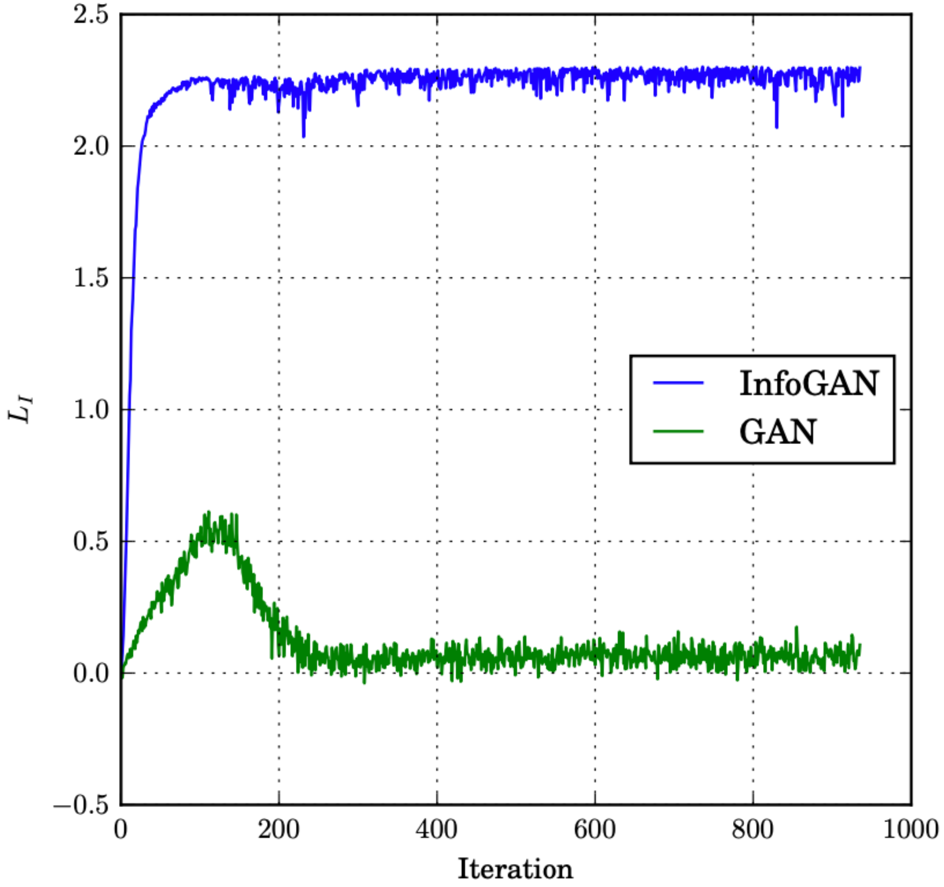
- Thus, we can easily compute the bound by sampling from the prior on the latent codes
- Additionally, we can easily approximate L_I via MC simulation
- Note that L_I can be maximized w.r.t. the auxiliary distribution $q(c|x)$ and w.r.t. generator G_{θ_g}via the reparameterization trick, and can thus be added to GAN’s training procedure with no additional cost
- Lastly, the lower bound becomes tight when $q(c|x) \rightarrow p(c|x)$ and variational lower bound attains its maximum when $L_I\left(G_{\theta_g}, q\right) = H(c)$

Implementation Notes

- Parameterize the auxiliary distribution $q(c|x)$ to be a neural network
- The auxiliary distribution and discriminator D share convolutional layers
- Observed that $L_I(G_{\theta_g}, q)$ always converges faster than normal GANs
- For categorical latent code c_i we use a softmax nonlinearity to represent $q(c_i|x)$
- For continuous latent code c_j we treat $q(c_j|x)$ as a factored Gaussian

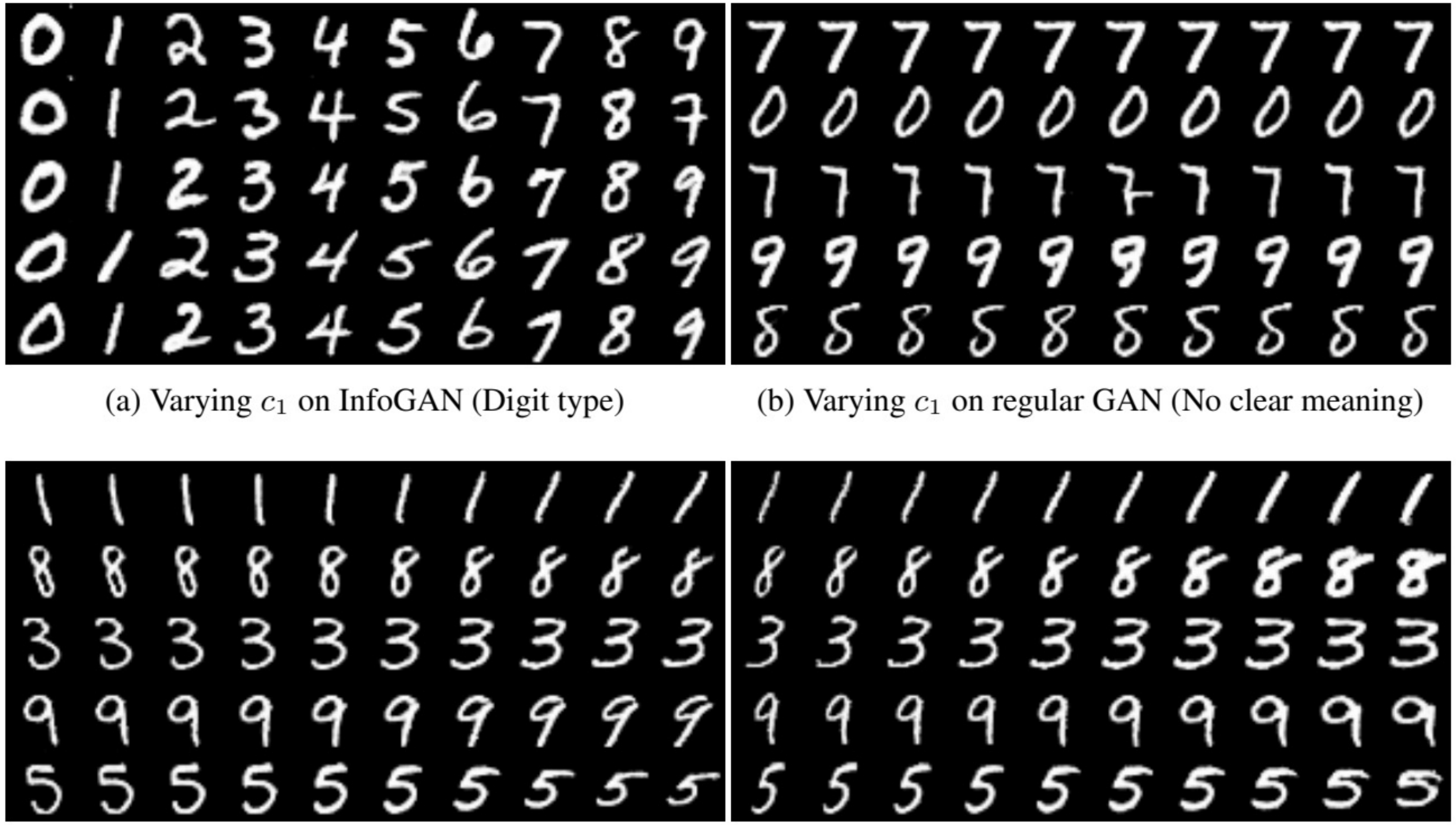
Experiments

- Goal: Investigate if mutual information can be maximized efficiently
  - Can we push lower bound to $H(c)$?
- Train InfoGAN on MNIST with uniform categorical latent codes
- Train regular GAN with an auxiliary distribution q where generator is not trained to maximize mutual information with latent codes
- Conclusion: In GAN, there is no guarantee that generator will use latent codes



Experiments

- Goal: Evaluate if InfoGAN can learn disentangled and interpretable representations
- MNIST
  - Model latent codes with one categorical code to model discontinuous variation in data
  - Two continuous codes to capture variations that are continuous in nature (style)



(a) Varying c_1 on InfoGAN (Digit type)

(b) Varying c_1 on regular GAN (No clear meaning)

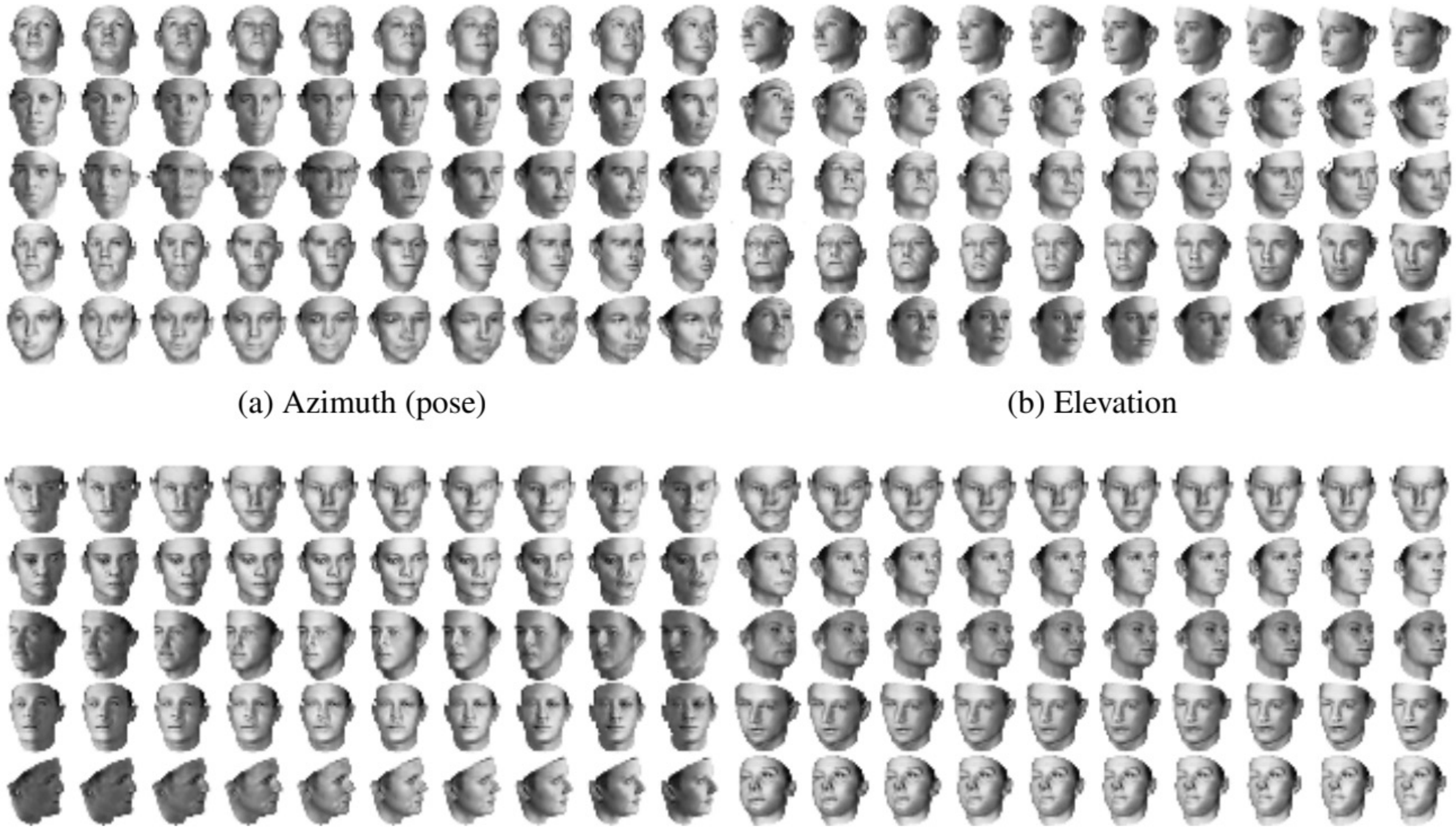
(c) Varying c_2 from -2 to 2 on InfoGAN (Rotation)

(d) Varying c_3 from -2 to 2 on InfoGAN (Width)

Figure 2: **Manipulating latent codes on MNIST:** *In all figures of latent code manipulation, we will use the convention that in each one latent code varies from left to right while the other latent codes and noise are fixed. The different rows correspond to different random samples of fixed latent codes and noise. For instance, in (a), one column contains five samples from the same category in c_1, and a row shows the generated images for* 10 *possible categories in c_1 with other noise fixed.* In (a), each category in c_1 largely corresponds to one digit type; in (b), varying c_1 on a GAN trained without information regularization results in non-interpretable variations; in (c), a small value of c_2 denotes left leaning digit whereas a high value corresponds to right leaning digit; in (d), c_3 smoothly controls the width. We reorder (a) for visualization purpose, as the categorical code is inherently unordered.

Experiments

- Goal: Evaluate if InfoGAN can learn disentangled and interpretable representations
- Faces
  - Continuous latent codes allow InfoGAN to learn a disentangles representation that recovers the azimuth (pose), elevation and lighting
- Chairs
  - InfoGAN can continuously interpolate between similar chair types of different widths using a single continuous code



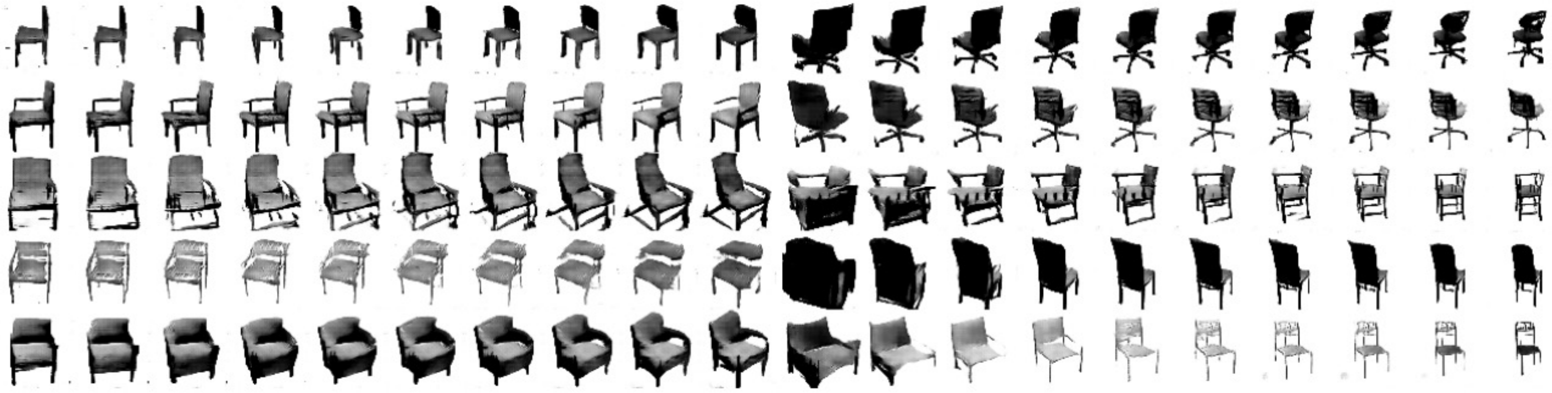
(a) Azimuth (pose)

(b) Elevation

(c) Lighting

(d) Wide or Narrow

Figure 3: **Manipulating latent codes on 3D Faces:** We show the effect of the learned continuous latent factors on the outputs as their values vary from -1 to 1. In (a), we show that one of the continuous latent codes consistently captures the azimuth of the face across different shapes; in (b), the continuous code captures elevation; in (c), the continuous code captures the orientation of lighting; and finally in (d), the continuous code learns to interpolate between wide and narrow faces while preserving other visual features. For each factor, we present the representation that most resembles prior supervised results [7] out of 5 random runs to provide direct comparison.



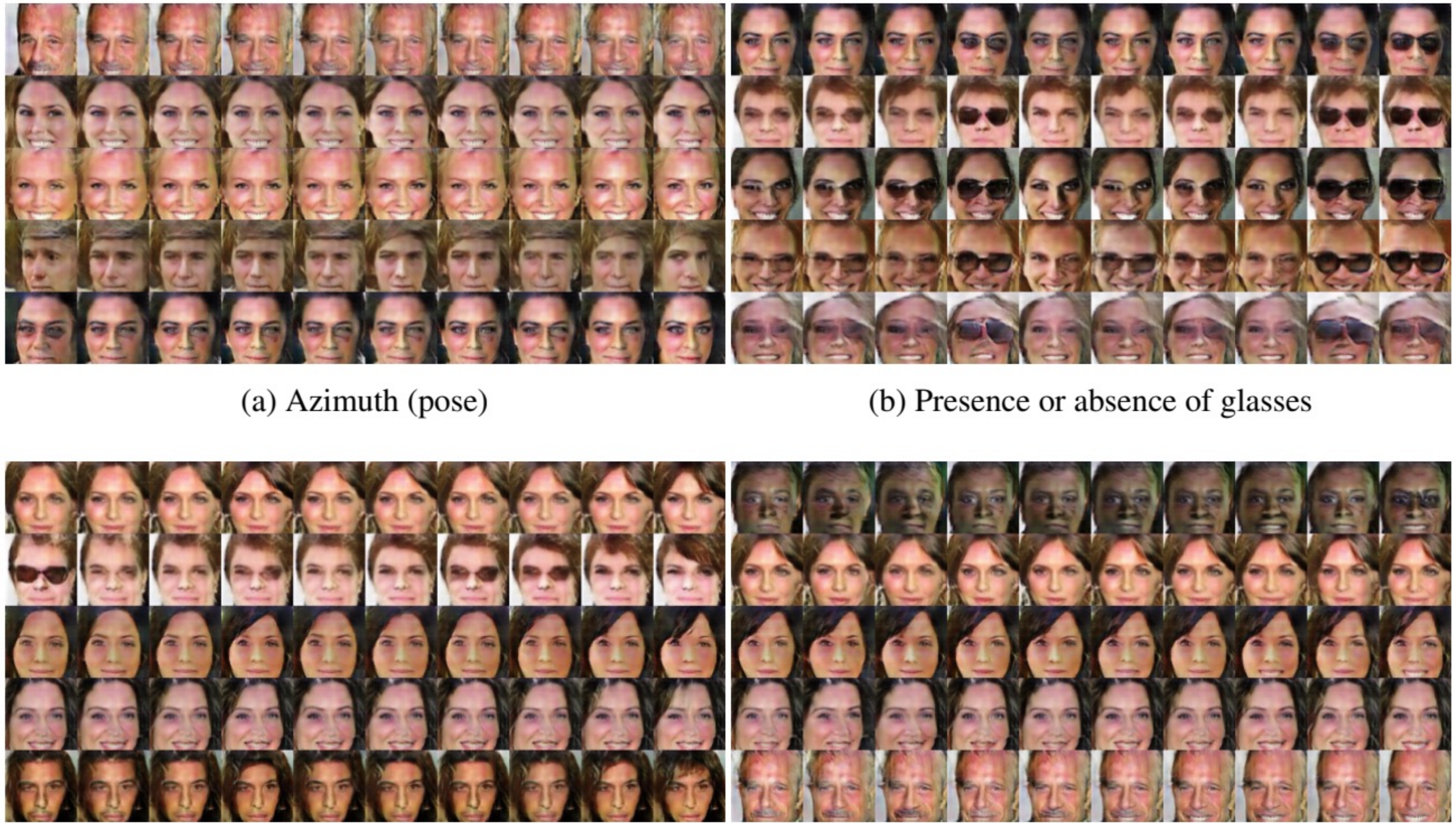
(a) Rotation

(b) Width

Figure 4: **Manipulating latent codes on 3D Chairs:** In (a), we show that the continuous code captures the pose of the chair while preserving its shape, although the learned pose mapping varies across different types; in (b), we show that the continuous code can alternatively learn to capture the widths of different chair types, and smoothly interpolate between them. For each factor, we present the representation that most resembles prior supervised results [7] out of 5 random runs to provide direct comparison.

Experiments

- Goal: Evaluate if InfoGAN can learn disentangled and interpretable representations
- CelebA
  - Model latent variation as 10 uniform categorical variables and still show that InfoGAN can recover azimuth without variance of poses within the dataset
  - Can additionally disentangle other sematic feature such as presence/absence of glasses and varying hairstyles and emotions



(a) Azimuth (pose)

(b) Presence or absence of glasses

(c) Hair style

(d) Emotion

Figure 6: **Manipulating latent codes on CelebA:** (a) shows that a categorical code can capture the azimuth of face by discretizing this variation of continuous nature; in (b) a subset of the categorical code is devoted to signal the presence of glasses; (c) shows variation in hair style, roughly ordered from less hair to more hair; (d) shows change in emotion, roughly ordered from stern to happy.