Generative Adversarial Nets

Jun Gao

Outline

- What is Generative Adversarial Nets?
- Applications and Training Techniques
  - Text to Image Synthesis
  - Improved Techniques
- Model Extension
  - InfoGAN

Generative Adversarial Nets

Ian J. Goodfellow, Jean Pouget-AbadieY Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair Z Aaron Courville, Yoshua Bengio
University de Montreal
In NIPS 2014 (cited by 233)

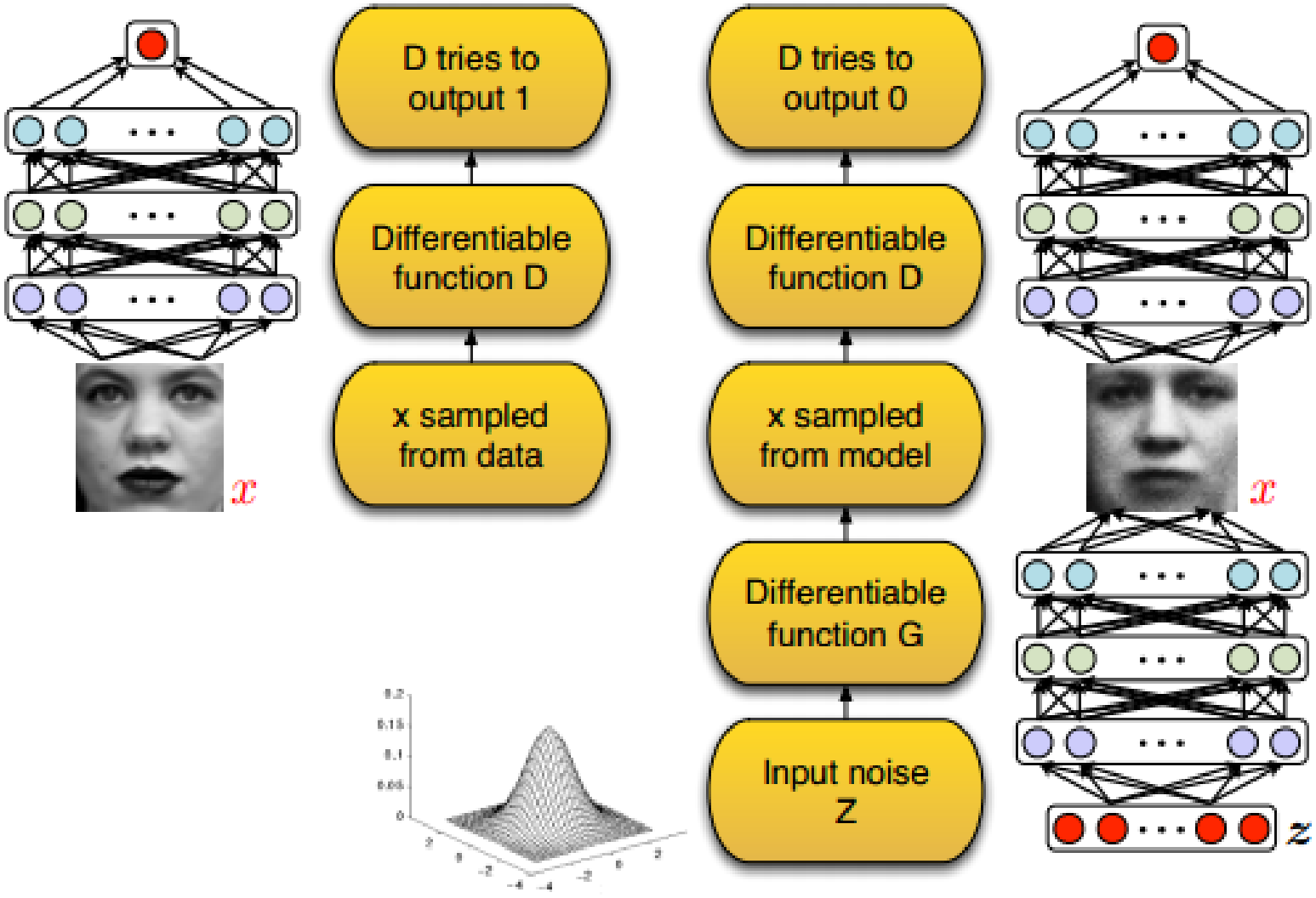
Motivation

- Striking successes of deep discriminative model
- Less impact of deep generative model
- Difficulty of approximating intractable probabilistic computations

$$p(x,y) = p(x)*p(y|x)$$

- Don't write a formula for p(x), just learn to sample directly.
- How? By playing a game.

Adversarial nets framework



NIPS 2014 workshop
---Ian Goodfellow

Loss

- Minmax objective function:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

- In practice, to estimate G we use:

$$\max_G \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log D(G(\boldsymbol{z}))]$$

Why? Stronger gradient for G when D is very good.

Algorithm

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for

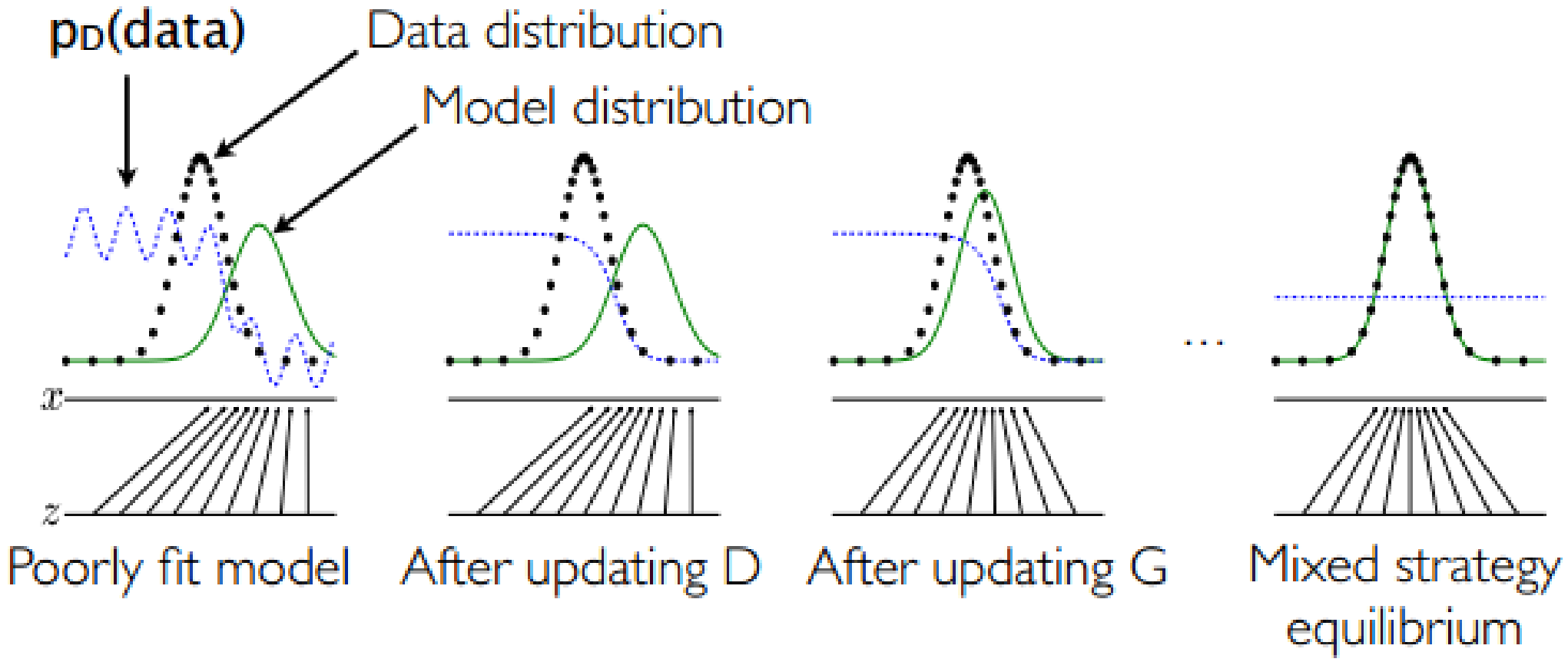
  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Learning process



NIPS 2014 workshop
---Ian Goodfellow

Theoretical results

- Assumption:
  - infinite data,
  - infinite model capacity,
  - direct updating of generator’s distribution.

Theoretical results

- Unique global optimum:
  - For G fixed, the optimal discriminator D is :

$$D_G^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})}$$

  - Reformulate training criterion of G:

$$C(G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{P_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x}\sim p_g}\left[\log \frac{p_g(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right]$$

  - The global minimum of C(G) is achieved if and only if $p_g = p_{data}$.

$$C(G) = -\log(4) + KL\left(p_{\text{data}} \middle\| \frac{p_{\text{data}} + p_g}{2}\right) + KL\left(p_g \middle\| \frac{p_{\text{data}} + p_g}{2}\right)$$

Theoretical results

Assumption:
- infinite data,
- infinite model capacity,
- direct updating of generator' s distribution.

- Convergence to optimal guaranteed:
  - If G and D have enough capacity, and at each step of Algorithm 1, the discriminator is allowed to reach its optimum given G, and is updated p_g so as to improve the criterion

$$\mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[\log(1 - D_G^*(\boldsymbol{x}))]$$

    then p_g converges to p_{data}
  - Proof: consider it as a function of p_g, It is a convex function!!!

- However: In practice we optimize θ_g instead of p_g itself… …

Question!
既然可以通过convex分析，为什么后面又无法收敛了呢？能不能从Optimization的角度来看？

Experiments

Model	MNIST	TFD
DBN [3]	138 ± 2	1909 ± 66
Stacked CAE [3]	121 ± 1.6	$\mathbf{2110 \pm 50}$
Deep GSN [5]	214 ± 1.1	1890 ± 29
Adversarial nets	$\mathbf{225 \pm 2}$	$\mathbf{2057 \pm 26}$

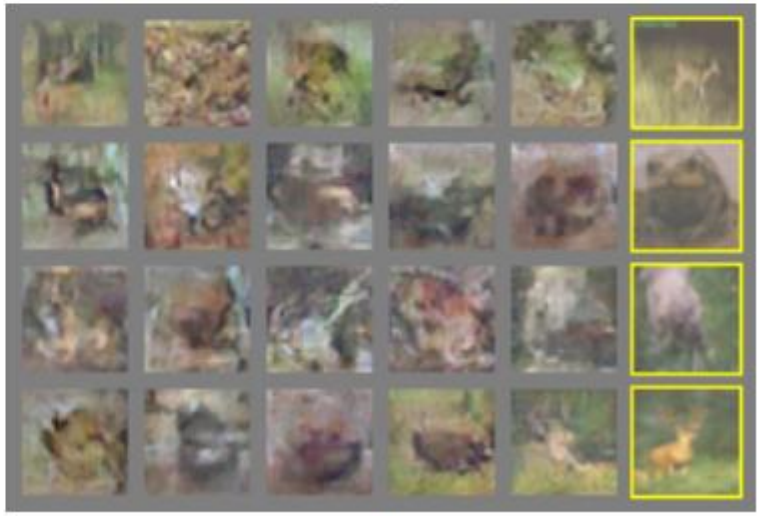
Estimate probability of test data under p_g by fitting a Gaussian Parzen windows to samples generated with G and reporting the log-likelihood under this distribution.

Experiments

a)

b)

c)



d)

Rightmost column shows the nearest training example of the neighboring sample

Pros && Cons

- Pros:
  - A new framework for generative model.
  - No need for Markov Chain, only backprop to obtain gradient. (Computional) !
  - Represent very sharp, even degenerate distribution.

- Cons:
  - No explicit representation of $p_g(x)$
  - The Helvetica scenario(output same value of x regardless of z)

Generative Adversarial Text to Image Synthesis

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, Honglak Lee
University of Michigan, Max Planck Institute for Informatics Germany
In ICML 2016 (cited by 7)

this small bird has a pink breast and crown, and black primaries and secondaries.

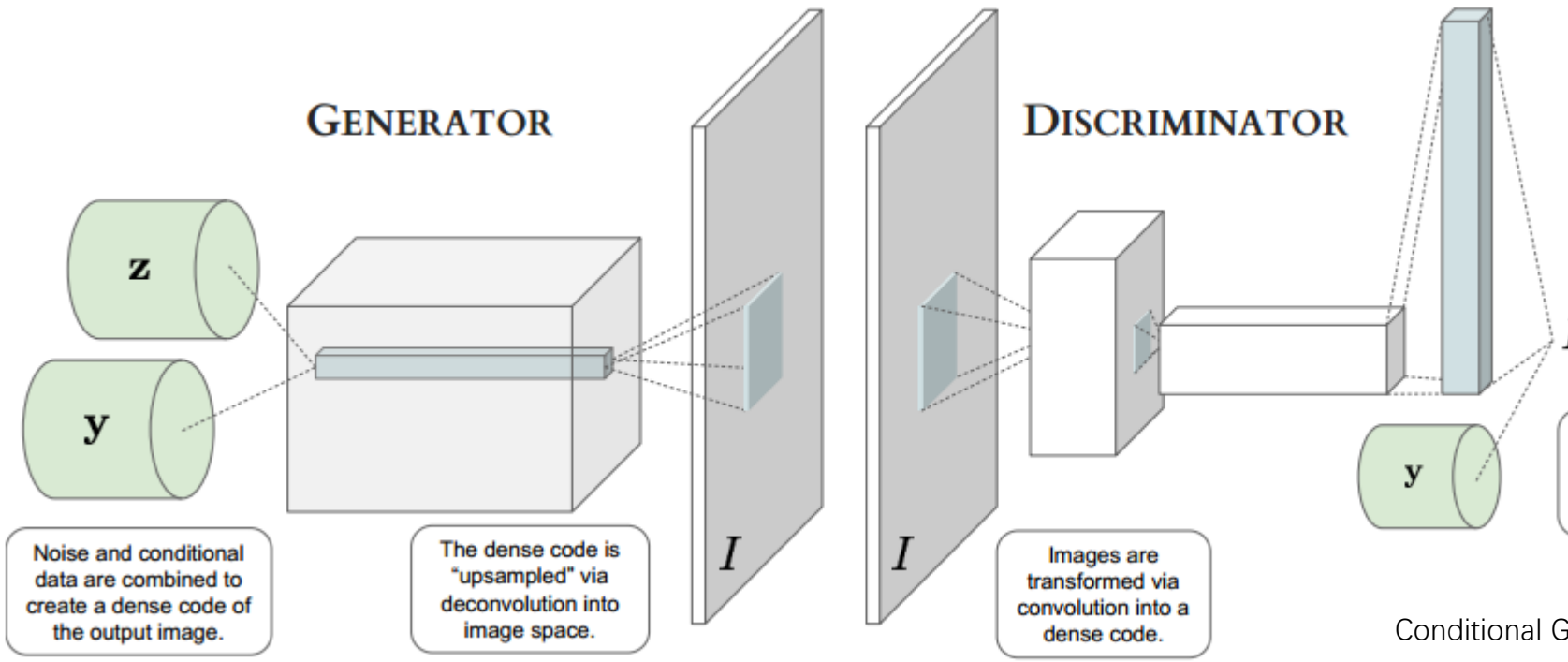
this magnificent fellow is almost all black with a red crest, and white cheek patch.

the flower has petals that are bright pinkish purple with white stigma

this white and yellow flower have thin white petals and a round yellow stamen

Conditional GAN

- Condition on external information (some attributes we want to constrain on output images)



Notation:

z: sampled Gaussian noise vector;

I : output image.

y : a vector of attributes we want to constrain on I (0.1…9 one-hot vector for MNIST, face attributes for face generation)

Conditional Generative Adversarial Nets for Convolutional Face Generation
--Jon Gauthier(Stanford)-Arxiv (cited by 16)

Conditional Generative Adversarial Nets
--Mehdi Mirza(Montereal)-Arxiv(cited by 25)

Results

Model	Test set neg. log-likelihood
Vanilla GAN [8]	3024 ± 22
Conditional GAN	$\mathbf{2934 \pm 22}$

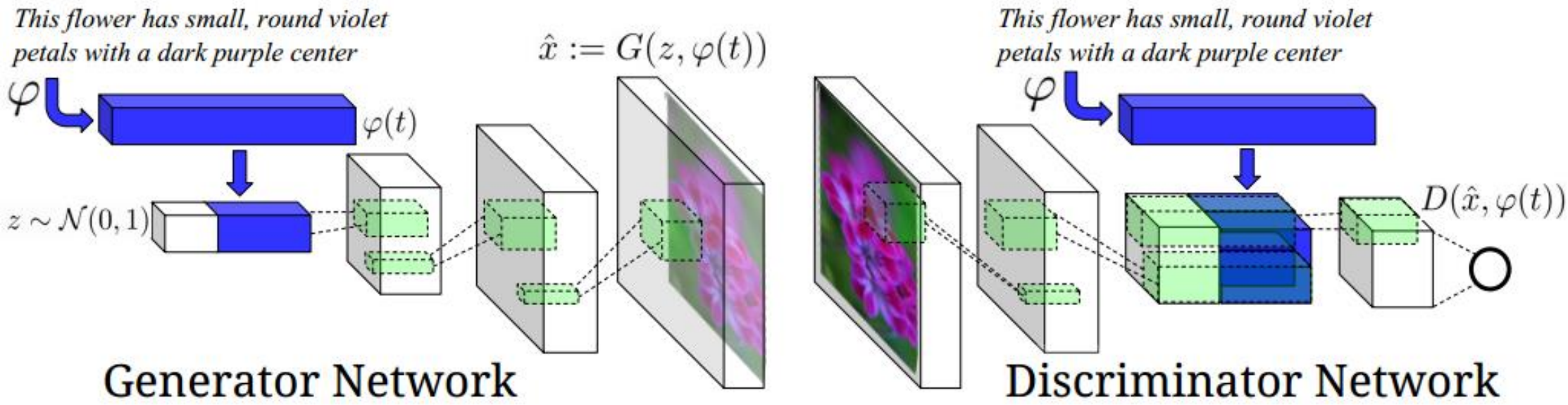
LFW-crop dataset
13,000 color images(32*32*3RGB)
73 different attributes in total,
selected 36 in the model

Analysis

- Conditional info. counts
- Conditional multi-modality
- Can we condition on text and output an image?
  - Use a character-level encoder to encode text
  - Then all the same!

this small bird has a pink breast and crown, and black primaries and secondaries.

Network architecture



Training details

- Matching-aware discriminator(GAN-CLS):
  - Not only real image, but also matching.
  - Discriminator: real image/matching text; fake image/arbitrary text; real image/arbitrary text.
- Learning with manifold interpolation(GAN-INT):
  - Interpolations between embedding pairs tend to be near the data manifold
  - Could generate amount of additional text by interpolations.
  - By satisfying D on interpolated text embeddings, G can learn to fill in gaps on the data manifold in between training points

$$\mathbb{E}_{t_1,t_2\sim p_{data}}[\log(1-D(G(z,\beta t_1+(1-\beta)t_2)))]$$

Algorithm

Algorithm 1 GAN-CLS training algorithm with step size α, using minibatch SGD for simplicity.

Input: minibatch images x, matching text t, mismatching $\hat{t}$, number of training batch steps S
for $n = 1$ **to** S **do**
  $h \leftarrow \varphi(t)$ {Encode matching text description}
  $\hat{h} \leftarrow \varphi(\hat{t})$ {Encode mis-matching text description}
  $z \sim \mathcal{N}(0, 1)^Z$ {Draw sample of random noise}
  $\hat{x} \leftarrow G(z, h)$ {Forward through generator}
  $s_r \leftarrow D(x, h)$ {real image, right text}
  $s_w \leftarrow D(x, \hat{h})$ {real image, wrong text}
  $s_f \leftarrow D(\hat{x}, h)$ {fake image, right text}
  $\mathcal{L}_D \leftarrow \log(s_r) + (\log(1 - s_w) + \log(1 - s_f))/2$
  $D \leftarrow D - \alpha \partial \mathcal{L}_D / \partial D$ {Update discriminator}
  $\mathcal{L}_G \leftarrow \log(s_f)$
  $G \leftarrow G - \alpha \partial \mathcal{L}_G / \partial G$ {Update generator}
end for

Results

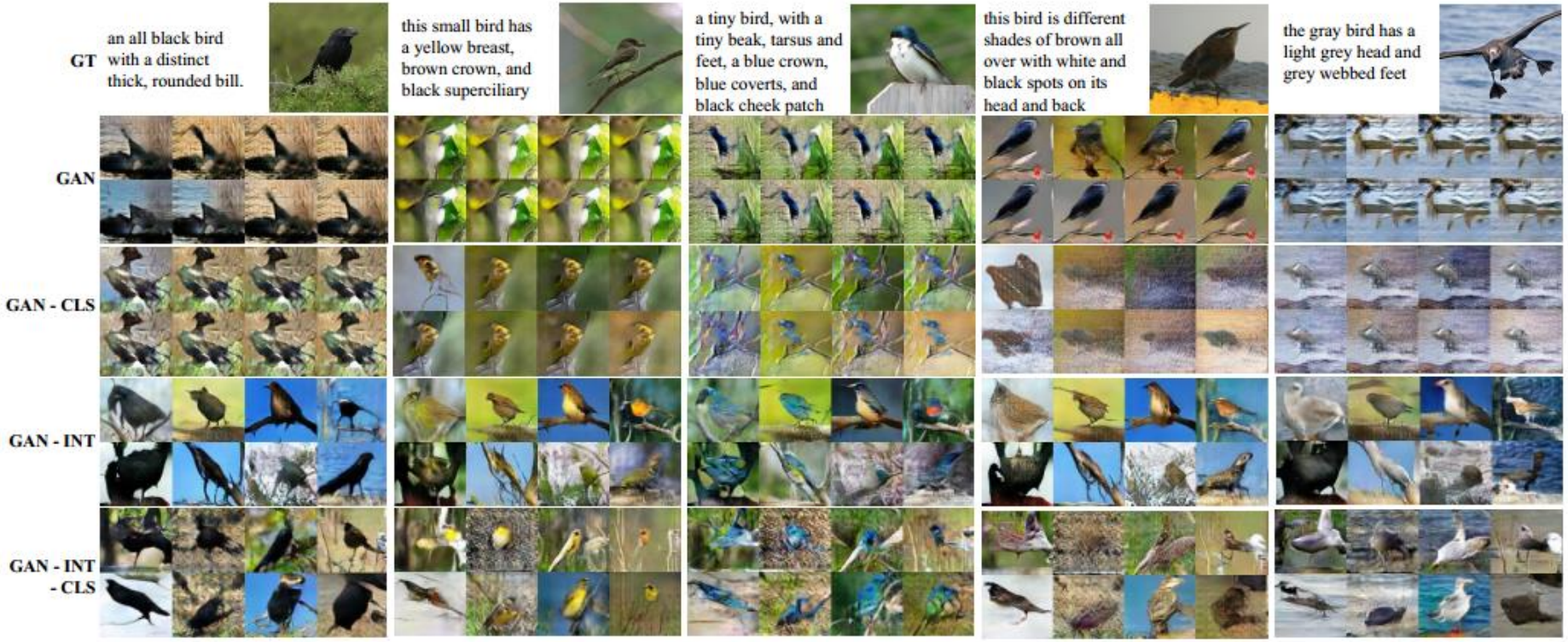


Figure 3. Zero-shot (i.e. conditioned on text from unseen test set categories) generated bird images using GAN, GAN-CLS, GAN-INT and GAN-INT-CLS. We found that interpolation regularizer was needed to reliably achieve visually-plausible results.

Zero-shot!!

CUB dataset:
11788 images of birds,
Five text per image.
200 different categories,
150train+val, 50 test

Results

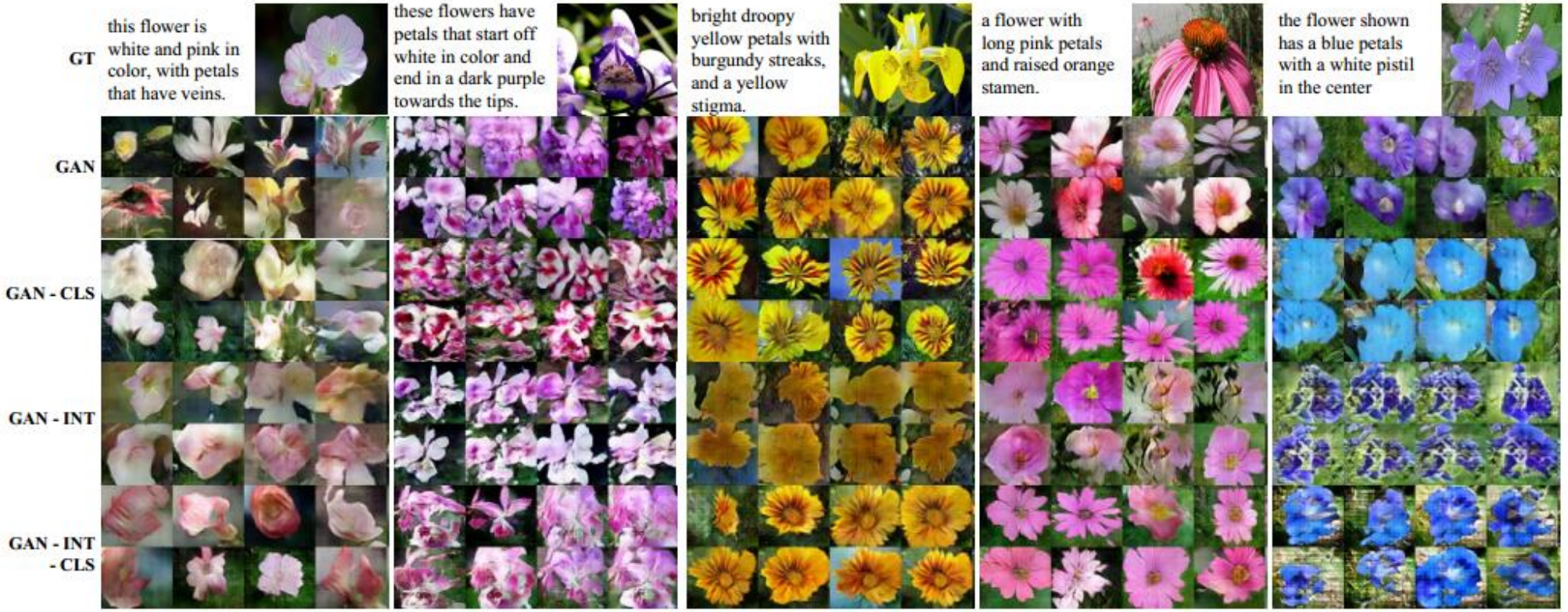


Figure 4. Zero-shot generated flower images using GAN, GAN-CLS, GAN-INT and GAN-INT-CLS. All variants generated plausible images. Although some shapes of test categories were not seen during training (e.g. columns 3 and 4), the color information is preserved.

Zero-shot!!

Oxford-102 dataset:
8189 images of flowers,
Five text per image
102 different categories,
82train+val, 20 test

Results

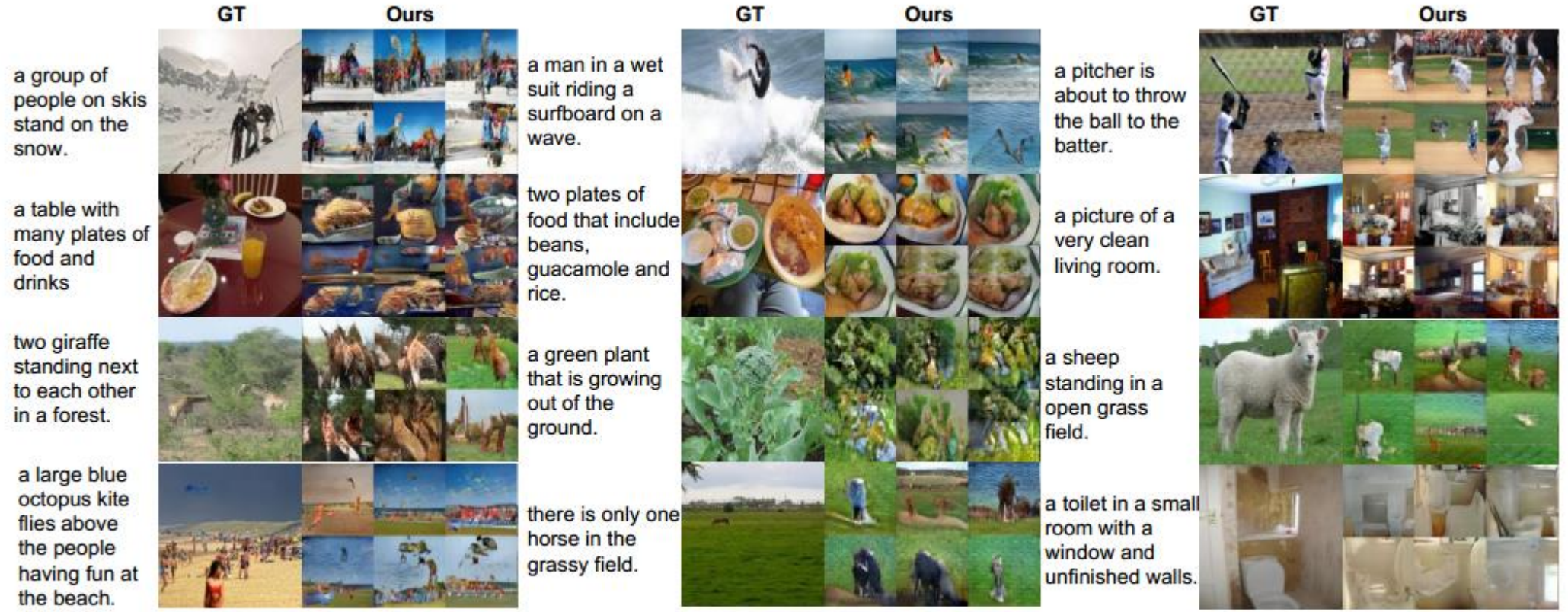


Figure 7. Generating images of general concepts using our GAN-CLS on the MS-COCO validation set. Unlike the case of CUB and Oxford-102, the network must (try to) handle multiple objects and diverse backgrounds.

MS COCO

Use GAN-CLS
All parameters same

Contribution

- Framework for text to image synthesis .
- Manifold interpolation strategy.
- Zero-shot synthesis.
- Generalizability.

Improved Techniques for Training GANs

Tim Salimans，Ian Goodfellow，Wojciech Zaremba，Vicki Cheung，Alec Radford，Xi Chen
OpenAI
In NIPS 2016 (cited by 17)

Motivation

- Unstable training process.
- The lack of proper evaluation metric.
- Semi-supervised learning.

Toward Convergent GAN Training

- Feature matching
  - Not focus on the output of Discriminator, but features of data.

$$||\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\mathbf{f}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\mathbf{f}(G(\boldsymbol{z}))||_2^2$$

  - $f(x)$:output of intermediate layer of discriminator.

Toward Convergent GAN Training

- Minibatch discrimination
  - How to avoid generator output similar images?
  - Force discriminator looks at samples combination.

Let $\mathbf{f}(\boldsymbol{x}_i) \in \mathbb{R}^A$ $T \in \mathbb{R}^{A \times B \times C}$.

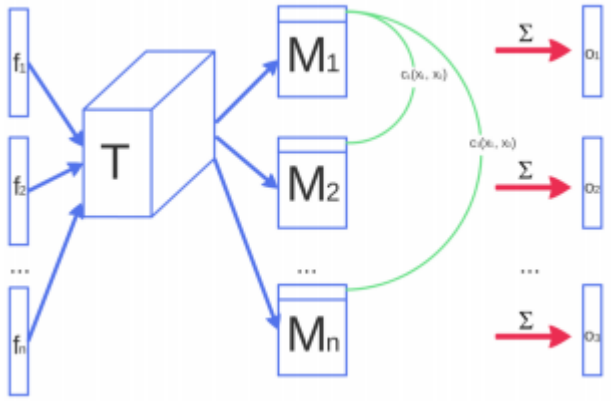
$M_i \in \mathbb{R}^{B \times C}$.

$c_b(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-||M_{i,b} - M_{j,b}||_{L_1}) \in \mathbb{R}$.

$$o(\boldsymbol{x}_i)_b = \sum_{j=1}^{n} c_b(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathbb{R}$$

$$o(\boldsymbol{x}_i) = \left[o(\boldsymbol{x}_i)_1, o(\boldsymbol{x}_i)_2, \ldots, o(\boldsymbol{x}_i)_B\right] \in \mathbb{R}^B$$

$o(\mathbf{X}) \in \mathbb{R}^{n \times B}$

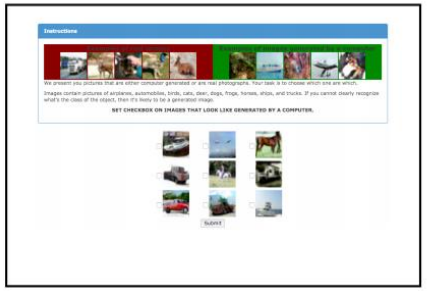


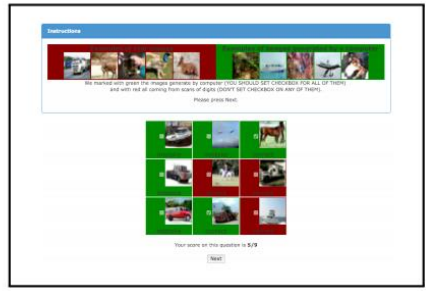
  - Separately compute these minibatch features for samples from the generator and from the training data

Toward Convergent GAN Training

- Historical averaging
  - Add a cost term: $||\boldsymbol{\theta} - \frac{1}{t}\sum_{i=1}^{t}\boldsymbol{\theta}[i]||^2$
- Label smoothing
  - Replaces the 0 and 1 targets for a classifier with smoothed values, like .9 or .1,
- Virtual batch normalization
  - Collect a reference batch of examples that are chosen once and fixed at the start of training.
  - While training, sample $\boldsymbol{x}$ is normalized using only it.

Assessment of image quality





- Human annotators.
- Inception score:
  - Use inception(Google) model pretrained on ImageNet
  - Get the conditional label distribution $p(y|x)$
  - Target:
    - $p(y|x)$ with low entropy
    - $\int p(y|\boldsymbol{x} = G(z))dz$ with high entropy

$$\exp\left(\mathbb{E}_{\boldsymbol{x}}\mathbf{KL}(p(y|\boldsymbol{x})||p(\bar{y}))\right)$$

Semi-supervised learning

- Treat generated image as an additional class
- Rewrite cost function of a classifier:

$$
\begin{aligned}
L &= -\mathbb{E}_{\boldsymbol{x},y\sim p_{\text{data}}(\boldsymbol{x},y)}[\log p_{\text{model}}(y|\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}\sim G}[\log p_{\text{model}}(y=K+1|\boldsymbol{x})] \\
&= L_{\text{supervised}} + L_{\text{unsupervised}}, \text{ where} \\
L_{\text{supervised}} &= -\mathbb{E}_{\boldsymbol{x},y\sim p_{\text{data}}(\boldsymbol{x},y)} \log p_{\text{model}}(y|\boldsymbol{x}, y<K+1) \\
L_{\text{unsupervised}} &= -\{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})} \log[1-p_{\text{model}}(y=K+1|\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim G} \log[p_{\text{model}}(y=K+1|\boldsymbol{x})]\},
\end{aligned}
$$

- We can see:

$$
L_{\text{unsupervised}} = -\{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})} \log D(\boldsymbol{x}) + \mathbb{E}_{z\sim \text{noise}} \log(1-D(G(\boldsymbol{z})))\}
$$

Results on MNIST

Figure 3: *(Left)* samples generated by model during semi-supervised training. Samples can be clearly distinguished from images coming from MNIST dataset. *(Right)* Samples generated with minibatch discrimination. Samples are completely indistinguishable from dataset images.

Model	Number of incorrectly predicted test examples for a given number of labeled samples			
	20	50	100	200
DGN [21]			333 ± 14	
Virtual Adversarial [22]			212	
CatGAN [14]			191 ± 10	
Skip Deep Generative Model [23]			132 ± 7	
Ladder network [24]			106 ± 37	
Auxiliary Deep Generative Model [23]			96 ± 2	
Our model	1677 ± 452	221 ± 136	93 ± 6.5	90 ± 4.2
Ensemble of 10 of our models	1134 ± 445	142 ± 96	86 ± 5.6	81 ± 4.3

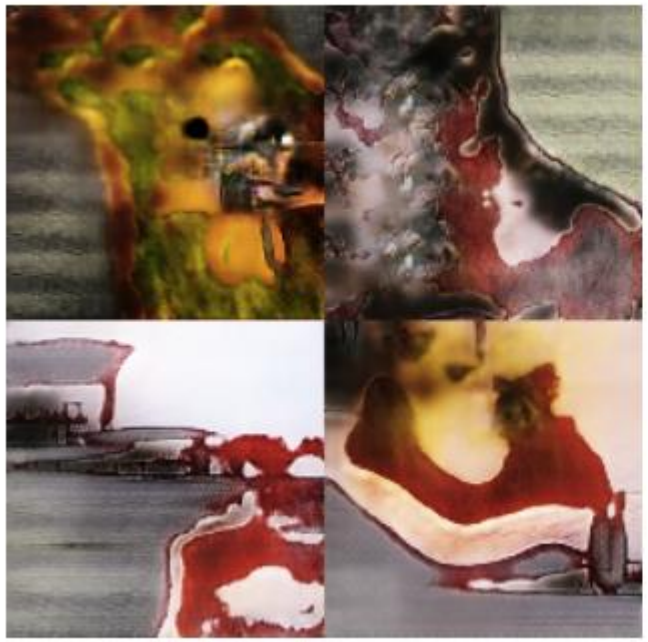
Minibatch discrimination allows generate visually appealing samples, while on semi-supervised learning feature matching works better.

Results on CIFAR-10

Figure 4: Samples generated during semi-supervised training on CIFAR-10 with feature matching (Section 3.1, *left*) and minibatch discrimination (Section 3.2, *right*).

Minibatch discrimination allows generate visually appealing samples, while on semi-supervised learning feature matching works better.

Results on ImageNet



Left: DCGAN; Right: Our model.

InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, Pieter Abbeel
UC Berkeley, OpenAI
In NIPS 2016 (cited by 8)

Motivation

- Representation learning in unsupervised learning framework.
- Generative model could “create” data directly.
- How to encourage it to learn interpretable and meaningful representations?
  - Maximize mutual information!

Method

- Decompose the input noise vector into two parts:
  - noise input$z$
  - latent code $\mathbf{c}$:target salient semantic features of the data distribution.
- Condition on c to generate images: $G(z, c)$
- Add mutual information as a regularization. $I(c; G(z, c))$
- Minmax game:

$$\min_G \max_D V_I(D, G) = V(D, G) - \lambda I(c; G(z, c))$$

Approximate to mutual information

- Hard to compute $P(c|x)$ directly ➔ define an auxiliary distribution $Q(c|x)$ to approximate $P(c|x)$.

$$
\begin{aligned}
I(c; G(z,c)) &= H(c) - H(c|G(z,c)) \\
&= \mathbb{E}_{x\sim G(z,c)}[\mathbb{E}_{c'\sim P(c|x)}[\log P(c'|x)]] + H(c) \\
&= \mathbb{E}_{x\sim G(z,c)}[\underbrace{D_{\mathrm{KL}}(P(\cdot|x) \parallel Q(\cdot|x))}_{\geq 0} + \mathbb{E}_{c'\sim P(c|x)}[\log Q(c'|x)]] + H(c) \\
&\geq \mathbb{E}_{x\sim G(z,c)}[\mathbb{E}_{c'\sim P(c|x)}[\log Q(c'|x)]] + H(c)
\end{aligned}
$$

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

$$
\begin{aligned}
L_I(G,Q) &= E_{c\sim P(c), x\sim G(z,c)}[\log Q(c|x)] + H(c) \\
&= E_{x\sim G(z,c)}[\mathbb{E}_{c'\sim P(c|x)}[\log Q(c'|x)]] + H(c) \\
&\leq I(c; G(z,c))
\end{aligned}
$$

InfoGAN

$$\min_{G,Q} \max_{D} V_{\mathrm{InfoGAN}}(D, G, Q) = V(D, G) - \lambda L_I(G, Q)$$

· Parametrize the auxiliary distribution Q as a neural network.
· Q and D share all convolutional layers
· there is one final fully connected layer to output parameters for the conditional distribution Q(c|x),

Experiments

- Mutual information maximization

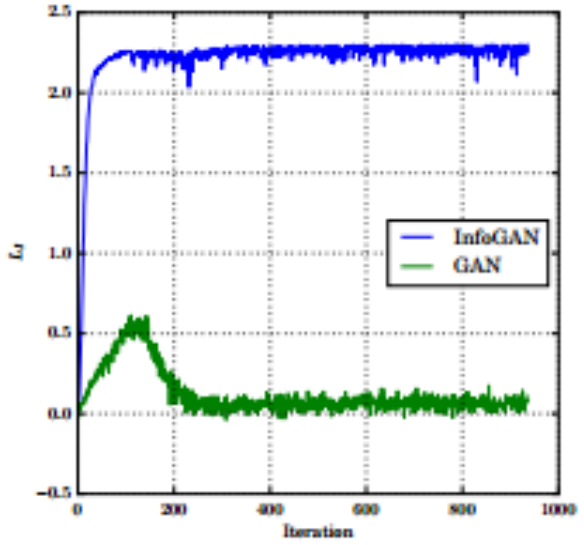


Figure 1: Lower bound L_I over training iterations

Experiments

- Achieves 5% error rate in classifying MNIST

(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)

Experiments

- Meaningful latent code(continuous).

(a) Rotation (b) Width

(a) Azimuth (pose) (b) Elevation

Contribution

- Unsupervised representation learning algorithm.
- Negligible computation cost compared to GAN.
- Mutual information regularization.

Some other GAN models

- Adversarial Autoencoder.
- Laplacian Pyramid of Adversarial Networks. (NIPS15)
- Energy-based GAN. (LeCun)

Conclusion

- A new framework for generative model
- Conditional GAN
- Hard to train
- InfoGAN

Two problems of GAN

- Training instability and sensitivity to hyper-parameter
- Mode missing, yield low entropy distribution.

VAE-GAN

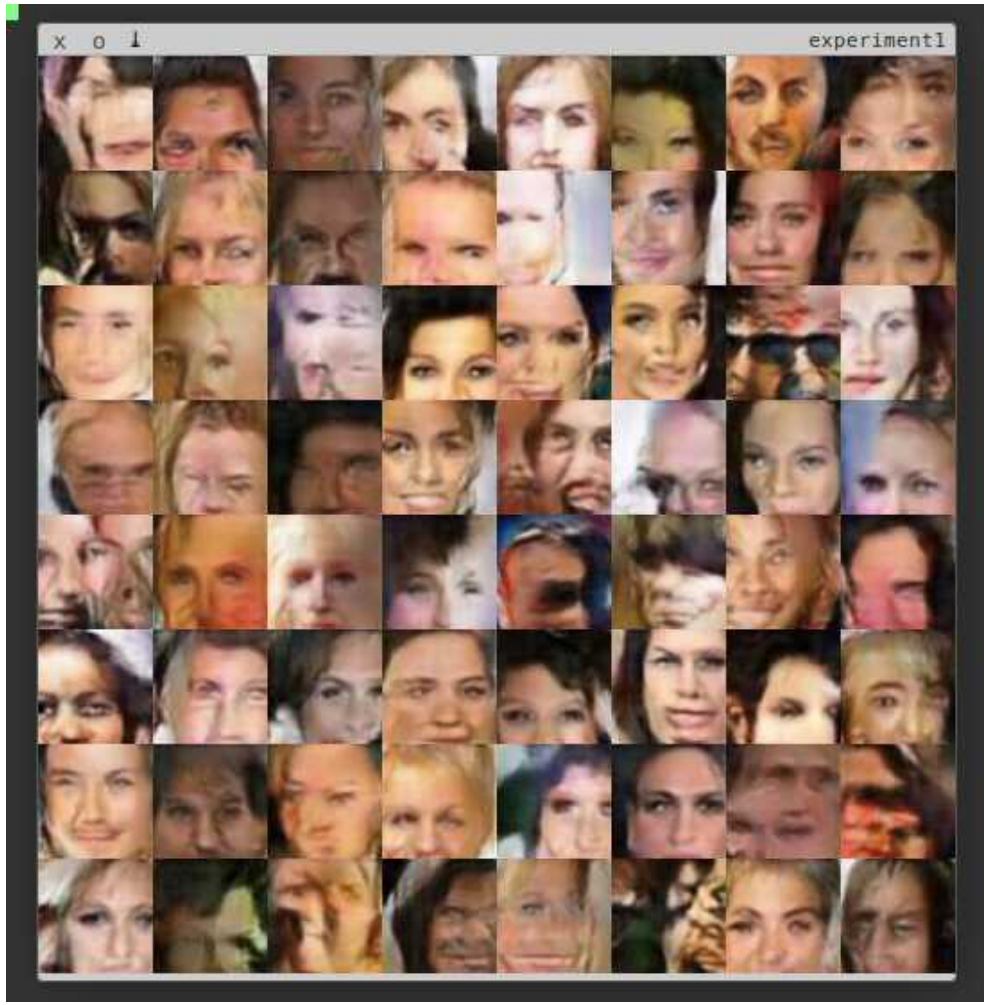
- Deterministic optimization target V.S. a learned discriminator.
- Incorporate supervised training target.

$$\mathbb{E}_{x \sim p_d}[d(x, G \circ E(x))]$$

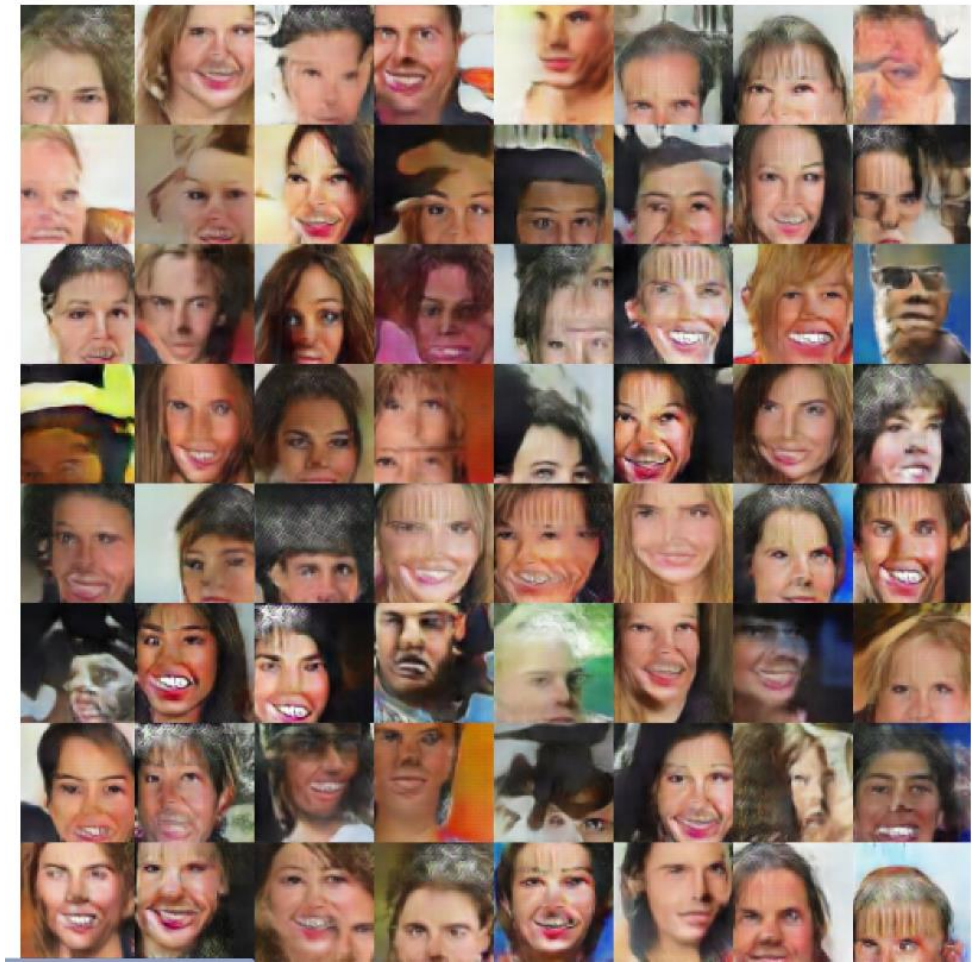
- Space distribution is not a Gaussian.
- Add another regularizer: $KL(E(x)||z)$
- Avoid missing mode problem at the same time.

ICLR-17-Bengio-MODE REGULARIZED GENERATIVE ADVERSARIAL NETWORKS

Results



My



Baseline

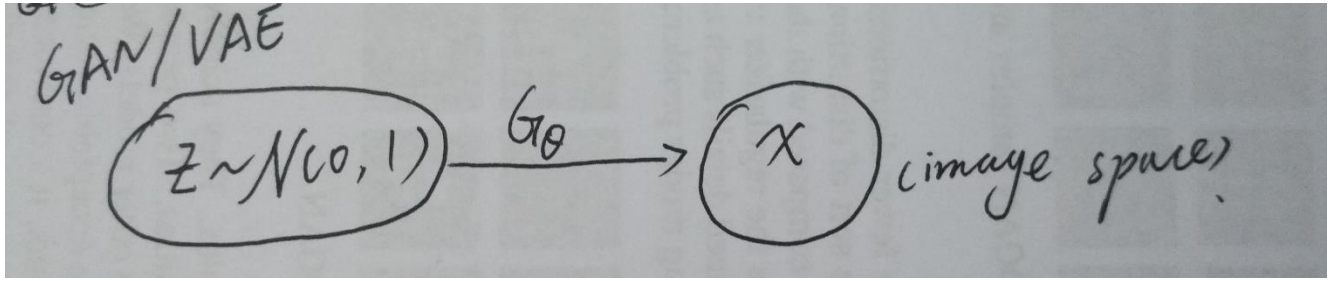
However

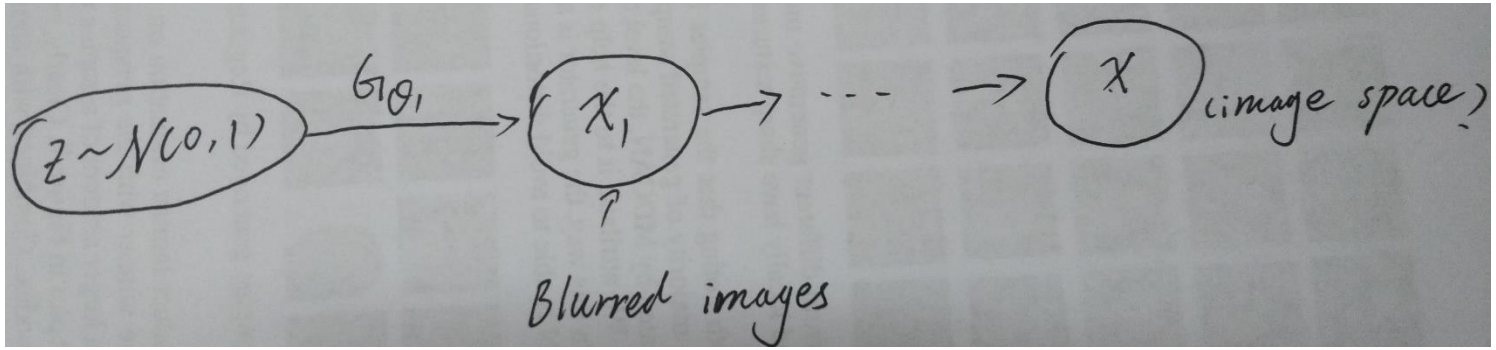
- ICML-16-Autoencoding beyond pixels using a learned similarity metric
- NIPS-16-Generating Images with Perceptual Similarity Metrics based on Deep Networks

- Idea:
  - combine 4 losses: GAN, KL, Reconstruction, feature matching

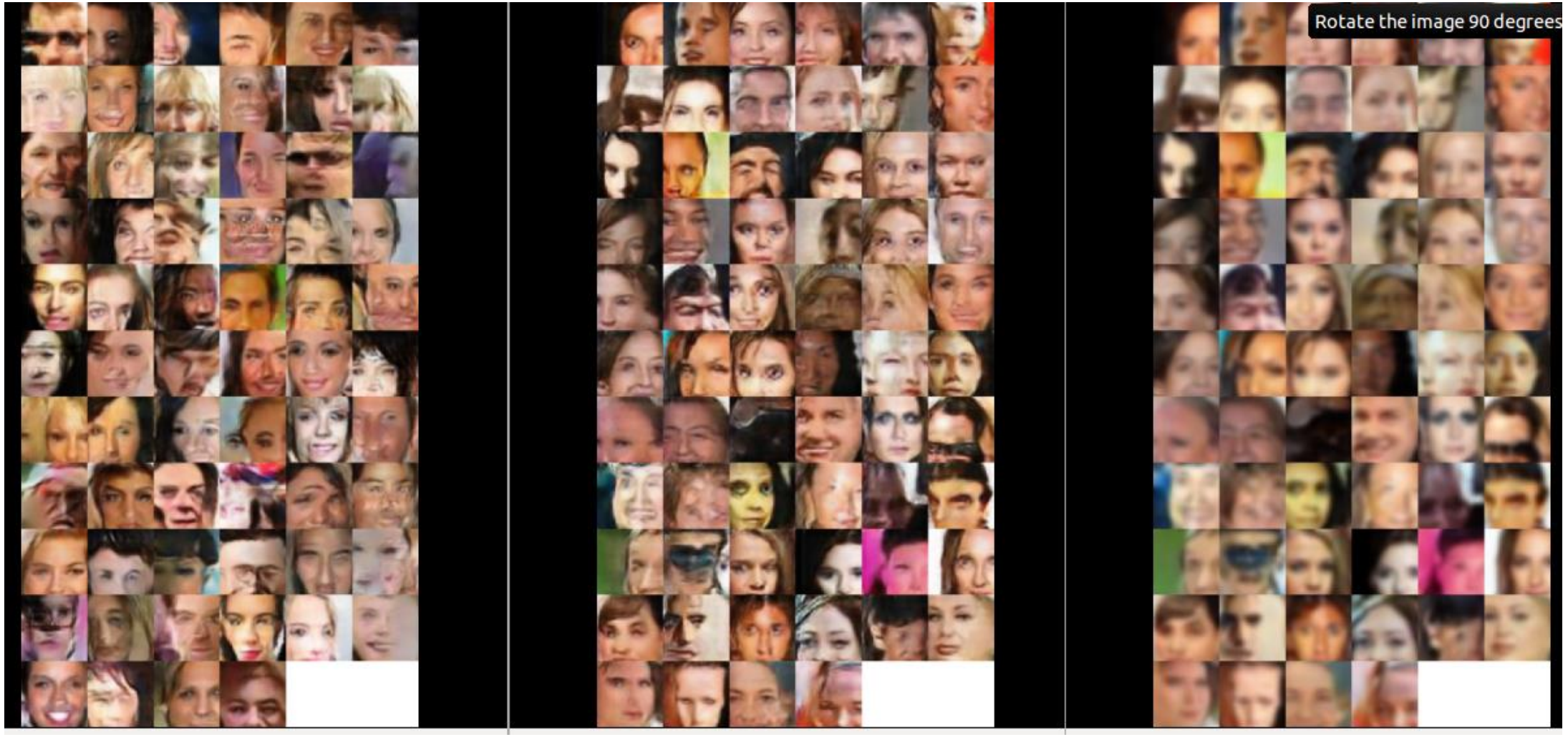
Image Generation Pipeline

- Blurred images first, then clearer images.
- Abstract model:





Results



Single-stage

Two-stage(clear)

Two-stage(blur)

Discussion

- Any other ways to improve training stability?
- Capacity of network? Why not “deep”?
- Problems of generate pipeline?

Thank You