Structured Generative Adversarial Networks

Zhijie Deng
Tsinghua University

Outline

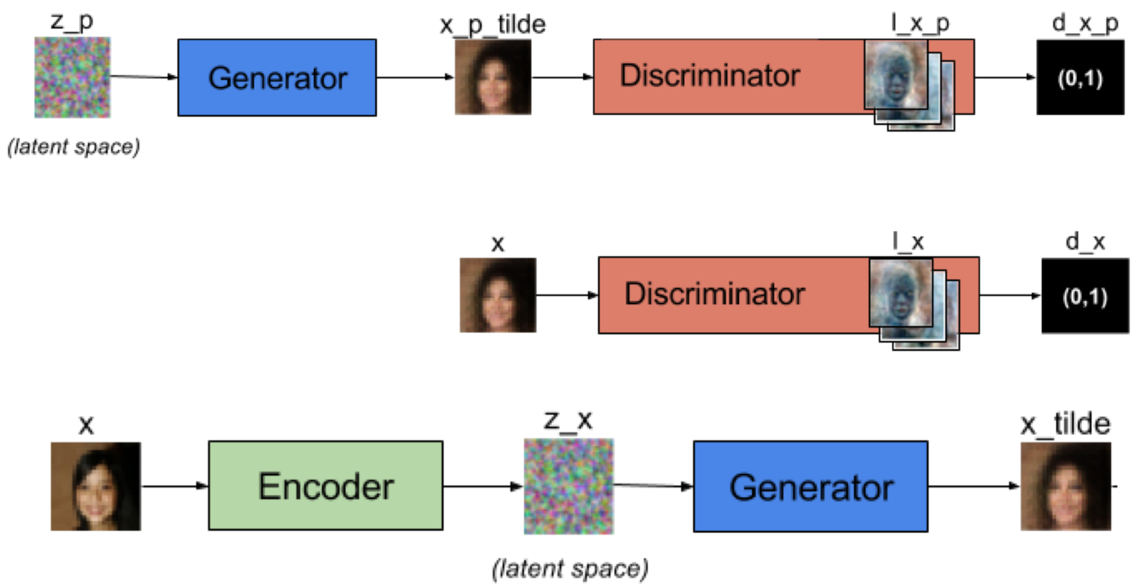
- Background
- Motivation and modelling
- Related work
- Experiments
- Conclusion

Generative Modeling

- Unsupervised learning
- Capture real distribution of data
- Learn hidden representations
- Generate new samples

Deep Generative Models

- Generative adversarial networks
- Variational auto-encoder, Auto-regressive networks, …

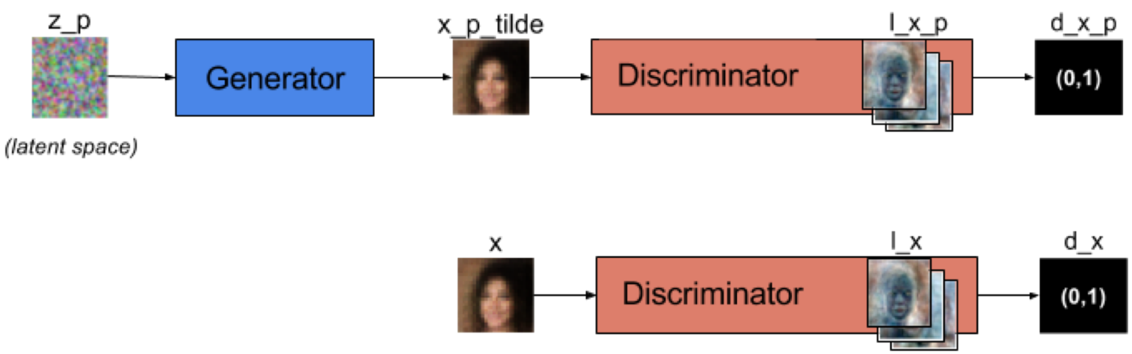


(Goodfellow et al., 2014; Kingma et al., 2013)

Generative Adversarial Networks

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

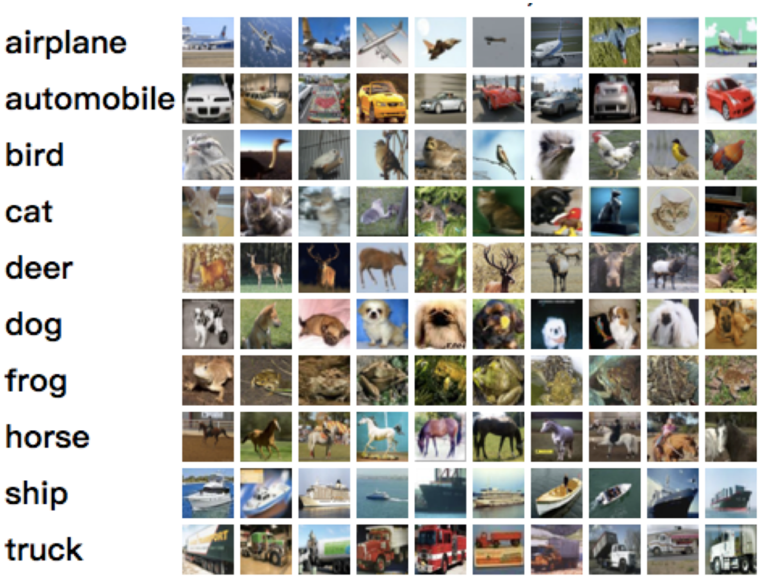
- Global optimum at: $p_G = p_{data}$
- z encodes all the information, e.g. object, background, style…
- Generator is not controllable



(Goodfellow et al., 2014)

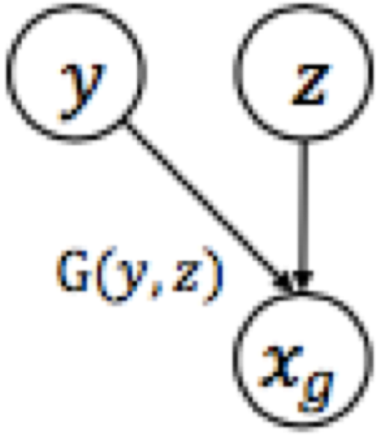
Conditional Generative Modeling

- condition y + random noise z → sample x
- Semi-supervised setting
- How to guarantee the controllability of generator?
  - Disentangle the semantics of our interest and other variations



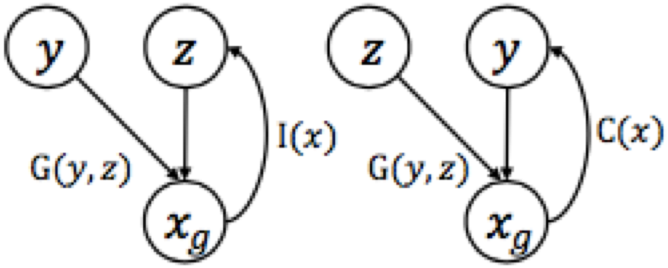
Disentangled hidden representations

- Split z of original GAN into two parts:
  - y: represents generative conditions
  - z: represents other variations
- Consider two parts separately:
  - Dedicated inference models C and I



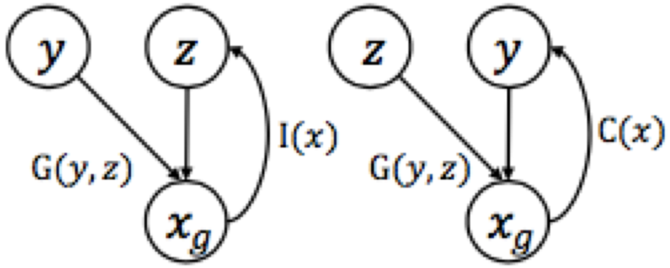
Models for understanding y and z

- y
  1. Learn from a small set of labeled data by MLE
  2. Reconstruct generation conditions from generated samples
- z
  1. Reconstruct random noises from generated samples



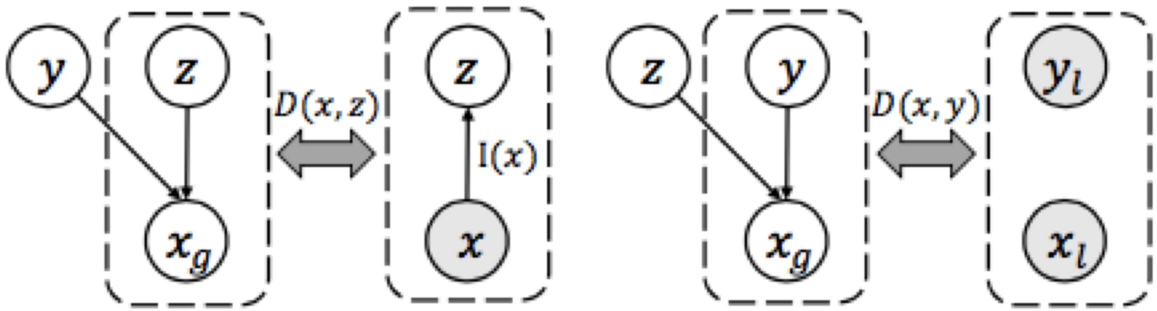
Collaborative games

- $\min_{C,G} \mathcal{R}_{\mathbf{y}} = -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim p(\boldsymbol{x},\boldsymbol{y})}[\log p_c(\boldsymbol{y}|\boldsymbol{x})] - \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim p_g(\boldsymbol{x},\boldsymbol{y})}[\log p_c(\boldsymbol{y}|\boldsymbol{x})]$
  - Implies: $\min_{C,G} \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y})} \left\| p_c(\mathbf{y}|G(\mathbf{y},\mathbf{z}_1)), p_c(\mathbf{y}|G(\mathbf{y},\mathbf{z}_2)) \right\|, \ \forall \mathbf{z}_1, \mathbf{z}_2 \sim p(\mathbf{z})$
- $\min_{I,G} \mathcal{R}_{\mathbf{z}} = -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{z})\sim p_g(\boldsymbol{x},\boldsymbol{z})}[\log p_i(\boldsymbol{z}|\boldsymbol{x})]$
  - Implies: $\min_{I,G} \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})} \left\| p_i(\mathbf{z}|G(\mathbf{y}_1,\mathbf{z})), p_i(\mathbf{z}|G(\mathbf{y}_2,\mathbf{z})) \right\|, \ \forall \mathbf{y}_1, \mathbf{y}_2 \sim p(\mathbf{y})$
- Drive the information encoded by y and z to be disentangled
- Guarantee the controllability of generator



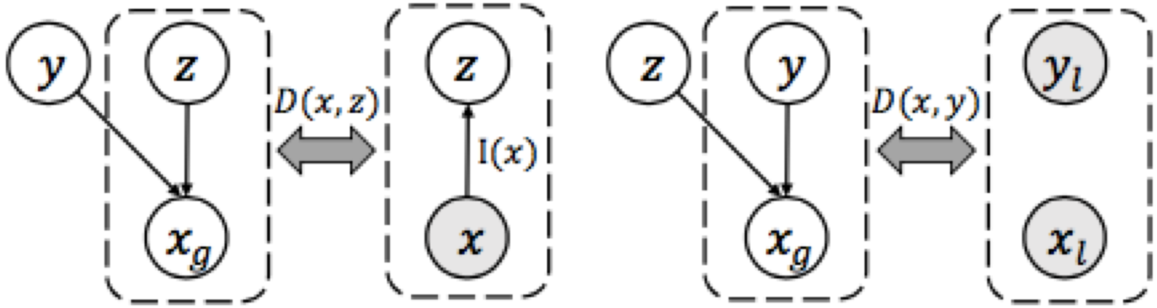
Models for estimating real distribution

- p(x, y, z): complicated; z is not visible
- Model p(x, y) and p(x, z) respectively
  - Draw inspiration from ALI and TripleGAN
  - Adversarial training avoids distributing the probability mass diffusely over data space



Adversarial games

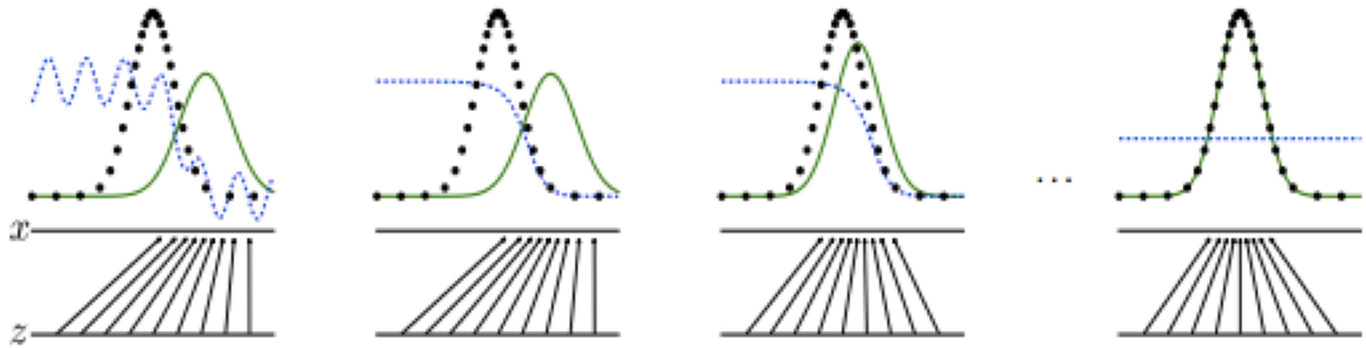
- $\min_{I,G} \max_{D_{xz}} \mathcal{L}_{xz} = \mathbb{E}_{x\sim p(x)}[\log(D_{xz}(x, I(x)))] + \mathbb{E}_{z\sim p(z), y\sim p(y)}[\log(1 - D_{xz}(G(y,z), z))]$
- $\min_{G} \max_{D_{xy}} \mathcal{L}_{xy} = \mathbb{E}_{(x,y)\sim p(x,y)}[\log(D_{xy}(x, y))] + \mathbb{E}_{y\sim p(y), z\sim p(z)}[\log(1 - D_{xy}(G(y,z), y))]$
- $p_g(x,z) = p_i(x,z)$ and $p_g(x,y) = p(x,y)$



Theoretical Guarantees

Theorem 3.1 *The global minimum of* $\max_{D_{\boldsymbol{xz}}} \mathcal{L}_{\boldsymbol{xz}}$ *is achieved if and only if* $p(\boldsymbol{x})p_i(\boldsymbol{z}|\boldsymbol{x}) = p(\boldsymbol{z})p_g(\boldsymbol{x}|\boldsymbol{z})$. *At that point* $D^*_{\boldsymbol{xz}} = \frac{1}{2}$. *Similarly, the global minimum of* $\max_{D_{\boldsymbol{xy}}} \mathcal{L}_{\boldsymbol{xy}}$ *is achieved if and only if* $p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{y})p_g(\boldsymbol{z}|\boldsymbol{y})$. *At that point* $D^*_{\boldsymbol{xy}} = \frac{1}{2}$.

Theorem 3.2 *There exists a generator* $G^*(\boldsymbol{y}, \boldsymbol{z})$ *of which the conditional distributions* $p_g(\boldsymbol{x}|\boldsymbol{y})$ *and* $p_g(\boldsymbol{x}|\boldsymbol{z})$ *can both achieve equilibrium in their own minimax games* $\mathcal{L}_{\boldsymbol{xy}}$ *and* $\mathcal{L}_{\boldsymbol{xz}}$.



(Goodfellow et al., 2014)

Theoretical Guarantees

Theorem 3.3 *Minimizing $\mathcal{R}_{\boldsymbol{z}}$ w.r.t. I will keep the equilibrium of the adversarial game $\mathcal{L}_{\boldsymbol{xz}}$. Similarly, minimizing $\mathcal{R}_{\boldsymbol{y}}$ w.r.t. C will keep the equilibrium of the adversarial game $\mathcal{L}_{\boldsymbol{xy}}$ unchanged.*

Proof: $\mathcal{R}_{\boldsymbol{z}} = KL(p_g(\boldsymbol{x}, \boldsymbol{z})||p_i(\boldsymbol{x}, \boldsymbol{z})) - \mathbb{E}_{(\boldsymbol{x},\boldsymbol{z})\sim p_g(\boldsymbol{x},\boldsymbol{z})}[\log \frac{p_g(\boldsymbol{z},\boldsymbol{x})}{p(\boldsymbol{x})}]$

Discussion

- $\min_{C,G} \mathcal{R}_y = -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim p(\boldsymbol{x},\boldsymbol{y})}[\log p_c(\boldsymbol{y}|\boldsymbol{x})] - \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim p_g(\boldsymbol{x},\boldsymbol{y})}[\log p_c(\boldsymbol{y}|\boldsymbol{x})]$
  - Ry helps $p_G(x, y)$ to approach $p(x, y)$
  - Ry is suitable for semi-supervised learning
- Use pairs generated by C in L_{xy}
  - Labeled data is lacking and may be biased
- SGAN is a combination of MLE-based methods and GAN-based methods

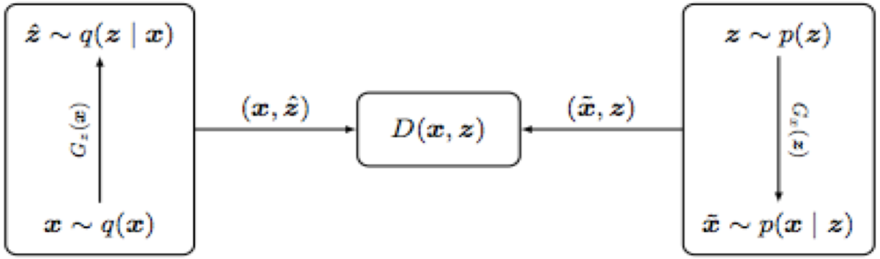
Algorithm

Algorithm 1 Training Structured Generative Adversarial Networks (SGAN).

Pretrain C by minimizing the first term of Eq. 4 w.r.t. C using $\boldsymbol{X}_l$.
repeat
  Sample a batch of $\boldsymbol{x}$: $\boldsymbol{x}_u \sim p(\boldsymbol{x})$.
  Sample batches of pairs $(\boldsymbol{x}, \boldsymbol{y})$: $(\boldsymbol{x}_l, \boldsymbol{y}_l) \sim p(\boldsymbol{x}, \boldsymbol{y})$, $(\boldsymbol{x}_g, \boldsymbol{y}_g) \sim p_g(\boldsymbol{x}, \boldsymbol{y})$, $(\boldsymbol{x}_c, \boldsymbol{y}_c) \sim p_c(\boldsymbol{x}, \boldsymbol{y})$.
  Obtain a batch $(\boldsymbol{x}_m, \boldsymbol{y}_m)$ by mixing data from $(\boldsymbol{x}_l, \boldsymbol{y}_l)$, $(\boldsymbol{x}_g, \boldsymbol{y}_g)$, $(\boldsymbol{x}_c, \boldsymbol{y}_c)$ with proper mixing portion.
  for $k = 1 \to K$ **do**
    Train $D_{\boldsymbol{xz}}$ by maximizing the first term of $\mathcal{L}_{\boldsymbol{xz}}$ using $\boldsymbol{x}_u$ and the second using $\boldsymbol{x}_g$.
    Train $D_{\boldsymbol{xy}}$ by maximizing the first term of $\mathcal{L}_{\boldsymbol{xy}}$ using $(\boldsymbol{x}_m, \boldsymbol{y}_m)$ and the second using $(\boldsymbol{x}_g, \boldsymbol{y}_g)$.
  end for
  Train I by minimizing $\mathcal{L}_{\boldsymbol{xz}}$ using $\boldsymbol{x}_u$ and $\mathcal{R}_{\boldsymbol{z}}$ using $\boldsymbol{x}_g$.
  Train C by minimizing $\mathcal{R}_{\boldsymbol{y}}$ using $(\boldsymbol{x}_m, \boldsymbol{y}_m)$ (see text).
  Train G by minimizing $\mathcal{L}_{\boldsymbol{xz}} + \mathcal{L}_{\boldsymbol{xz}} + \mathcal{R}_{\boldsymbol{y}} + \mathcal{R}_{\boldsymbol{z}}$ using $(\boldsymbol{x}_g, \boldsymbol{y}_g)$.
until convergence.

Related work: ALI

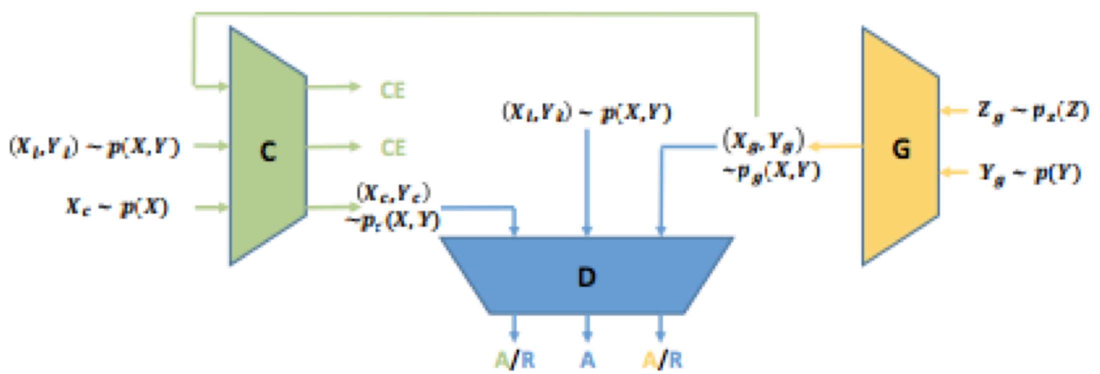
- Adversarially Learned Inference
  - $$\min_G \max_D \mathcal{L}(D, G, I) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log(D(\boldsymbol{x}, I(\boldsymbol{x})))] + \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[1-\log(D(G(\boldsymbol{z}), \boldsymbol{z}))]$$



(Dumoulin et al., 2016)

Related work: TripleGAN

- Three-player game:
  - $$\min_{G}\max_{D}\mathcal{L}(D,G,C)=\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim p_{data}(\boldsymbol{x}),\boldsymbol{y}}[\log(D(\boldsymbol{x},\boldsymbol{y}))]$$
    $$+(1-\alpha)\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}),\boldsymbol{y}\sim p(\boldsymbol{y})}[1-\log(D(G(\boldsymbol{z},\boldsymbol{y}),\boldsymbol{y}))]$$
    $$+\alpha\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}),\boldsymbol{y}\sim p_c(\boldsymbol{y}|\boldsymbol{x})}[1-\log(D(\boldsymbol{x},\boldsymbol{y}))]$$



(Li et al., 2017)

Evaluate controllability of generator

- Use a golden classifier to test the (x,y) pairs generated by different conditional generator

Model	# labeled samples		
	$n = 20$	$n = 50$	$n = 100$
CVAE-semi	33.05	10.72	5.66
TripleGAN	3.06	1.80	1.29
SGAN	**1.68**	**1.23**	**0.93**

Table 1: Errors (%) of generated samples classified by a classifier with 0.56% test error.

Semi-supervised classification

Method	MNIST			SVHN	CIFAR-10
	$n = 20$	$n = 50$	$n = 100$	$n = 1000$	$n = 4000$
Ladder [22]	-	-	**0.89(±0.50)**	-	20.40(±0.47)
VAE [12]	-	-	3.33(±0.14)	36.02(±0.10)	-
CatGAN [28]	-	-	1.39(±0.28)	-	19.58(±0.58)
ALI [5]	-	-	-	7.3	18.3
ImprovedGAN [27]	16.77(±4.52)	2.21(±1.36)	0.93 (±0.07)	8.11(±1.3)	18.63(±2.32)
TripleGAN [15]	5.40(±6.53)	1.59(±0.69)	0.92(±0.58)	5.83(±0.20)	18.82(±0.32)
SGAN	**4.0(±4.14)**	**1.29(±0.47)**	**0.89(±0.11)**	**5.73(±0.12)**	**17.26(±0.69)**

Table 2: Comparisons of semi-supervised classification errors (%) on MNIST, SVHN and CIFAR-10 test sets.

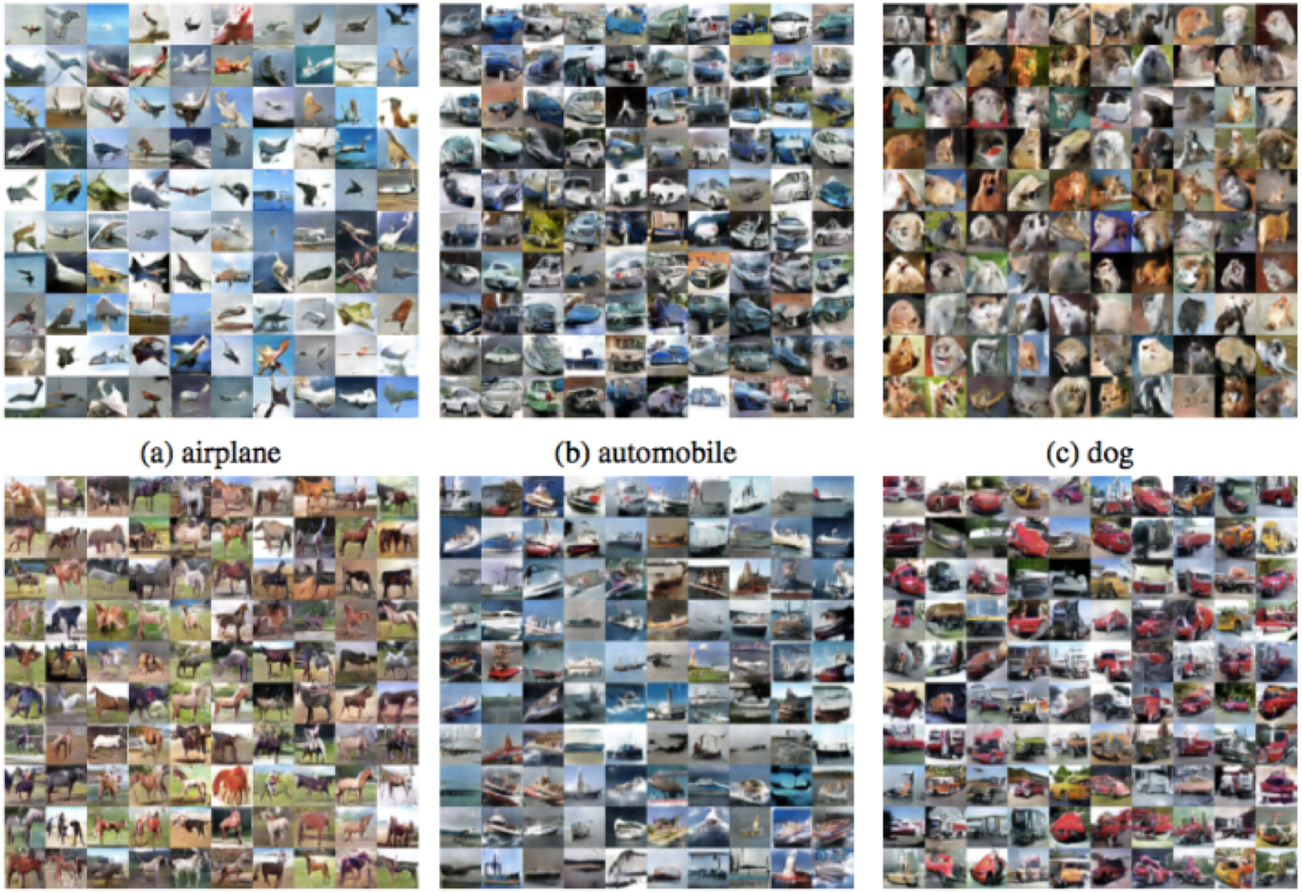
Conditional generation

Conditional generation

Ablation study

(a) w/o $\mathcal{R}_y, \mathcal{R}_z$ (b) w/o $\mathcal{R}_z$ (c) Full model

Visual quality



(a) airplane (b) automobile (c) dog (d) horse (e) ship (f) truck

Visual quality

- inception score： 6.91(±0.07)
  - TripleGAN： 5.08(±0.09)
  - Improved-GAN without MD： 3.87(±0.03)

Image progression

(a) MNIST

(b) SVHN

(c) CIFAR-10

Style transfer

(a) MNIST

(b) SVHN

(c) CIFAR-10

Conclusion

- SGAN
  - Two collaborative games and two adversarial games
  - Mutual bootstrapping trick to improve C and G
  - Disentangled representations and more controllable generator
  - Good performance on a spectrum of downstream applications

Reference

[1] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[C]//Advances in neural information processing systems. [S.l.: s.n.], 2014: 2672–2680.

[2] Kingma D P, Welling M. Auto-encoding variational bayes[J]. arXiv preprint arXiv:1312.6114, 2013.

[3] Salakhutdinov R, Hinton G. Deep boltzmann machines[C]//Artificial Intelligence and Statistics. [S.l.: s.n.], 2009: 448–455.

[4] Liang X, Hu Z, Zhang H, et al. Recurrent topic-transition gan for visual paragraph generation [J]. arXiv preprint arXiv:1703.07022, 2017.

[5] Hu Z, Yang Z, Liang X, et al. Controllable text generation[J]. arXiv preprint arXiv:1703.00955, 2017.

[6] Chen X, Duan Y, Houthooft R, et al. Infogan: Interpretable representation learning by information maximizing generative adversarial nets[C]//Advances in Neural Information Processing Systems. [S.l.: s.n.], 2016: 2172–2180.

[7] Salimans T, Goodfellow I, Zaremba W, et al. Improved techniques for training gans[C]// Advances in Neural Information Processing Systems. [S.l.: s.n.], 2016: 2226–2234.

[8] Donahue J, Krähenbühl P, Darrell T. Adversarial feature learning[J]. arXiv preprint arXiv:1605.09782, 2016.

Reference

[9] Dumoulin V, Belghazi I, Poole B, et al. Adversarially learned inference[J]. arXiv preprint arXiv:1606.00704, 2016.

[10] Yan X, Yang J, Sohn K, et al. Attribute2image: Conditional image generation from visual attributes[C]//European Conference on Computer Vision. [S.l.]: Springer, 2016: 776–791.

[11] Reed S, Akata Z, Yan X, et al. Generative adversarial text to image synthesis[C]//[S.l.: s.n.], .

[12] Reed S E, Akata Z, Mohan S, et al. Learning what and where to draw[C]//Advances in Neural Information Processing Systems. [S.l.: s.n.], 2016: 217–225.

[13] Odena A, Olah C, Shlens J. Conditional image synthesis with auxiliary classifier gans[J]. arXiv preprint arXiv:1610.09585, 2016.

[14] Kingma D P, Mohamed S, Rezende D J, et al. Semi-supervised learning with deep generative models[C]//Advances in Neural Information Processing Systems. [S.l.: s.n.], 2014: 3581–3589.

[15] Li C, Xu K, Zhu J, et al. Triple generative adversarial nets[J]. arXiv preprint arXiv:1703.02291, 2017.

Thanks!

- Questions?