Structured Generative Adversarial Networks

Zhijie Deng*, Hao Zhang*, Xiaodan Liang, Luona Yang, Shizhen Xu, Jun Zhu, Eric P. Xing (* equal contributions)

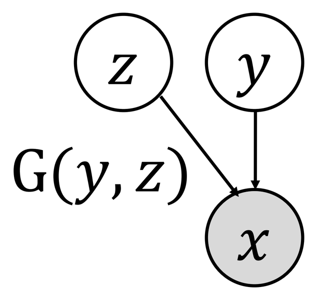
Tsinghua University, Carnegie Mellon University, Petuum Inc.

Problem

Semi-supervised conditional generative modeling

- ➢ Conditional generative models are quite useful
  - Generate data samples with designated semantics
  - Synthetic data help supervised training of downstream tasks
- ➢ Challenges: labels are scarce
  - How to accurately capture the conditions during generating process?
  - How to separate semantics of interest from other factors of variations?
- ➢ Problem: ensure *controllability* and *disentanglability*
  - *Controllability*: the ability to conditionally generate data strictly following the designated semantics
  - *Disentanglability*: the ability to disentangle the modeled semantic of interest from other factors

Intuition

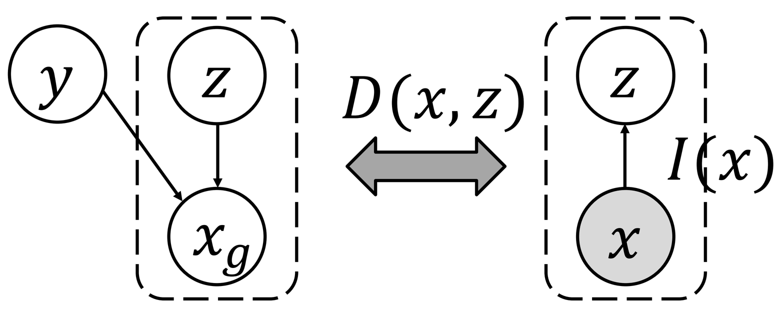


- ➢ Hidden space shall be structured as
  - Semantic of our interest $\boldsymbol{y}$
  - Other factors of variations $\boldsymbol{z}$
- ➢ Hence our goal: learn a generated model $p_g(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{z})$ with
  - *Controllability*: semantics of our interest are fully captured by y
  - *Disentangliblity*: y and z are not cluttered as much as possible
  - However, directly learning $p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$ is difficult

Characterizing $p(\boldsymbol{x},\boldsymbol{z})$ *and* $p(\boldsymbol{x},\boldsymbol{y})$ instead of $p(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$

Model

- ➢ Step 1: Learn joint distribution $p(\boldsymbol{x},\boldsymbol{z})$
  - Introduce an inference network $I(\boldsymbol{x})\colon \boldsymbol{x}\to\boldsymbol{z}$
  - Estimate $p(\boldsymbol{z}|\boldsymbol{x})$ via adversarial learned inference

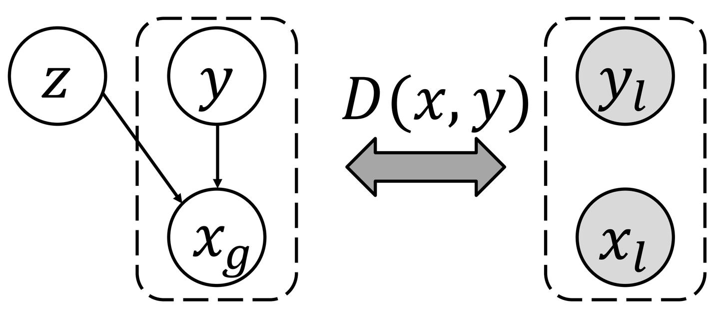


$$p_g(\boldsymbol{x},\boldsymbol{z}) = p(\boldsymbol{z})p_g(\boldsymbol{x}|\boldsymbol{z})$$

⇕ Match

$$p_i(\boldsymbol{x},\boldsymbol{z}) = p(\boldsymbol{x})p_g(\boldsymbol{z}|\boldsymbol{x})$$

- ➢ Step 2: Learn joint distribution $p(\boldsymbol{x},\boldsymbol{y})$
  - Estimate $p(\boldsymbol{y}|\boldsymbol{x})$ via adversarial learned inference

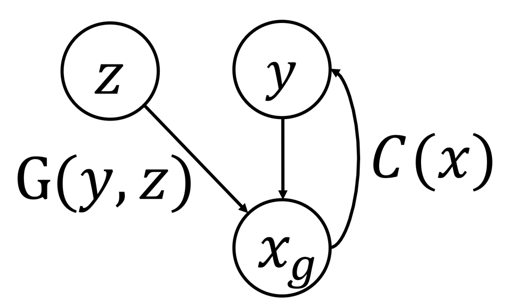


$$p_g(\boldsymbol{x},\boldsymbol{y}) = p(\boldsymbol{y})p_g(\boldsymbol{x}|\boldsymbol{y})$$

⇕ Match

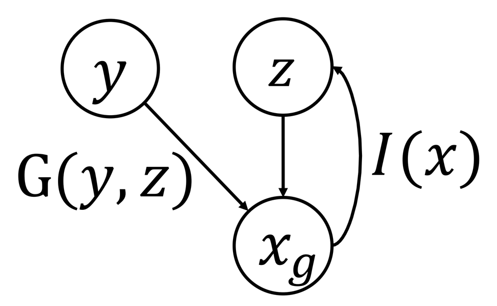
$$(\boldsymbol{x}_l,\boldsymbol{y}_l) \sim p(\boldsymbol{x},\boldsymbol{y})$$

- ➢ Step 3: Enforce $\boldsymbol{y}$ to capture all semantic of interest
  - Therefore, enhance the controllability of the generator
  - Introduce an inference network $C(\boldsymbol{x})\colon \boldsymbol{x}\to\boldsymbol{y}$
  - Minimize reconstruction error $\mathcal{R}_y$



$$\min_{C,G} \mathcal{R}_y = -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim p(\boldsymbol{x},\boldsymbol{y})}[\log p_c(\boldsymbol{y}|\boldsymbol{x})] - \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim p_g(\boldsymbol{x},\boldsymbol{y})}[\log p_c(\boldsymbol{y}|\boldsymbol{x})]$$

- ➢ Step 4: Enforce $\boldsymbol{z}$ to capture other factors of variations
  - Therefore, enhance the disentanglability of the generator
  - Reuse the inference network $I(\boldsymbol{x})\colon \boldsymbol{x}\to\boldsymbol{z}$
  - Minimize reconstruction error $\mathcal{R}_z$



$$\min_{I,G} \mathcal{R}_z = -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{z})\sim p_g(\boldsymbol{x},\boldsymbol{z})}[\log p_i(\boldsymbol{z}|\boldsymbol{x})]$$

Training

- ➢ Key training techniques
  - Augment labeled dataset with $(\boldsymbol{x}_c,\boldsymbol{y}_c)\sim p_c(\boldsymbol{x},\boldsymbol{y})$
  - Mix $(\boldsymbol{x}_c,\boldsymbol{y}_c)\sim p_c(\boldsymbol{x},\boldsymbol{y})$, $(\boldsymbol{x}_g,\boldsymbol{y}_g)\sim p_g(\boldsymbol{x},\boldsymbol{y})$ with labeled data with appropriate mixing proportion

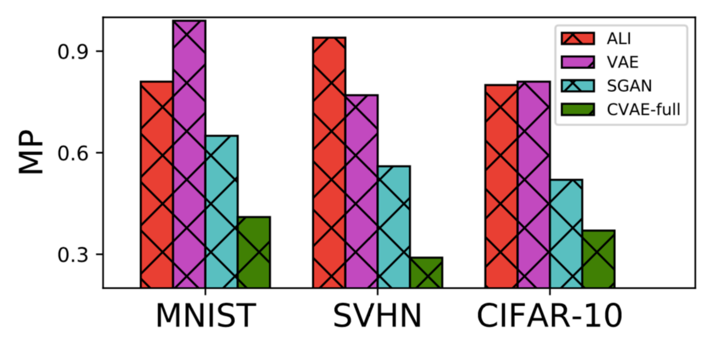
Algorithm 1 Training Structured Generative Adversarial Networks (SGAN).

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 $(\boldsymbol{x},\boldsymbol{y})$: $(\boldsymbol{x}_l,\boldsymbol{y}_l)\sim p(\boldsymbol{x},\boldsymbol{y})$, $(\boldsymbol{x}_g,\boldsymbol{y}_g)\sim p_g(\boldsymbol{x},\boldsymbol{y})$, $\boxed{(\boldsymbol{x}_c,\boldsymbol{y}_c)\sim p_c(\boldsymbol{x},\boldsymbol{y})}$.
  Obtain a batch $\boxed{(\boldsymbol{x}_m,\boldsymbol{y}_m)}$ 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 $\boxed{(\boldsymbol{x}_m,\boldsymbol{y}_m)}$ 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 $\boxed{(\boldsymbol{x}_m,\boldsymbol{y}_m)}$ (see text).
  Train G by minimizing $\mathcal{L}_{\boldsymbol{xy}} + \mathcal{L}_{\boldsymbol{xz}} + \mathcal{R}_{\boldsymbol{y}} + \mathcal{R}_{\boldsymbol{z}}$ using $(\boldsymbol{x}_g,\boldsymbol{y}_g)$.
until convergence.

Results

- ➢ Improved controllability and disentanglability
  - Evaluate controllability: generate samples with designated semantics, classify the samples using gold classifiers
  - Evaluate disentanglability: mutual predictability measure (MP)

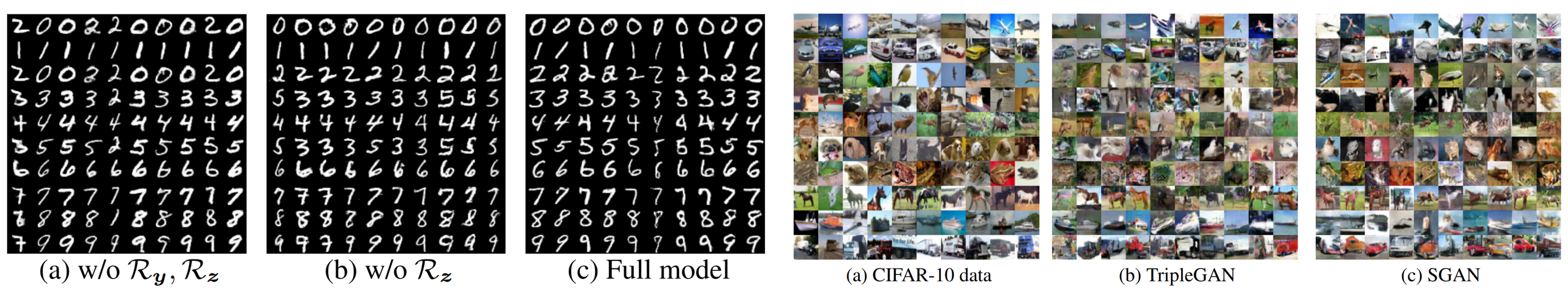
Model	# labeled samples		
	$n=20$	$n=50$	$n=100$
CVAE-semi	33.05	10.72	5.66
TripleGAN	3.06	1.80	1.29
SGAN	**1.68**	**1.23**	**0.93**



- ➢ Better results on semi-supervised classification
  - State-of-the-art results across multiple standard datasets
  - More advantages at low-shot settings

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)**

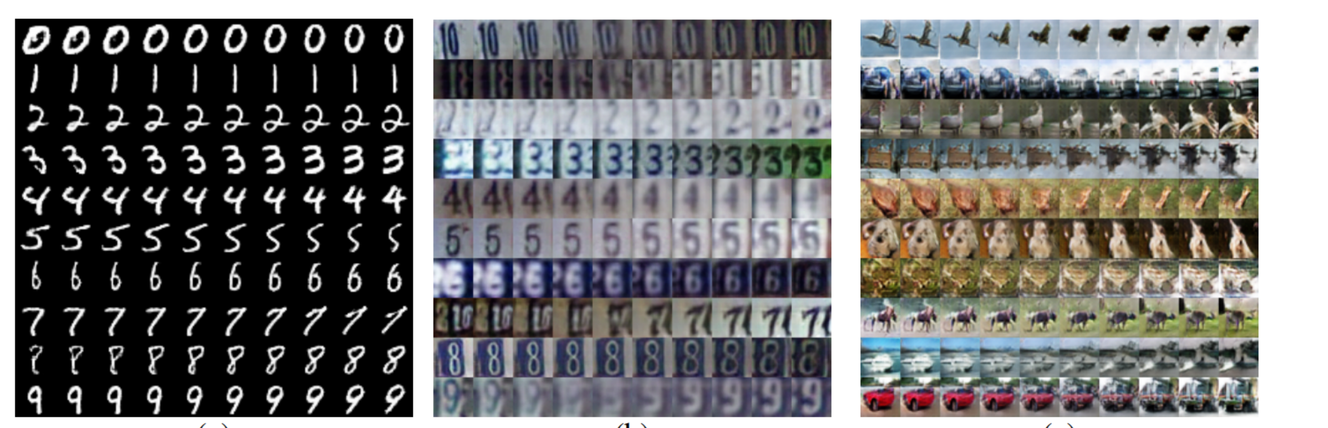
- ➢ Controllable generation
  - Ablation studies reveal that $\mathcal{R}_y$ and $\mathcal{R}_z$ help align the semantics
- ➢ Visual quality
  - Report an inception score 6.91(±0.07), higher than that of TripleGAN and Improved-GAN w/o minibatch discrimination



(a) w/o $\mathcal{R}_y, \mathcal{R}_z$ (b) w/o $\mathcal{R}_z$ (c) Full model (a) CIFAR-10 data (b) TripleGAN (c) SGAN

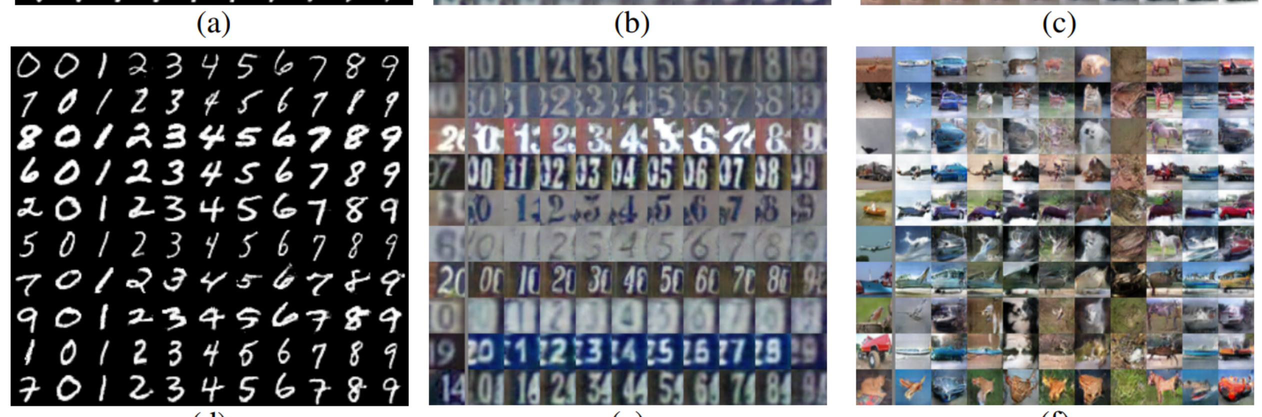
- ➢ SGAN enables more interesting applications
  - Image progression: generate images with interpolated $\boldsymbol{z}$ – SGAN generalizes instead of memorizing data
  - Style transfer: infer $\boldsymbol{z}$ given an image, generate a new image with the same $\boldsymbol{z}$ but different semantic of interest $\boldsymbol{y}$

Image progression



(a) (b) (c)

Style transfer



(d) (e) (f)