Understanding Generative Adversarial Networks

Balaji Lakshminarayanan

DeepMind

Joint work with:
Shakir Mohamed, Mihaela Rosca, Ivo Danihelka, David Warde-Farley, Liam Fedus, Ian Goodfellow, Andrew Dai & others

Problem statement

Learn a **generative model**:

$$\mathbf{x} = \mathcal{G}_{\theta}(\mathbf{z}'); \qquad \mathbf{z}' \sim q(\mathbf{z})$$

Goal: given samples $x_1 \ldots x_n$ from true distribution $p^*(x)$, find θ

$p_\theta(x)$ is not available -> can't maximize density directly

However, we can **sample from $p_\theta(x)$ efficiently**

High level overview of GANs

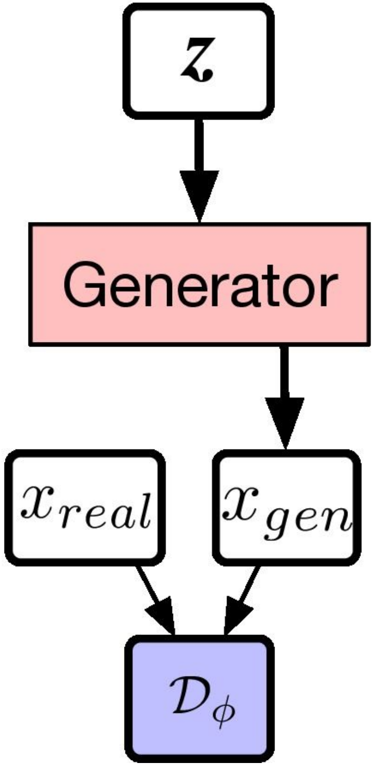
Goodfellow et al. 2014

Discriminator: Train a classifier to distinguish between the two distributions *using samples*

Generator: Train to generate samples that fool the discriminator

Minimax game alternates between training discriminator and generator

- Nash equilibrium corresponds to minima of Jensen Shannon divergence
- Need a bunch of tricks to stabilize training in practice

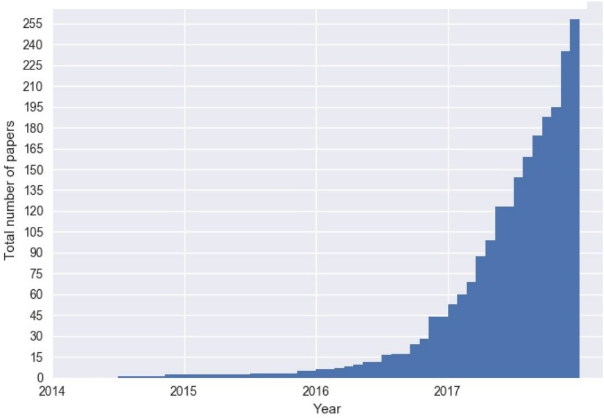


GANs: Hope or Hype?

Ferenc Huszar
@fhuszar

practicin' the alphabet with my son:
A is for AffGAN
B is for B-GAN
C is for Conditional GAN
D is for DCGAN
E is for EBGAN
F is for f-GAN

https://github.com/junyanz/CycleGAN/blob/master/imgs/horse2zebra.gif



https://github.com/hindupuravinash/the-gan-zoo

https://github.com/tkarras/progressive_growing_of_gans

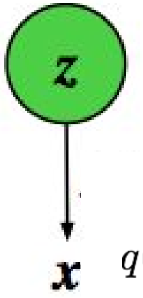
How do GANs relate to other ideas in probabilistic machine learning?

Learning in implicit generative models

Shakir Mohamed and Balaji Lakshminarayanan**

Implicit Models

Stochastic procedure that generates data



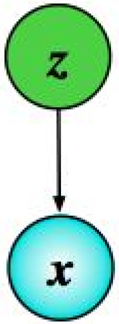
$$\mathbf{x} = \mathcal{G}(\mathbf{z}'), \quad \mathbf{z}' \sim q(\mathbf{z})$$

$$q(\mathbf{x}) = \frac{\partial}{\partial x_1} \cdots \frac{\partial}{\partial x_d} \int_{\{\mathcal{G}(\mathbf{x}) \leq \mathbf{x}\}} q(\mathbf{z}) d^m \mathbf{z}$$

Examples: stochastic simulators of complex physical systems (climate, ecology, high-energy physics etc)

Prescribed Models

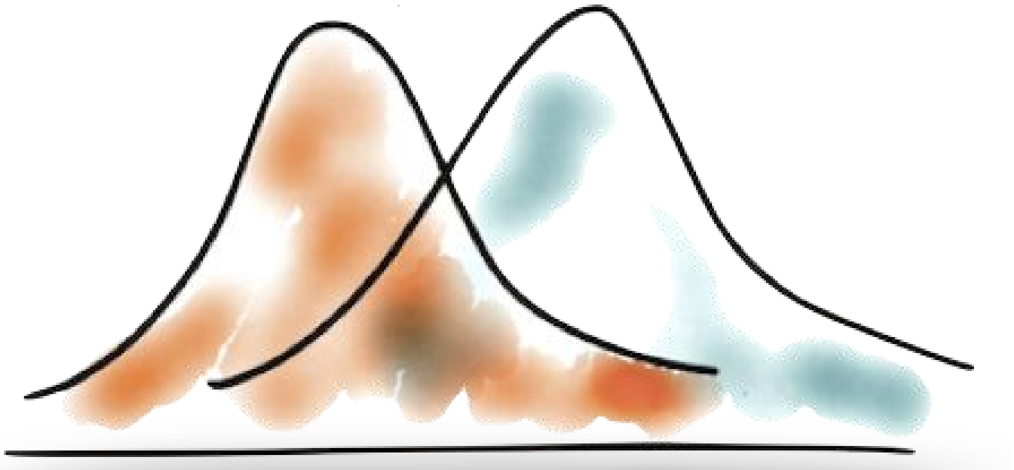
Provide knowledge of the probability of observations & specify a *conditional* log-likelihood function.

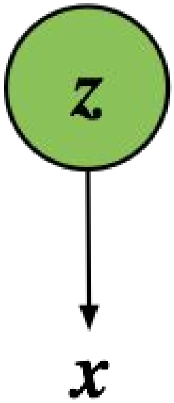


$$\mathbf{x}' \sim p(\mathbf{x}|\mathcal{G}(\mathbf{z}')), \quad \mathbf{z}' \sim q(\mathbf{z})$$

$$q(\mathbf{x}) = \int p(\mathbf{x}|\mathcal{G}(\mathbf{z}))q(\mathbf{z})$$

Learning by Comparison





We compare the estimated distribution to the true distribution **using samples.**

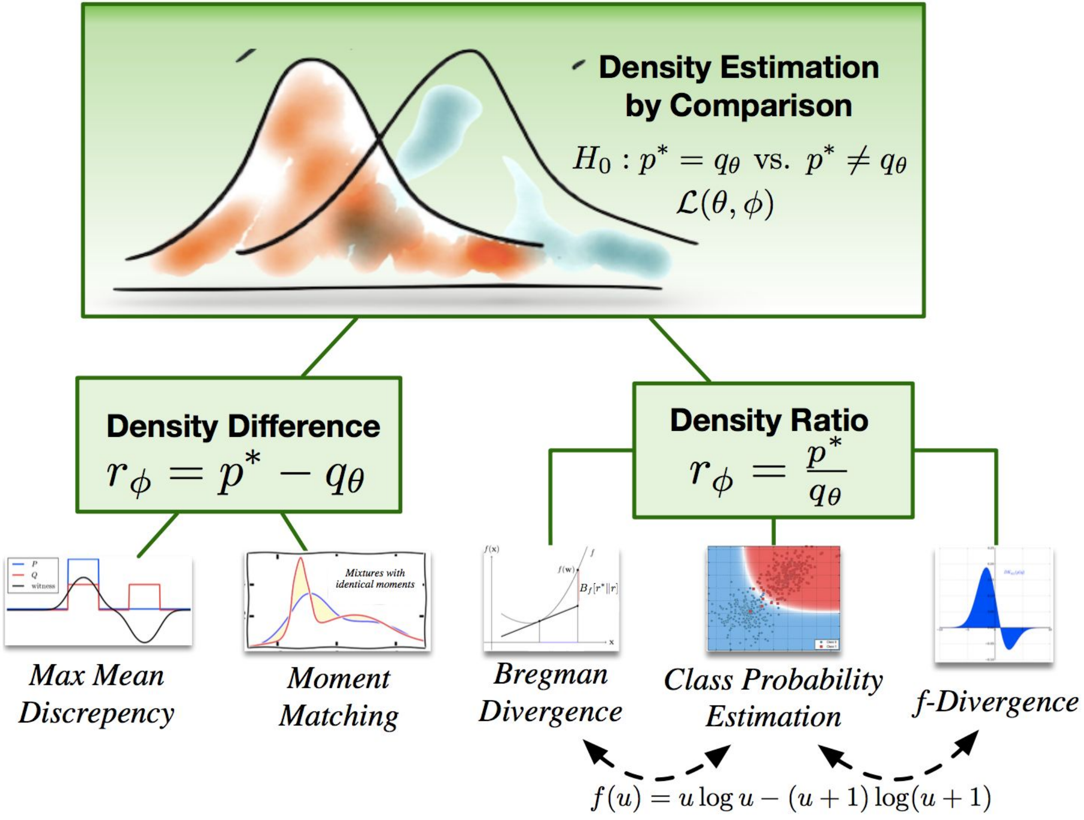
Learning by Comparison

Comparison

Use a hypothesis ***test or comparison*** to build an auxiliary model to indicate how data simulated from the model differs from observed data.

Learning generator

Adjust model parameters to better match the data distribution using the comparison.



High-level idea

Define a joint loss function $L(\varphi, \theta)$ and alternate between:

Comparison loss *("discriminator")*: **arg min$_\varphi$ $L(\varphi, \theta)$**

Generative loss: **arg min$_\theta$ $-L(\varphi, \theta)$**

Global optimum is $q_\theta = p^*$ and

- Density ratio $r_\varphi = 1$ or
- Density difference $r_\varphi = 0$

How do we compare distributions?

Density Ratios and Classification

Density Ratio $\frac{p^*(\mathbf{x})}{q(\mathbf{x})}$

Bayes' Rule $p(\mathbf{x}|y) = \frac{p(y|\mathbf{x})p(\mathbf{x})}{p(y)}$

Real Data | **Simulated Data**

Combine data $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\} = \{\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_{\hat{n}}, \tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_{\tilde{n}}\}$

Assign labels $\{y_1, \ldots, y_N\} = \{+1, \ldots, +1, -1, \ldots, -1\}$

Sugiyama et al, 2012

Density Ratios and Classification

Density ratio

$$\frac{p^*(\mathbf{x})}{q(\mathbf{x})} = \frac{p(\mathbf{x}|y=1)}{p(\mathbf{x}|y=-1)}$$

Bayes' substitution

$$= \frac{p(y=+1|\mathbf{x})p(\mathbf{x})}{p(y=+1)} \Big/ \frac{p(y=-1|\mathbf{x})p(\mathbf{x})}{p(y=-1)}$$

Class probability

$$\frac{p^*(\mathbf{x})}{q(\mathbf{x})} = \frac{p(y=1|\mathbf{x})}{p(y=-1|\mathbf{x})}$$

Computing a density ratio is equivalent to class probability estimation.

Class Probability Estimation

$$\mathcal{D}(\mathbf{x};\phi) = p(\mathbf{y} = +1|\mathbf{x}) = \frac{r}{r+1}$$

Loss	Objective Function ($\mathcal{D} := \mathcal{D}(\mathbf{x};\phi)$)
Bernoulli loss	$\pi\mathbb{E}_{p^*(\mathbf{x})}[-\log\mathcal{D}] + (1-\pi)\mathbb{E}_{q_\theta(\mathbf{x})}[-\log(1-\mathcal{D})]$
Brier score	$\pi\mathbb{E}_{p^*(\mathbf{x})}[(1-\mathcal{D})^2] + (1-\pi)\mathbb{E}_{q_\theta(\mathbf{x})}[\mathcal{D}^2]$
Exponential loss	$\pi\mathbb{E}_{p^*(\mathbf{x})}\left[\left(\frac{1-\mathcal{D}}{\mathcal{D}}\right)^{\frac{1}{2}}\right] + (1-\pi)\mathbb{E}_{q_\theta(\mathbf{x})}\left[\left(\frac{\mathcal{D}}{1-\mathcal{D}}\right)^{\frac{1}{2}}\right]$
Misclassification	$\pi\mathbb{E}_{p^*(\mathbf{x})}[\mathbb{I}[\mathcal{D} \leq 0.5]] + (1-\pi)\mathbb{E}_{q_\theta(\mathbf{x})}[\mathbb{I}[\mathcal{D} > 0.5]]$
Hinge loss	$\pi\mathbb{E}_{p^*(\mathbf{x})}\left[\max\left(0, 1-\log\frac{\mathcal{D}}{1-\mathcal{D}}\right)\right] + (1-\pi)\mathbb{E}_{q_\theta(\mathbf{x})}\left[\max\left(0, 1+\log\frac{\mathcal{D}}{1-\mathcal{D}}\right)\right]$
Spherical	$\pi\mathbb{E}_{p^*(\mathbf{x})}[-\alpha\mathcal{D}] + (1-\pi)\mathbb{E}_{q_\theta(\mathbf{x})}[-\alpha(1-\mathcal{D})]; \quad \alpha = (1-2\mathcal{D}+2\mathcal{D}^2)^{-1/2}$

Table 1. Proper scoring rules that can be minimised in class probability-based learning of implicit generative models.

Other loss functions for training classifier, e.g. Brier score leads to LS-GAN

Related: Unsupervised as Supervised Learning, Classifier ABC

Divergence minimization (f-GAN)

$$D_f\left[p^*(\mathbf{x})\|q_\theta(\mathbf{x})\right] = \int q_\theta(\mathbf{x}) f\left(\frac{p^*(\mathbf{x})}{q_\theta(\mathbf{x})}\right) d\mathbf{x}$$
$$= \mathbb{E}_{q_\theta(\mathbf{x})}\left[f(r(\mathbf{x}))\right]$$
$$\geq \sup_t \mathbb{E}_{p^*(\mathbf{x})}[t(\mathbf{x})] - \mathbb{E}_{q_\theta(\mathbf{x})}[f^\dagger(t(\mathbf{x}))]$$

Minimize a lower bound on f-divergence between p* and q_θ

Choices of *f* recover KL(p*||q) (maximum likelihood), KL(q||p*) and JS(p*||q)

Can use different f-divergences for learning ratio vs learning generator

Density ratio estimation

$$
\begin{aligned}
&B_f(r^*(\mathbf{x}) \| r_\phi(\mathbf{x})) \\
&= \int \left(f(r^*(\mathbf{x})) - f(r_\phi(\mathbf{x})) - f'(r_\phi(\mathbf{x})) \left[r^*(\mathbf{x}) - r_\phi(\mathbf{x}) \right] \right) q_\theta(\mathbf{x}) d\mathbf{x} \\
&= \mathbb{E}_{q_\theta(\mathbf{x})} \left[r_\phi(\mathbf{x}) f'(r_\phi(\mathbf{x})) - f(r_\phi(\mathbf{x})) \right] - \mathbb{E}_{p^*}[f'(r_\phi(\mathbf{x}))] + D_f[p^*(\mathbf{x}) \| q_\theta(\mathbf{x})] \\
&= \mathcal{L}_B(r_\phi(\mathbf{x})) + D_f[p^*(\mathbf{x}) \| q_\theta(\mathbf{x})]
\end{aligned}
$$

Optimize a Bregman divergence between r* and r_φ

Special cases include least squares importance fitting (LSIF)

Ratio loss ends up being identical to that of f-divergence

Moment-matching

$$\mathcal{L}(\phi, \boldsymbol{\theta}) = (\mathbb{E}_{p^*(\mathbf{x})}[s(\mathbf{x})] - \mathbb{E}_{q_\theta(\mathbf{x})}[s(\mathbf{x})])^2$$
$$= (\mathbb{E}_{p^*(\mathbf{x})}[s(\mathbf{x})] - \mathbb{E}_{q(\mathbf{z})}[s(\mathcal{G}(\mathbf{z}; \boldsymbol{\theta}))])^2$$

Used by

- Generative moment matching networks
- Training generative neural networks via Maximum Mean Discrepancy optimization

Connects to optimal transport literature (e.g. Wasserstein GAN)

Summary of the approaches

Class probability estimation

- Build a classifier to distinguish real from fake samples.
- Original GAN solution.

Density ratio matching

- Directly minimise the expected error between the true ratio and an estimate of it.

Divergence Minimisation

- Minimise a generalised divergence between the true density p* and the product r(x)q(x).
- *f*-GAN approach.

Moment matching

- Match the moments between p* and r(x)q(x)
- MMD, optimal transport, etc.

How do we learn generator?

In GANs, the generator is **differentiable**

- Generator loss is of the following form e.g. f-divergence D_f = E_q [f(r)]
- Can apply **re-parametrization trick**

$$
\begin{aligned}
J &= E_{q_\theta(\mathbf{x})}[\ell(\mathbf{x})] &&= E_{q(\mathbf{z})}[\ell(\mathcal{G}_{\boldsymbol{\theta}}(\mathbf{z}))] \\
\nabla_{\boldsymbol{\theta}} J &= \nabla_{\boldsymbol{\theta}} E_{q(\mathbf{z})}[\ell(\mathcal{G}_{\boldsymbol{\theta}}(\mathbf{z}))] &&= E_{q(\mathbf{z})}[\nabla_{\boldsymbol{\theta}} \ell(\mathcal{G}_{\boldsymbol{\theta}}(\mathbf{z}))]
\end{aligned}
$$

Choice of f-divergence

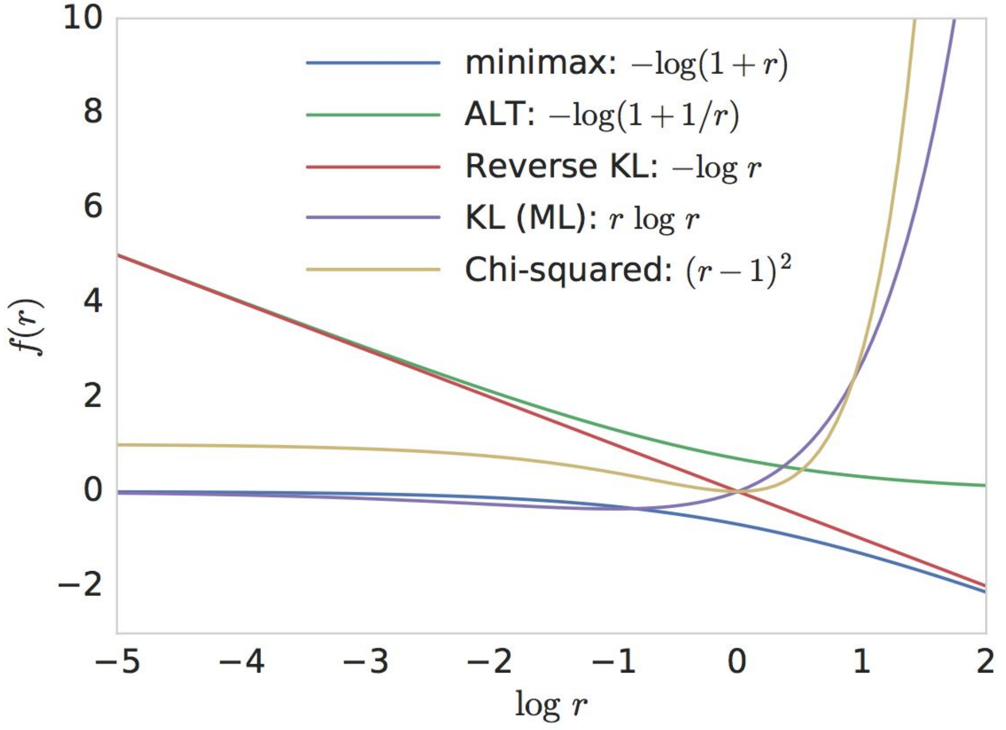


Figure 2. Objective functions for different choices of f.

Density ratio estimation literature has investigated choices of f

However, that's only half of the puzzle. We need non-zero gradients for D_f = E_q [f(r)] to learn generator

- r<<1 early on in training
- Non-saturating alternative loss

We also need additional constraints on the discriminator

Summary: Learning in Implicit Generative Models

Unifying view* of GANs that connects to literature on

- Density ratio estimation
  - … but they don’t focus on learning generator
- Approximate Bayesian computation (ABC) and likelihood-free inference
  - Low dimensional, better understanding of theory
  - Bayesian inference over parameters
  - Simulators are usually not differentiable (can we approximate them?)

Motivates new loss functions: can decouple generator loss from discriminator loss

GAN-like ideas can be used in other places where density ratio appears

Comparing GANs to Maximum Likelihood training using Real-NVP

Comparison of maximum likelihood and GAN-based training of Real NVPs

Ivo Danihelka, Balaji Lakshminarayanan, Benigno Uria, Daan Wierstra and Peter Dayan

Generative Models and Algorithms

Model	Inference	Algorithm
Prescribed Models Directed latent variable models, DLGM, state space	**Maximum Marginal Likelihood** Variational Inference	**VAE** Lower bound on likelihood
Implicit Models Generator nets, normalising flows, SDEs, mechanistic simulations	**Hypothesis Test** Likelihood ratio and Bayes risk	**GAN**

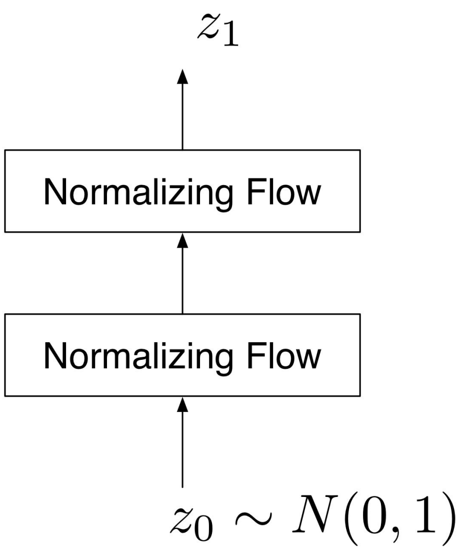
Generative Models and Algorithms

Model	Inference	Algorithm
Prescribed Models Directed latent variable models, DLGM, state space	**Maximum Marginal Likelihood** Variational Inference	**VAE** Lower bound on likelihood
Implicit Models Generator nets, normalising flows, SDEs, mechanistic simulations	**Hypothesis Test** Likelihood ratio and Bayes risk	**GAN**

Comparing inference algorithms for a fixed model

Generator is Real NVP (Dinh et al., 2016)

$$\log P(z_1) = \log P(z_0) - \log |\det \frac{dz_1}{dz_0}|$$



1. Train by maximum likelihood (MLE).
2. Train a generator by Wasserstein GAN.
3. Compare.

Complementary to “On the quantitative analysis of decoder-based models” by Wu et al., 2017

Wasserstein GAN

For general distributions:

$$W_d(P_r, P_g) = \sup_{\|f\|_L \leq 1} \mathop{\mathbb{E}}_{x_r \sim P_r}[f(x_r)] - \mathop{\mathbb{E}}_{x_g \sim P_g}[f(x_g)]$$

Considering all 1-Lipschitz function
(i.e., functions with **bounded derivatives**).

f(x) is a "*critic*".
The critic should give
high value to real samples and
low value to generated samples.

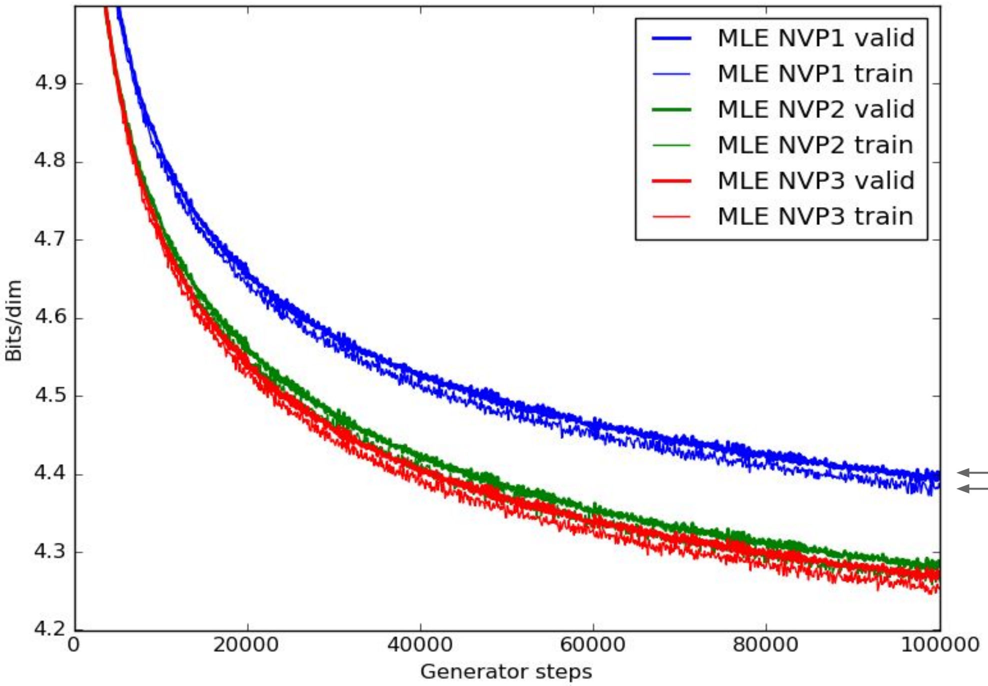
Bounded by:

a) Weight clipping (Wasserstein GAN; "WGAN").
b) Gradient penalty (Improved Training; "WGAN-GP")

Idea: use an independent Wasserstein critic to evaluate generators

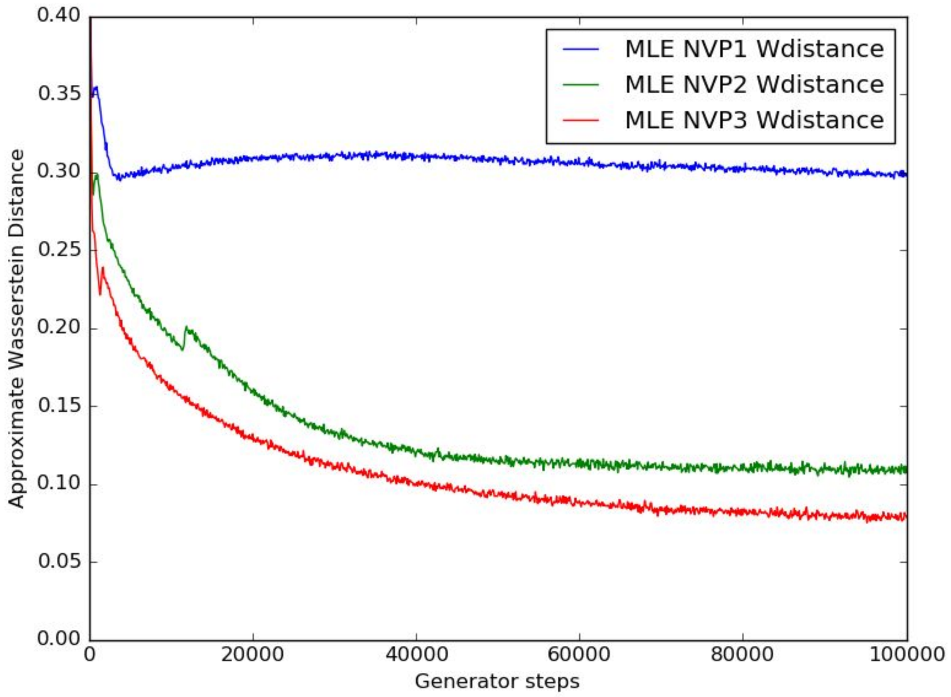
Bits/dim for NVP

Dataset:
CelebA 32x32.

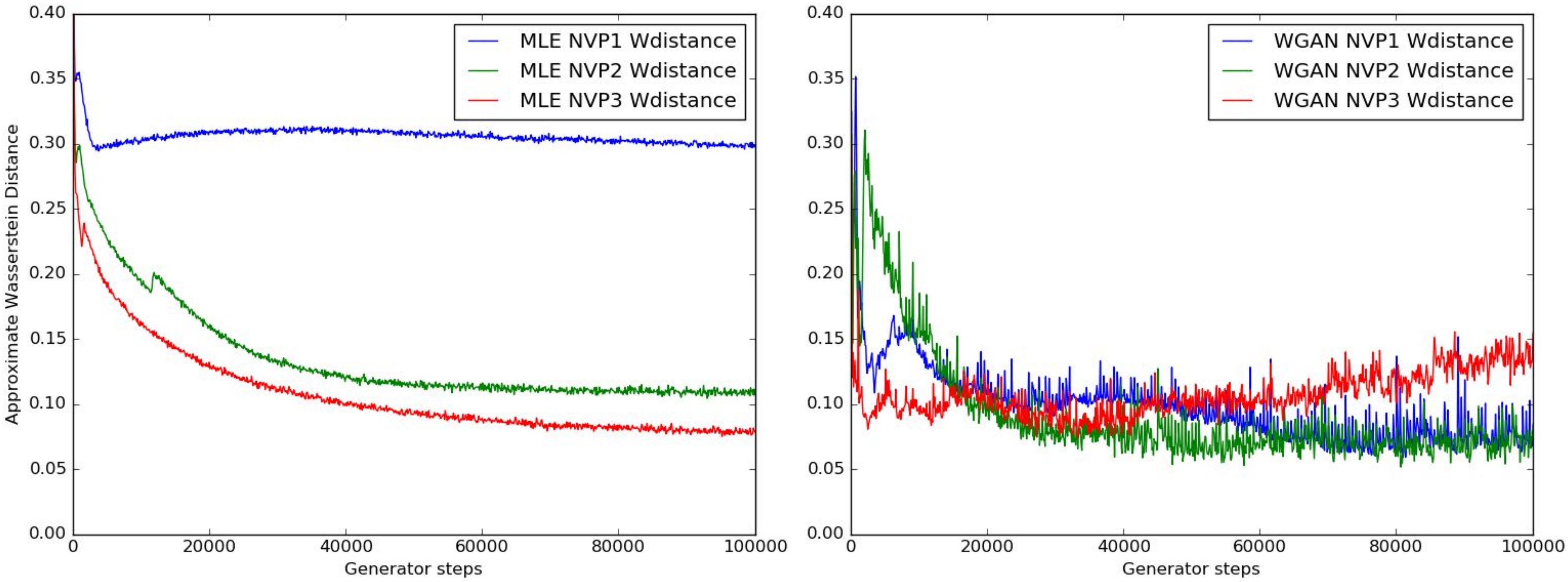


Valid
Train

Wasserstein Distance for NVPs



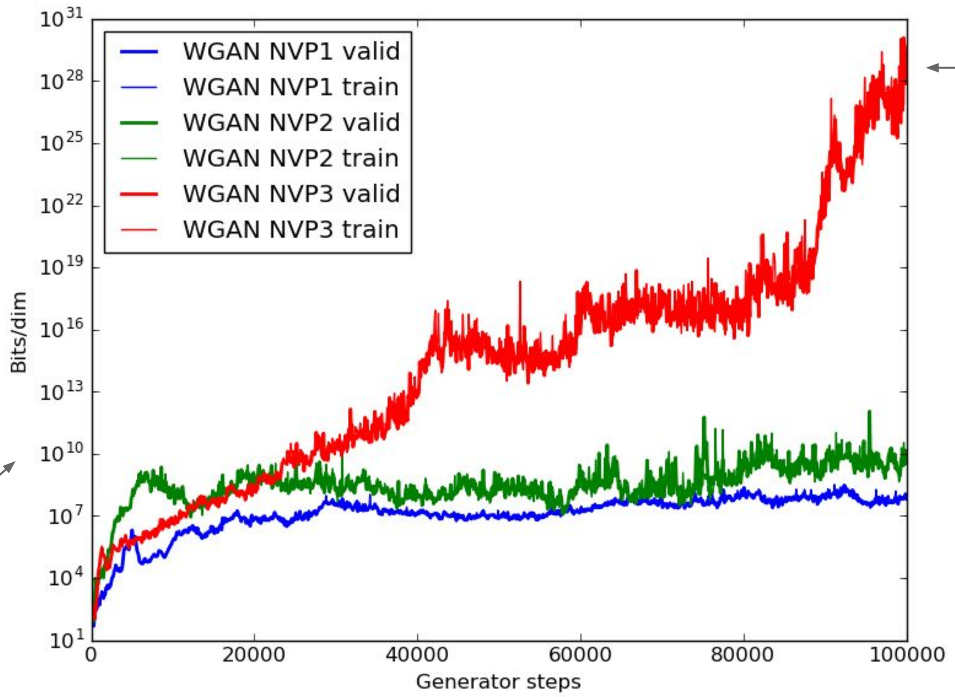
Wasserstein Distance Minimized by WGAN



MLE vs. WGAN Training

MLE vs. WGAN Training (shallower generator)

Bits/dim for NVPs Trained by WGAN



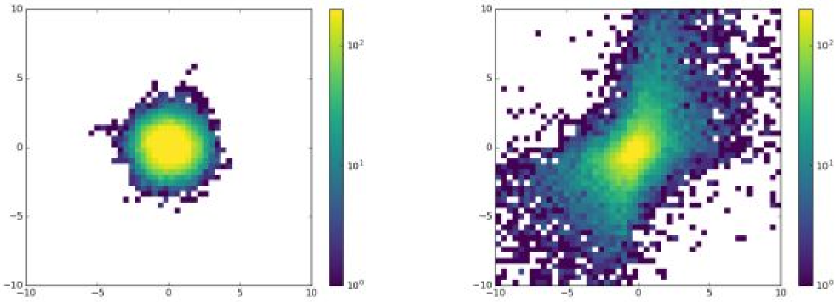
Valid
Train

Log scale

Worse than uniform pdf with 8 bits/dim.

Summary

- Wasserstein distance can compare models.
- Wasserstein distance can be approximated by training a critic.
- Training by WGAN leads to nicer samples but significantly worse log-probabilities.
- Latent codes from WGAN training are non-Gaussian

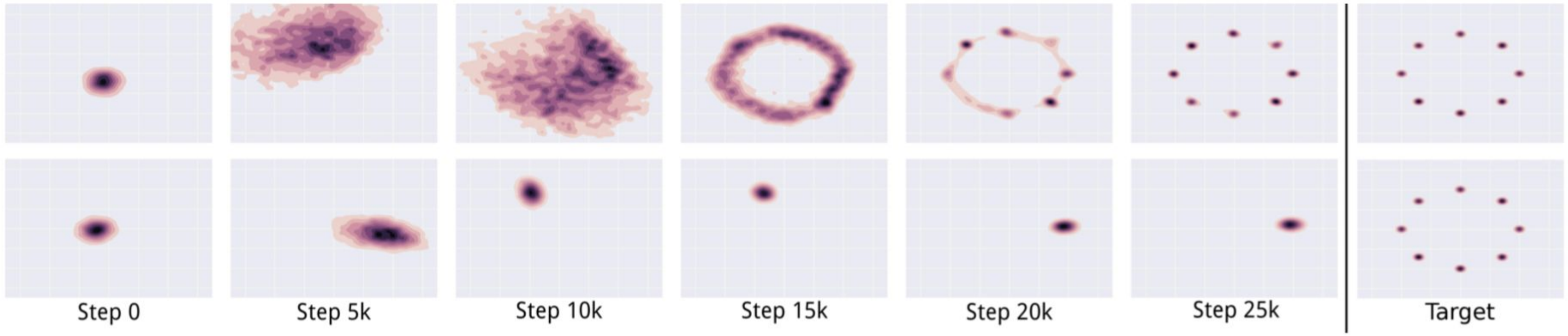


How do we combine VAEs and GANs to get the best of both worlds?

Variational approaches for auto-encoding generative adversarial networks

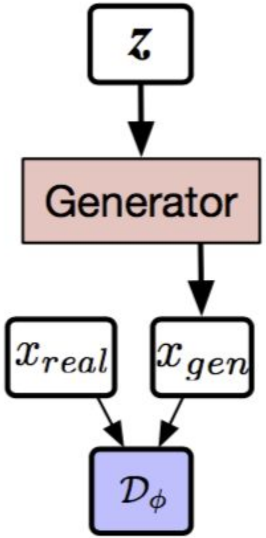
Mihaela Rosca, Balaji Lakshminarayanan*, David Warde-Farley and Shakir Mohamed*

Motivating problem: Mode collapse

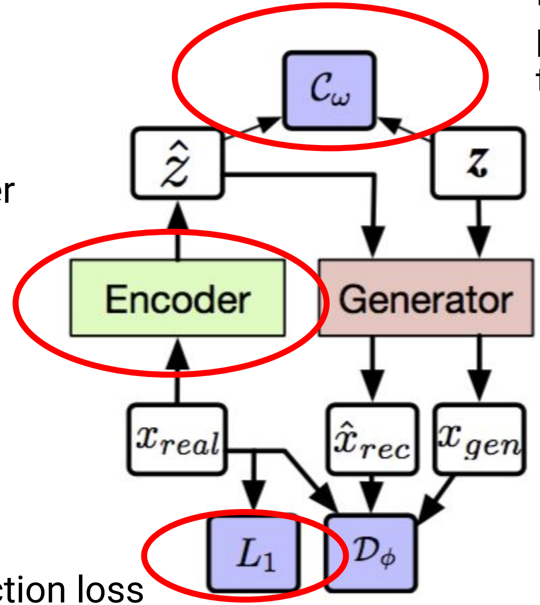


- MoG toy example from “Unrolled GAN” paper
- VAEs have other problems, but do not suffer from mode-collapse
    - Can we add auto-encoder to GANs?

Adding auto-encoder to GANs



Implicit encoder



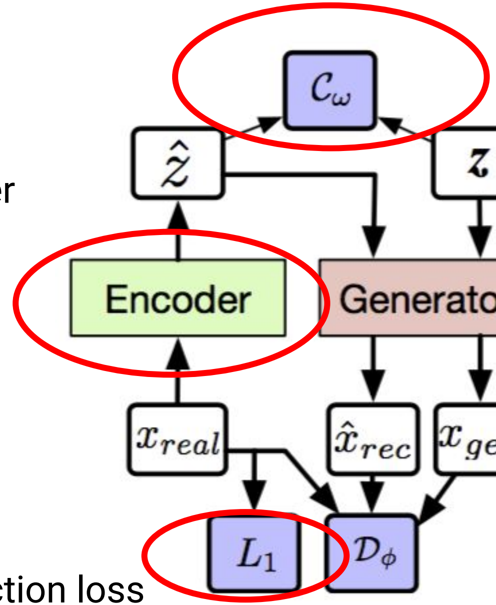
Penalty to match posterior codes to prior

Reconstruction loss

How does it relate to Evidence Lower Bound (ELBO) in VAEs?

Penalty to match posterior codes to prior

Implicit encoder



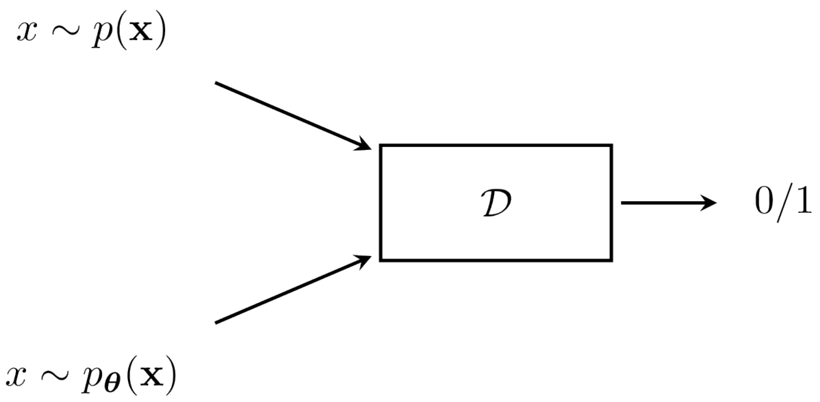
Reconstruction loss

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \geq \mathbb{E}_{q_\eta(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] - \mathbf{KL}[q_\eta(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]$$

Recap: Density ratio trick

Estimate the ratio of two distributions only from samples, by building a binary **classifier** to distinguish between them.

$$\frac{p(\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{x})} = \frac{\mathcal{D}(x)}{1 - \mathcal{D}(x)}$$



Revisiting ELBO in Variational Auto-Encoders

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \geq \mathbb{E}_{q_\eta(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] - \mathrm{KL}[q_\eta(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]$$

LIKELIHOOD TERM

$$\mathbb{E}_{q_\eta(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] = \mathbb{E}_{q_\eta(\mathbf{z}|\mathbf{x})}[\log(\frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})}{p(\mathbf{x})}p(\mathbf{x}))]$$

$$= \mathbb{E}_{q_\eta(\mathbf{z}|\mathbf{x})}[\log \underbrace{\frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})}{p(\mathbf{x})}}_{\text{ratio}}] + \underbrace{\mathbb{E}_{q_\eta(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x})]}_{\text{constant}}$$

$$\mathbf{x} \sim p^*(\mathbf{x}) \longrightarrow \boxed{\mathcal{D}} \longrightarrow 0/1$$

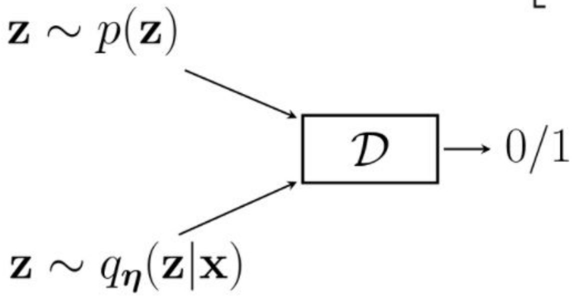
$$\mathbf{x} \sim p_{\boldsymbol{\theta}}(\mathbf{x}) \longrightarrow \boxed{\mathcal{D}}$$

Revisiting ELBO in Variational Auto-Encoders

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \geq \mathbb{E}_{q_\eta(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] - \mathrm{KL}[q_\eta(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]$$

THE KL TERM

$$-\mathrm{KL}[q_\eta(\mathbf{z}|\mathbf{x})||p(\mathbf{z})] = \mathbb{E}_{q_\eta(\mathbf{z}|\mathbf{x})}\left[\log \frac{p(\mathbf{z})}{q_\eta(\mathbf{z}|\mathbf{x})}\right]$$

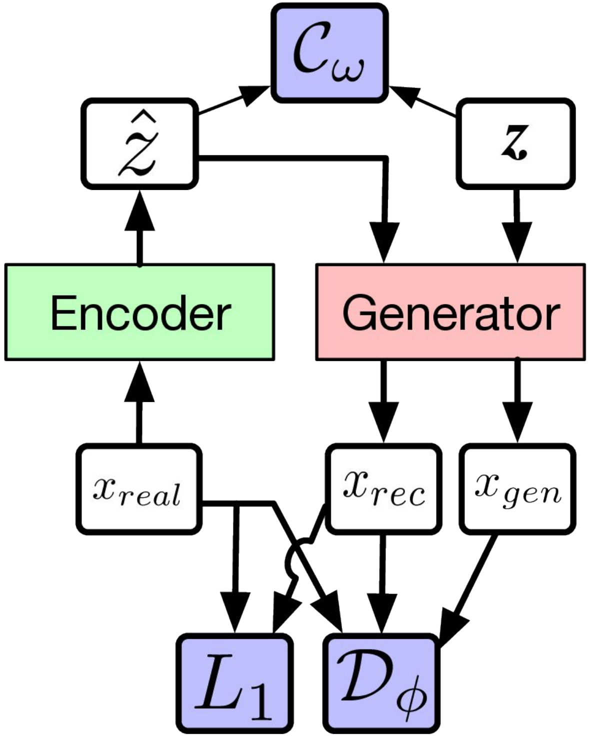


Encoder can be implicit!

More flexible distributions

Putting it all together

$$-\mathrm{KL}[q_\eta(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})] \approx \mathbb{E}_{q_\eta(\mathbf{z}|\mathbf{x})}\left[\log \frac{\mathcal{C}_{\boldsymbol{\omega}}(\mathbf{z})}{1-\mathcal{C}_{\boldsymbol{\omega}}(\mathbf{z})}\right]$$



$$\mathbb{E}_{q_\eta(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] \approx \left[\log \frac{\mathcal{D}_{\boldsymbol{\phi}}(\mathcal{G}_{\boldsymbol{\theta}}(\mathbf{z}))}{1-\mathcal{D}_{\boldsymbol{\phi}}(\mathcal{G}_{\boldsymbol{\theta}}(\mathbf{z}))}\right]$$

$$\mathbb{E}_{q_\eta(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] \approx \mathbb{E}_{q_\eta(\mathbf{z}|\mathbf{x})}[-\lambda||\mathbf{x}-\mathcal{G}_{\boldsymbol{\theta}}(\mathbf{z})||_1]$$

Combining VAEs and GANs

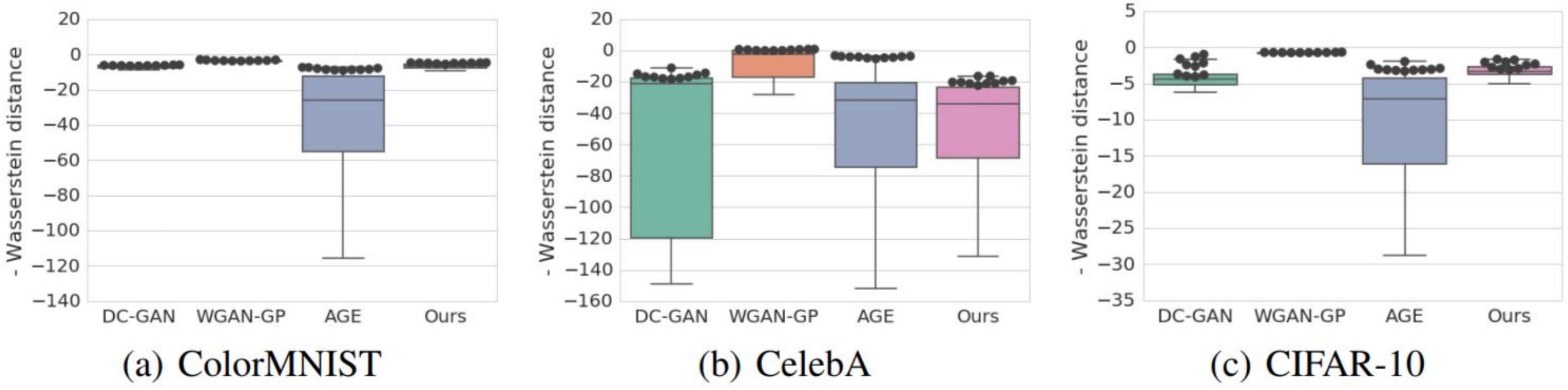
$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \geq \mathbb{E}_{q_\eta(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] - \mathbf{KL}[q_\eta(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]$$

- Likelihood: Reconstruction vs “synthetic likelihood” term
- KL: Analytical vs “code discriminator”
- Can recover various hybrids of VAEs and GANs

Algorithm	Likelihood		Prior		
	Observer	Ratio estimator ("synthetic")	KL (analytic)	KL (approximate)	Ratio estimator
VAE	✓		✓		
DCGAN		✓			
VAE-GAN	✓	*	✓		
Adversarial-VB	✓				✓
AGE	✓			✓	
α-GAN (ours)	✓	✓			✓

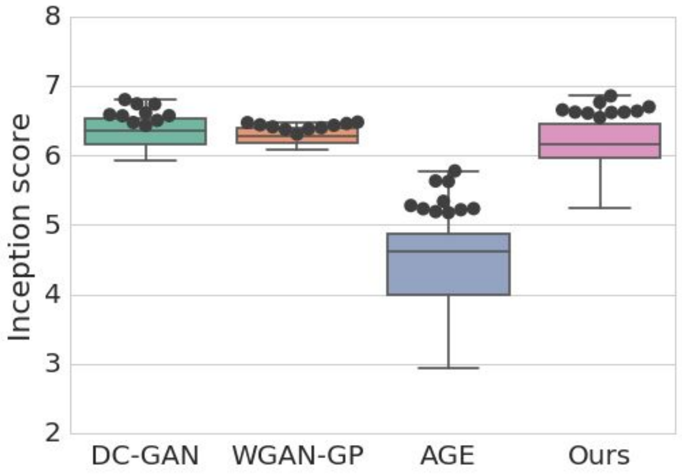
Table 1: Comparison of different approaches for training generative latent variable models.

Evaluating different variants

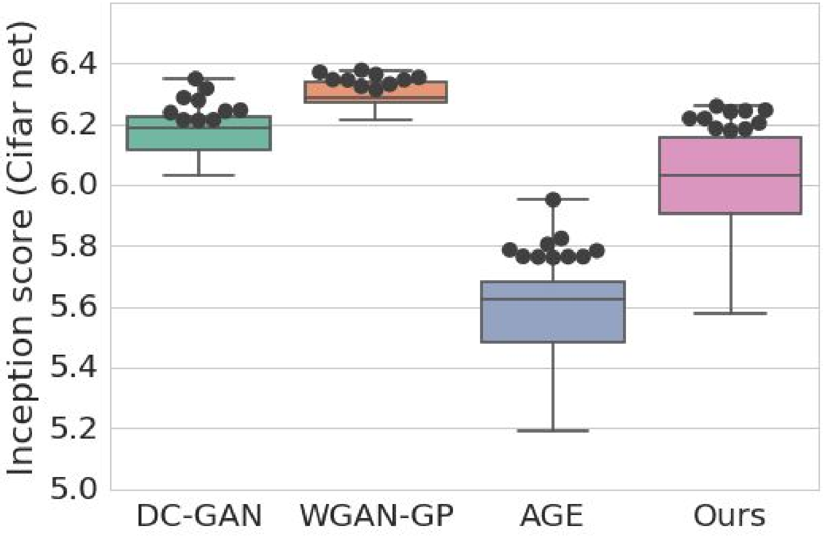


Our VAE-GAN hybrid is competitive with state-of-the-art GANs

Cifar10 - Inception score



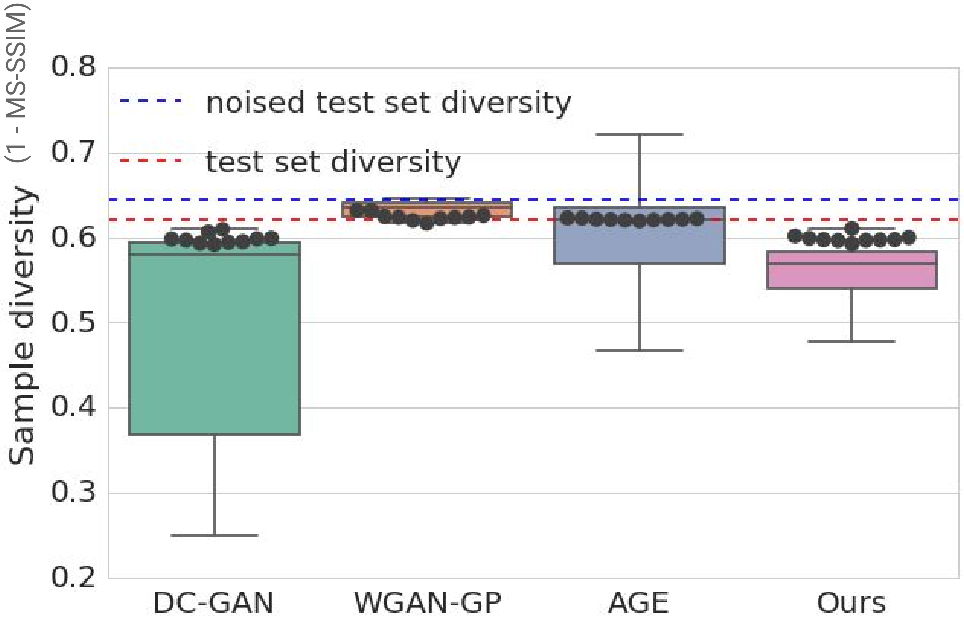
Classifier trained on Imagenet



Classifier trained on Cifar10

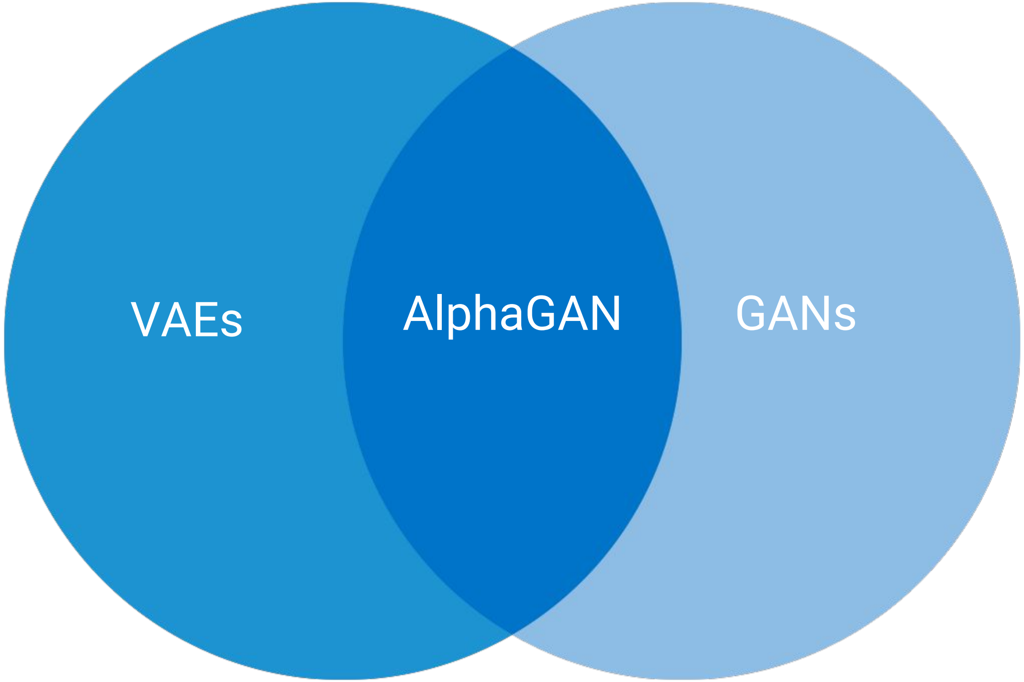
Improved Techniques for Training GANs T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen

CelebA - sample diversity



Summary: VAEs and GANs

- variational inference
  - reconstructions
  - encoder network
- the posterior latents match the prior



- implicit decoder
- Can use implicit encoder: discriminators used to match distributions

Bridging the gap between theory & practice

Many paths to equilibrium: GANs do not need to decrease a divergence at every step

William Fedus, Mihaela Rosca*, Balaji Lakshminarayanan, Andrew Dai, Shakir Mohamed & Ian Goodfellow*

Differences between GAN theory and practice

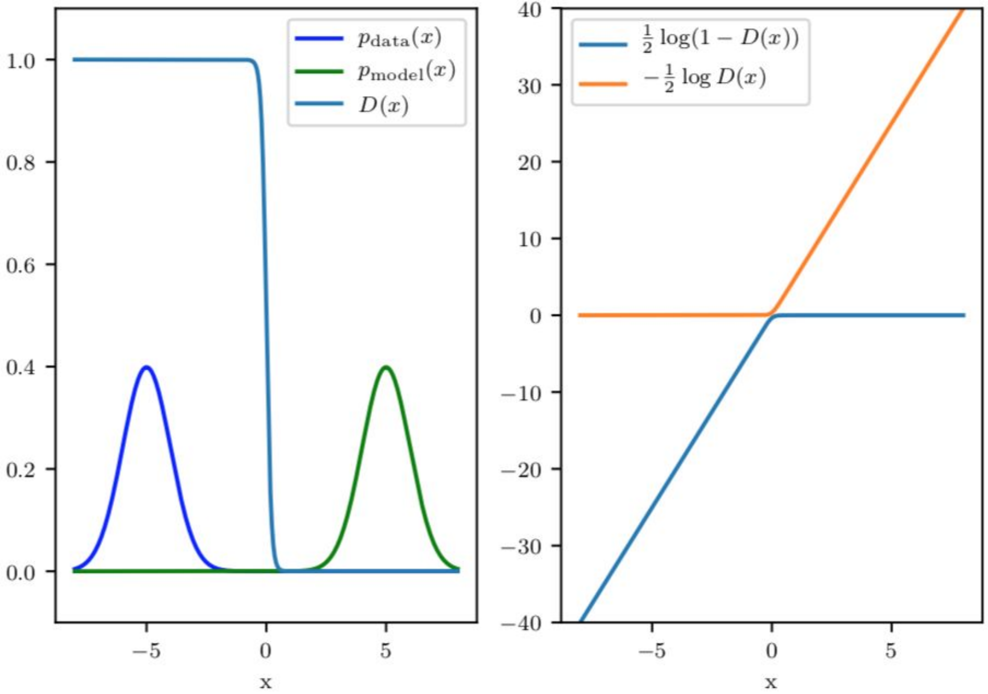
Lots of new GAN variants have been proposed (e.g. Wasserstein GAN)

- Loss functions & regularizers motivated by new theory
- Significant difference between theory and practice

How do we bridge this gap?

- Synthetic datasets where theory predicts failure
- Add new regularizers to original non-saturating GAN

Non-Saturating GAN



Gradient Penalties for Discriminators

$$\tilde{J}^{(D)}(D,G) = -\underset{x\sim p_{\text{data}}}{\mathbb{E}}[\log D(x)] - \underset{z\sim p_z}{\mathbb{E}}[\log(1-D(G(z)))] + \lambda \underset{\hat{x}\sim p_{\hat{x}}}{\mathbb{E}}\left[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2\right]$$

To formalize the above, both proposed gradient penalties of the form:

$$\underset{\hat{x}\sim p_{\hat{x}}}{\mathbb{E}}\left[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2\right],$$

where $p_{\hat{x}}$ is defined as the distribution defined by the sampling process:

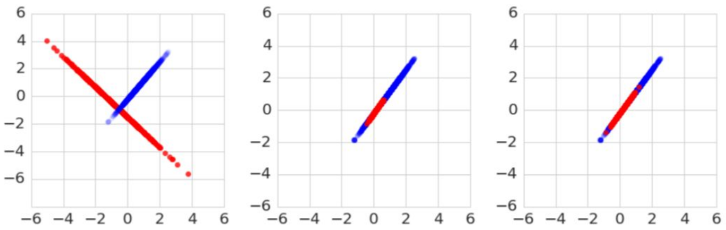
$$x \sim p_{\text{data}}; \qquad x_{\text{model}} \sim p_{\text{model}}; \qquad x_{\text{noise}} \sim p_{\text{noise}}$$

DRAGAN $\tilde{x} = x + x_{\text{noise}}$

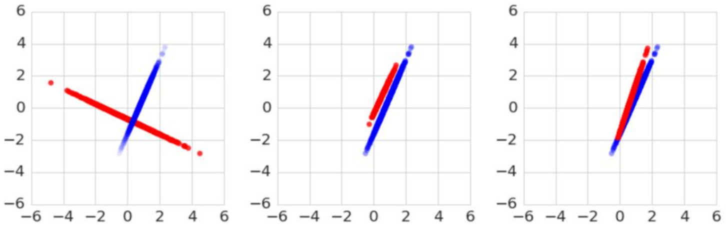
WGAN-GP $\tilde{x} = x_{\text{model}}$

$$\alpha \sim U(0,1)$$

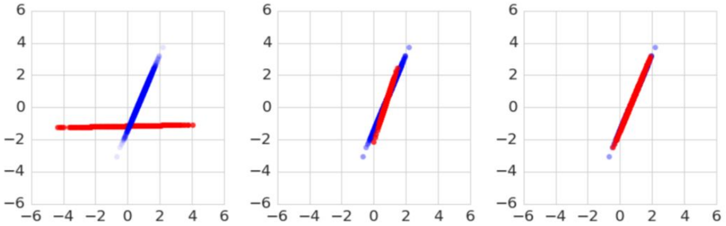
$$\hat{x} = \alpha x + (1-\alpha)\tilde{x}.$$



(a) Non-saturating GAN training at 0, 10000 and 20000 steps.



(b) GAN-GP training at 0, 10000 and 20000 steps.

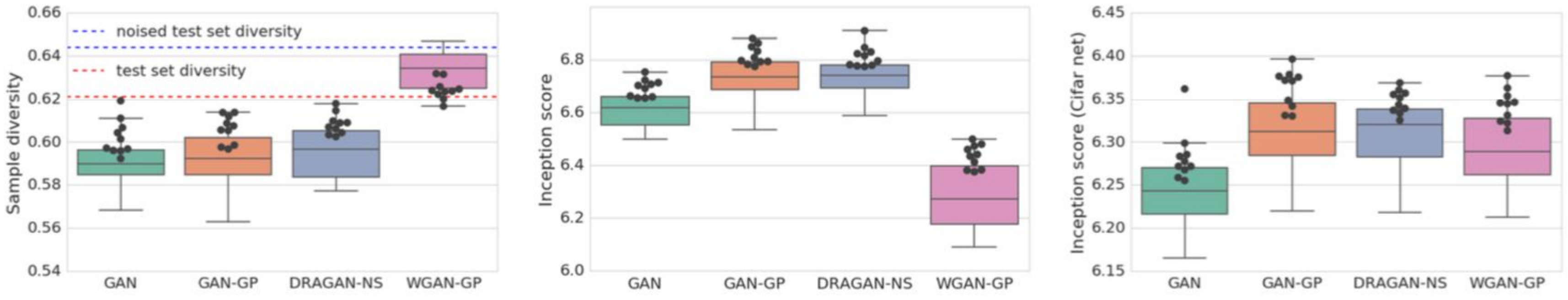


(c) DRAGAN-NS training at 0, 10000 and 20000 steps.

Comparisons on synthetic dataset where Jensen Shannon divergence fails

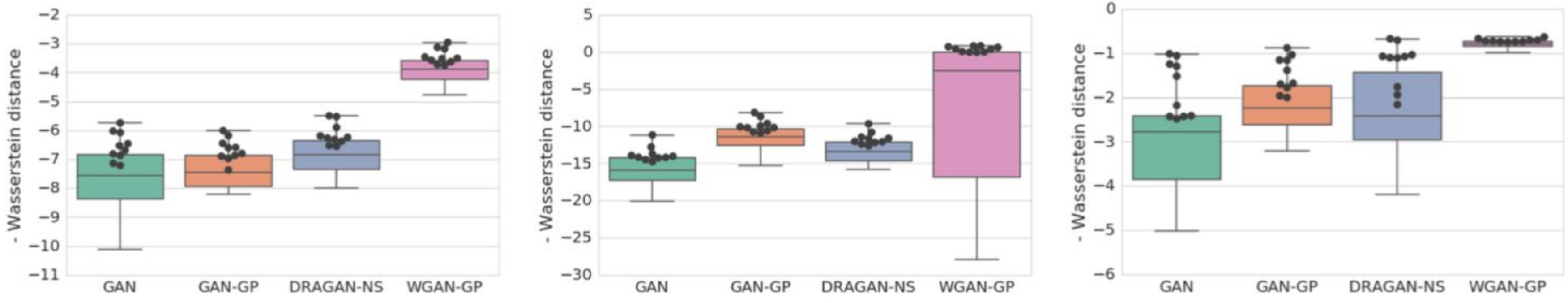
- Gradient penalties lead to better performance

Results on real datasets



(a) CelebA (b) Inception Score (ImageNet) (c) Inception Score (CIFAR)

Results on real datasets



(a) Color MNIST (b) CelebA (c) CIFAR-10

Summary

Some surprising findings:

- Gradient penalties stabilize (non-Wasserstein) GANs as well
- Think not just about the ideal loss function but also the optimization

"In theory, there is no difference between theory and practice. In practice, there is."

- Better ablation experiments will help bridge this gap and move us closer to the holy grail

Other interesting research directions

Overloading GANs and "Adversarial training"

Originally formulated as a minimax game between a discriminator and generator

Recent insights:

- **Density ratio trick**: discriminator estimates a density ratio. Can replace density ratios and f-divergences in message passing with discriminators.

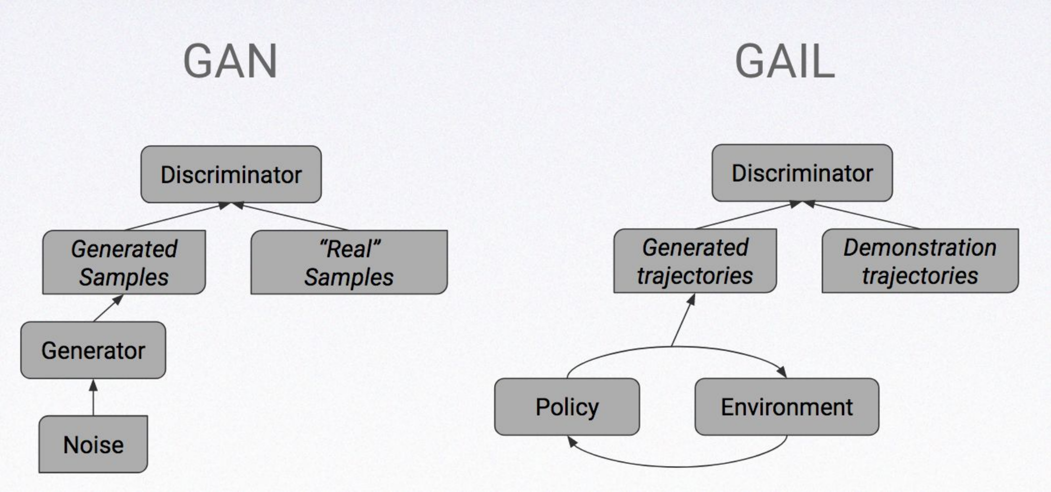
$$r_\phi(\mathbf{x}) = \frac{p^*(\mathbf{x})}{p_\theta(\mathbf{x})} = \frac{p(\mathbf{x}|y=1)}{p(\mathbf{x}|y=0)} = \frac{p(y=1|\mathbf{x})}{p(y=0|\mathbf{x})} = \frac{\mathcal{D}_\phi(\mathbf{x})}{1-\mathcal{D}_\phi(\mathbf{x})}$$

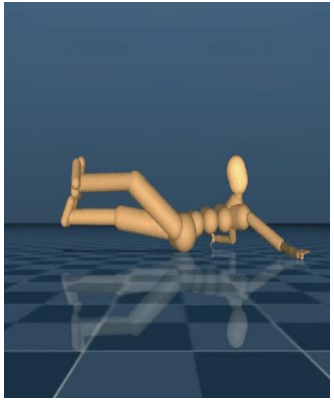
- **Implicit/Adversarial variational inference**: Implicit models can be used for flexible variational inference (require only samples, no need for densities)
- **Adversarial loss**: Discriminator provides a mechanism to "learn" what is realistic, this is better than using a (gaussian) likelihood to train generator.

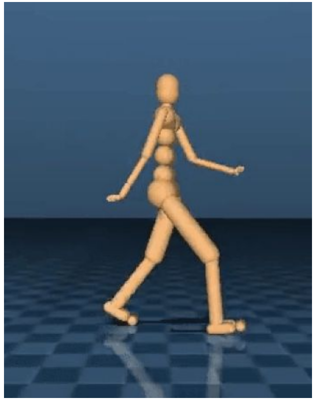
GANs for imitation learning

Use a separate network (discriminator) to "learn" what is realistic

Adversarial imitation learning: RL Reward comes from a discriminator







Learning human behaviors from motion capture by adversarial imitation
Josh Merel, Yuval Tassa, Dhruva TB, Sriram Srinivasan, Jay Lemmon, Ziyu Wang, Greg Wayne, Nicolas Heess

Lots of other exciting research

- Research
  - Using ideas from convergence of Nash equilibria
  - Connections to RL (actor-critic methods)
  - Control theory (e.g. numerics of GANs)

- Applications
  - Class-conditional generation,
  - Text-to-image generation
  - Image-to-image translation
  - Single image super-resolution
  - Domain adaptation

And many more ...

Summary

Ways to stabilize GAN training

- Combine with Auto-encoder
- Gradient penalties

Tools developed in GAN literature are intriguing even if you don’t care about GANs

- Density ratio trick is useful in other areas (e.g. message passing)
- Implicit variational approximations
- Learn a realistic loss function than use a loss of convenience
- How do we handle non-differentiable simulators?
  - Search using differentiable approximations?

Thanks!

Learning in implicit generative models, Shakir Mohamed* and Balaji Lakshminarayanan*

Variational approaches for auto-encoding generative adversarial networks, Mihaela Rosca*, Balaji Lakshminarayanan*, David Warde-Farley and Shakir Mohamed

Comparison of maximum likelihood and GAN-based training of Real NVPs, Ivo Danihelka, Balaji Lakshminarayanan, Benigno Uria, Daan Wierstra and Peter Dayan

Many paths to equilibrium: GANs do not need to decrease a divergence at every step, William Fedus*, Mihaela Rosca*, Balaji Lakshminarayanan, Andrew Dai, Shakir Mohamed and Ian Goodfellow

Slide credits: *Mihaela Rosca, Shakir Mohamed, Ivo Danihelka, David Warde-Farley, Danilo Rezende*

Papers available on my webpage http://www.gatsby.ucl.ac.uk/~balaji/