Stabilizing Training of Generative Adversarial Networks through Regularization

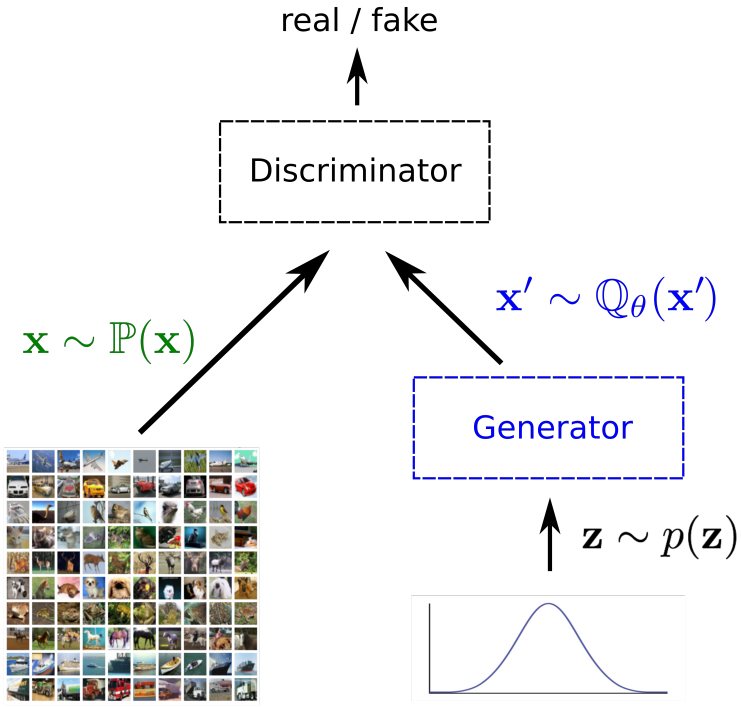
Kevin Roth
Department of Computer Science
ETH Zürich
kevin.roth@inf.ethz.ch

Aurelien Lucchi
Department of Computer Science
ETH Zürich
aurelien.lucchi@inf.ethz.ch

Sebastian Nowozin
Microsoft Research
Cambridge, UK
sebastian.Nowozin@microsoft.com

Thomas Hofmann
Department of Computer Science
ETH Zürich
thomas.hofmann@inf.ethz.ch

GANs



GAN objective [Goodfellow et al.]

$$\min_G \max_D \; \mathbb{E}_{x\sim P}\left[\log D(x)\right] + \mathbb{E}_{z\sim p(z)}\left[\log(1 - D(G(z)))\right]$$

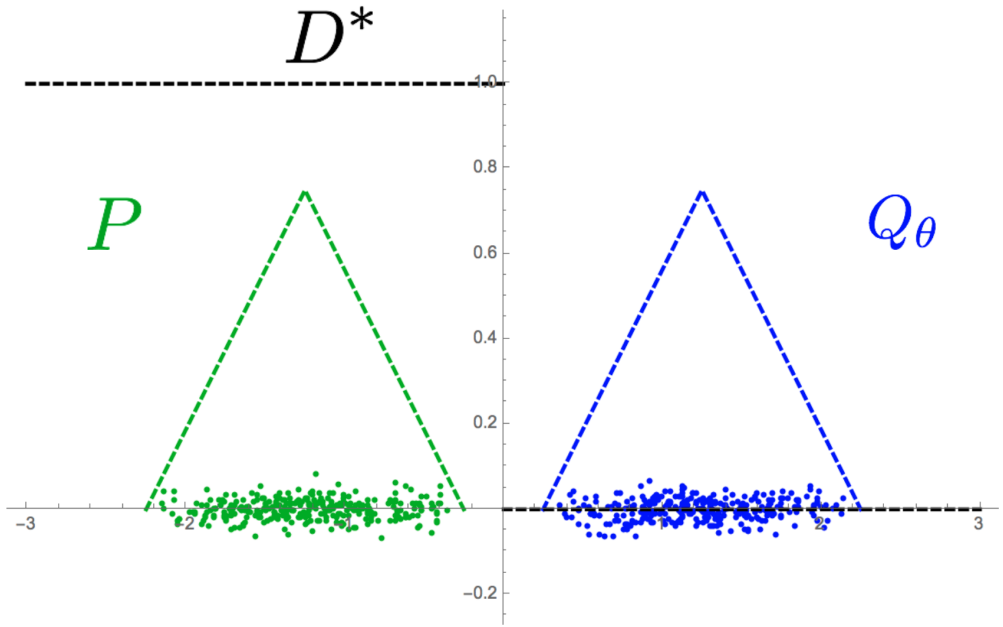
$$= \min_G \; D_{JS}(P\|Q_\theta) \quad \text{for Bayes-optimal } D^*(x)$$

GANs are exciting, but ...

they're also notoriously hard to train!

Problem: *dimensional mismatch* or

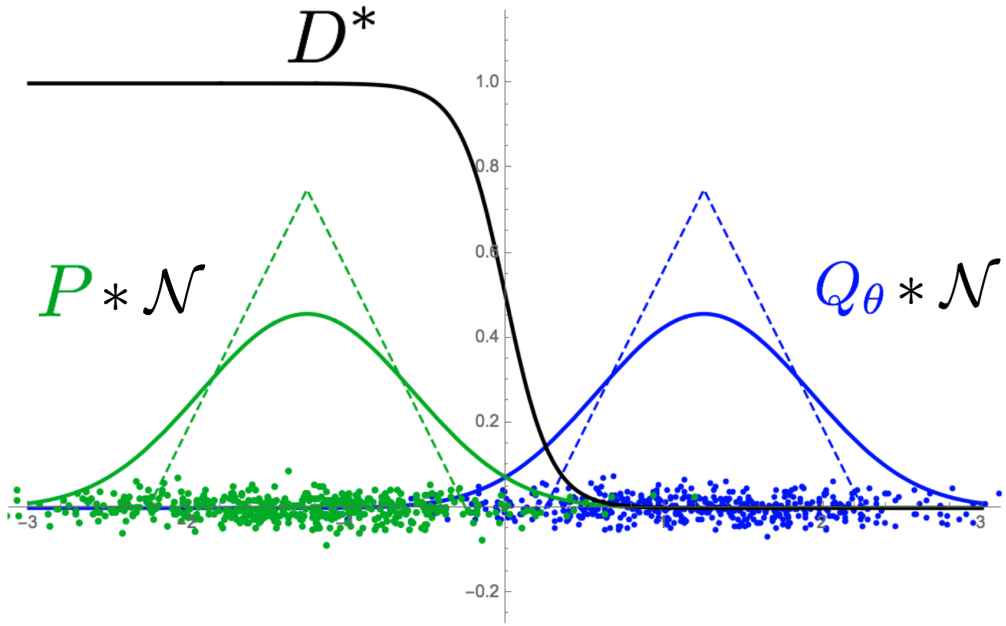
$$\text{supp}(P) \cap \text{supp}(Q_\theta) = \varnothing$$



$\Rightarrow$ undefined f-divergence $D_f(P||Q_\theta) = ???$

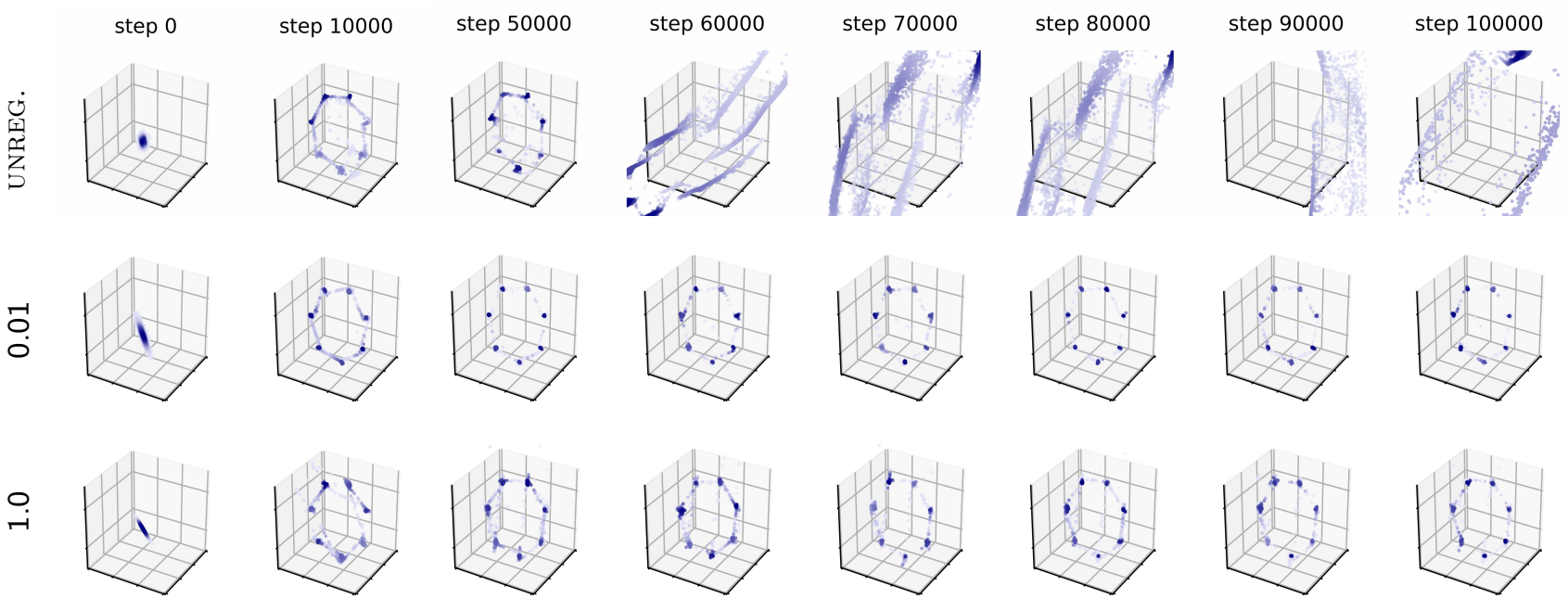
Solution: Adding Noise (Convolving Densities)

$$x \to x + \xi, \quad \xi \sim \mathcal{N}(0, \gamma)$$



=> Regularizing Discriminator

Submanifold Mixture



Gradient-norm regularized GANs for different levels of regularization γ

Background

From f-divergences ...

Measure "similarity" between distributions P and Q (that possess densities p and q, absolutely continuous w.r.t. base measure $\mathrm{d}x$),

$$D_f(P\|Q) := \int_{\mathcal{X}} q(x) f\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x,$$

where $f : \mathbb{R}_+ \to \mathbb{R}$ is convex and satisfies $f(1) = 0$.

Legendre - Fenchel duality:

$$f(u) = \sup_{t \in \mathrm{dom}_{f^c}} \{tu - f^c(t)\}$$

parametrizing f through its tangents or supporting hyperplanes.

... to f-GANs [Nowozin et al.]

$$
\begin{aligned}
D_f(P\|Q) &= \int_{\mathcal{X}} q(x) \sup_{t \in \operatorname{dom} f^c} \left\{ t\frac{p(x)}{q(x)} - f^c(t) \right\} \mathrm{d}x \\
&\geq \sup_{\psi \in \Psi} \left(\int_{\mathcal{X}} p(x)\, \psi(x)\, \mathrm{d}x - \int_{\mathcal{X}} q(x)\, f^c(\psi(x))\, \mathrm{d}x \right) \\
&= \sup_{\psi \in \Psi} \left(\mathbf{E}_{x \sim P} \left[\psi(x) \right] - \mathbf{E}_{x \sim Q_\theta} \left[f^c(\psi(x)) \right] \right),
\end{aligned}
$$

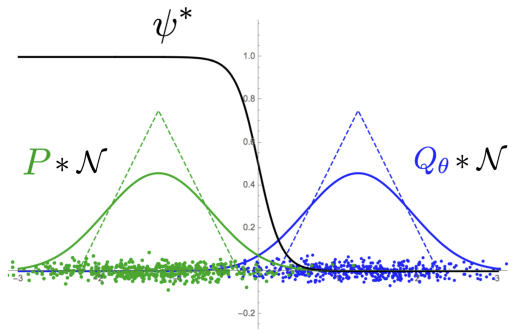
where Ψ is an arbitrary class of discriminants $\psi : \mathcal{X} \to \mathbb{R}$.

Bound is tight for the optimal discriminant [Nguyen et al.]

$$
\psi^*(x) = f' \left(\frac{p(x)}{q(x)} \right)
$$

Regularization

Training with Noise



- Practitioner:

  Explicitly adding noise $\boldsymbol{\xi} \sim \Lambda = \mathcal{N}(0, \gamma\,\mathbb{I})$ to $\mathbf{x} \sim \mathbb{P}, \mathbb{Q}$

- Theory: Convolving Distributions

$$\mathbf{E}_{\mathbb{P}}\mathbf{E}_{\Lambda}[h(\psi(\mathbf{x}+\boldsymbol{\xi}))] = \int h(\psi(\mathbf{x})) \int p(\mathbf{x}-\boldsymbol{\xi})\lambda(\boldsymbol{\xi})d\boldsymbol{\xi}\, d\mathbf{x}$$
$$= \mathbf{E}_{\mathbb{P}*\Lambda}[h \circ \psi]$$

$=>$ Convolved f-GAN Objective

$$F(\mathbb{P} * \Lambda_\gamma, \mathbb{Q} * \Lambda_\gamma; \psi) = \mathbf{E}_{\mathbb{P}*\Lambda_\gamma}\left[\psi\right] - \mathbf{E}_{\mathbb{Q}*\Lambda_\gamma}\left[f^c \circ \psi\right]$$

Analytic Approximation

- For small noise variance γ we can Taylor expand ψ around $\boldsymbol{\xi} = 0$ [Bishop 95]:

$$\psi(\mathbf{x} + \boldsymbol{\xi}) = \psi(\mathbf{x}) + [\nabla \psi(\mathbf{x})]\, \boldsymbol{\xi} + \frac{1}{2}\, \boldsymbol{\xi}^T [\nabla^2 \psi(\mathbf{x})]\, \boldsymbol{\xi} + \mathcal{O}(\boldsymbol{\xi}^3)$$

- Third-order approximation

$$F_\gamma = F_0 + \frac{\gamma}{2} \left\{ \mathbf{E}_{\mathbb{P}} [\triangle \psi] - \mathbf{E}_{\mathbb{Q}} [\triangle (f^c \circ \psi)] \right\} + \mathcal{O}(\gamma^2)$$

- Interpret: Laplace $\triangle = \mathrm{Tr}(\nabla^2)$ measures how much ψ and $f^c \circ \psi$ differ from their local average

Simplification

- Chain-rule:

$$\triangle(f^c \circ \psi) = \ (f^{c\prime\prime} \circ \psi) \cdot ||\nabla\psi||^2 + (f^{c\prime} \circ \psi)\, \triangle\psi$$

- Property of optimal discriminant ψ^*

$$(f^{c\prime} \circ \psi^*)\, d\mathbb{Q} = d\mathbb{P}$$

$=>$ Convenient cancellation at $\psi = \psi^* + \mathcal{O}(\gamma)$ [Bishop 95]:

$$\mathbf{E}_{\mathbb{P}}\left[\triangle\psi^*\right] - \mathbf{E}_{\mathbb{Q}}\left[\triangle(f^c \circ \psi^*)\right] = -\mathbf{E}_{\mathbb{Q}}[(f^{c\prime\prime} \circ \psi^*) \cdot ||\nabla\psi^*||^2]$$

Regularized f-GAN

$$F_\gamma(\mathbb{P}, \mathbb{Q}_\theta; \psi) = \mathbf{E}_{\mathbb{P}}[\psi] - \mathbf{E}_{\mathbb{Q}_\theta}[f^c \circ \psi] - \frac{\gamma}{2}\Omega_f(\mathbb{Q}_\theta; \psi)$$

$$\Omega_f(\mathbb{Q}_\theta; \psi) := \mathbf{E}_{\mathbb{Q}_\theta}\left[(f^{c\prime\prime} \circ \psi) \cdot \|\nabla\psi\|^2\right]$$

- "soft Lipschitz" constraint
- non-negative weighting function $f^{c\prime\prime} \geq 0$
- lower variance compared to explicitly adding noise
- easy to implement
- can train indefinitely without collapse!

Algorithm 1 Regularized f-GAN (with annealing). Default values $\gamma_0 = 2.0$, $\alpha = 0.005$, $n_\psi = 1$

Require: Initial noise variance γ_0, annealing decay rate α, number of discriminator update steps n_ψ per generator iteration, minibatch size m, number of training iterations T

Require: Initial discriminator parameters ω_0, initial generator parameters θ_0

for $t = 1, ..., T$ **do**

$\gamma \leftarrow \gamma_0 \cdot \alpha^{t/T}$ # annealing

for $1, ..., n_\psi$ **do**

Sample minibatch of real data $\{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(m)}\} \sim \mathbb{P}$.

Sample minibatch of latent variables from prior $\{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(m)}\} \sim p(\mathbf{z})$.

$$F(\omega, \theta) = \frac{1}{m} \sum_{i=1}^{m} \left[\psi_\omega(\mathbf{x}^{(i)}) - f^c\Big(\psi_\omega(G_\theta(\mathbf{z}^{(i)}))\Big) \right]$$

$$\Omega_f(\omega, \theta) = \frac{1}{m} \sum_{i=1}^{m} f^{c\prime\prime}\Big(\psi_\omega(G_\theta(\mathbf{z}^{(i)}))\Big) \left|\left| \nabla_{\tilde{\mathbf{x}}} \psi_\omega(\tilde{\mathbf{x}})\big|_{\tilde{\mathbf{x}}=G_\theta(\mathbf{z}^{(i)})} \right|\right|^2$$

$$\omega \leftarrow \omega + \nabla_\omega \Big(F(\omega, \theta) - \frac{\gamma}{2} \Omega_f(\omega, \theta) \Big)$$ # gradient ascent

end for

Sample minibatch of latent variables from prior $\{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(m)}\} \sim p(\mathbf{z})$.

$$F(\omega, \theta) = \frac{1}{m} \sum_{i=1}^{m} -f^c \left(\psi_\omega(G_\theta(\mathbf{z}^{(i)})) \right)$$

$\theta \leftarrow \theta - \nabla_\theta F(\omega, \theta)$ # gradient descent

end for

The gradient-based updates can be performed with any gradient-based learning rule. We used Adam in our experiments.

tf code → github.com/rothk

```
# ------------------------------------------------------------------------------
#     JS-Regularizer
# ------------------------------------------------------------------------------

  def Discriminator_Regularizer(self, D1, D1_logits, D1_arg, D2, D2_logits, D2_arg):
      grad_D1_logits = tf.gradients(D1_logits, D1_arg)[0]
      grad_D2_logits = tf.gradients(D2_logits, D2_arg)[0]
      grad_D1_logits_norm = tf.norm(tf.reshape(grad_D1_logits, [self.batch_size,-1]), axis=1, keep_dims=True)
      grad_D2_logits_norm = tf.norm(tf.reshape(grad_D2_logits, [self.batch_size,-1]), axis=1, keep_dims=True)
      reg_D1 = tf.multiply(tf.square(1.0-D1), tf.square(grad_D1_logits_norm))
      reg_D2 = tf.multiply(tf.square(D2), tf.square(grad_D2_logits_norm))
      return tf.reduce_mean(reg_D1 + reg_D2)
```

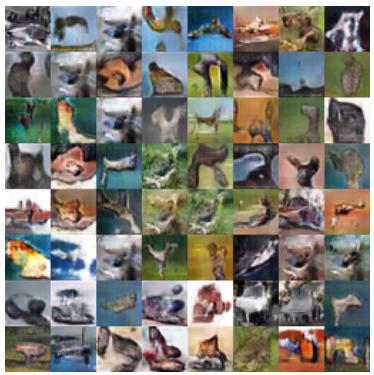
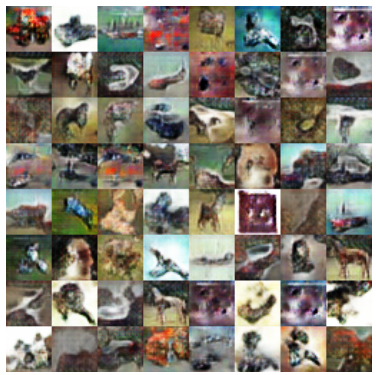
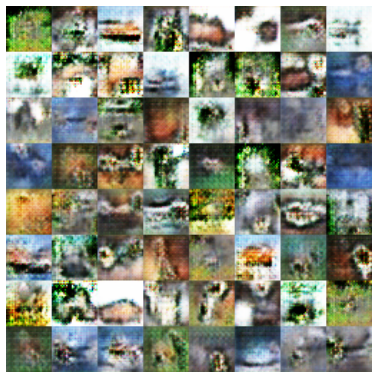
Experimental Results

Regularization vs. Explicitly Adding Noise

UNREGULARIZED

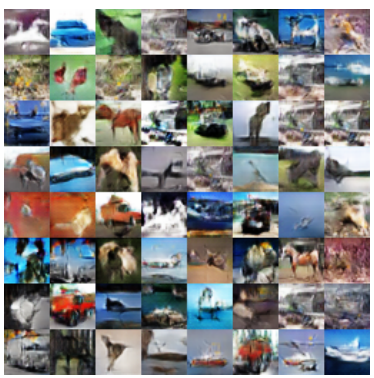
EXPLICIT NOISE

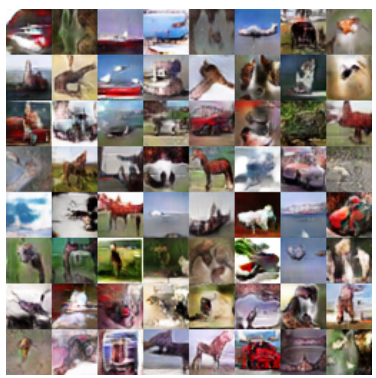
0.01 0.1 1.0

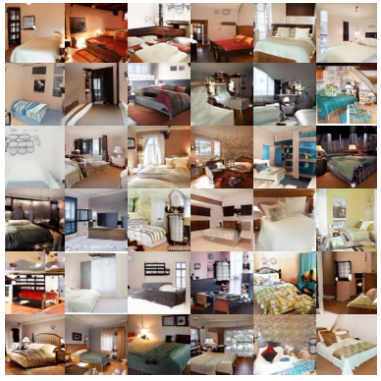
REGULARIZED

0.001 0.01 0.1 1.0

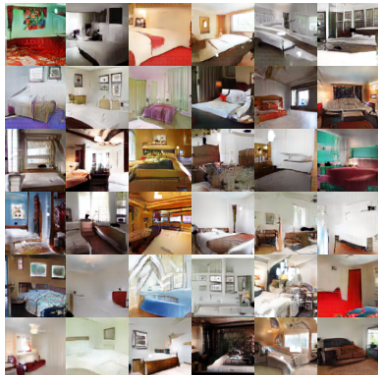


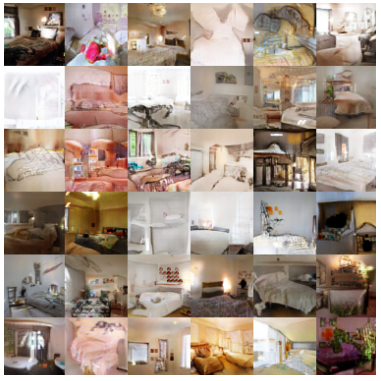
Stability across various architectures



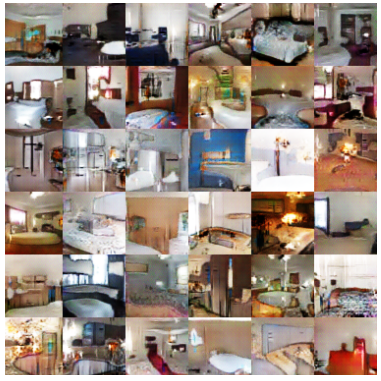
ResNet



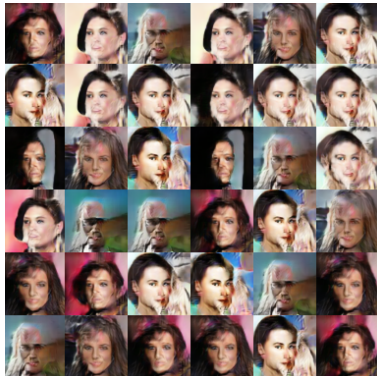
DCGAN



No Normalization

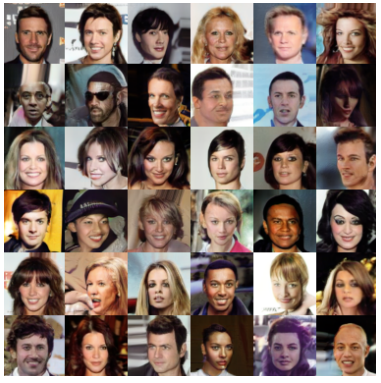


TANH



UNREG.

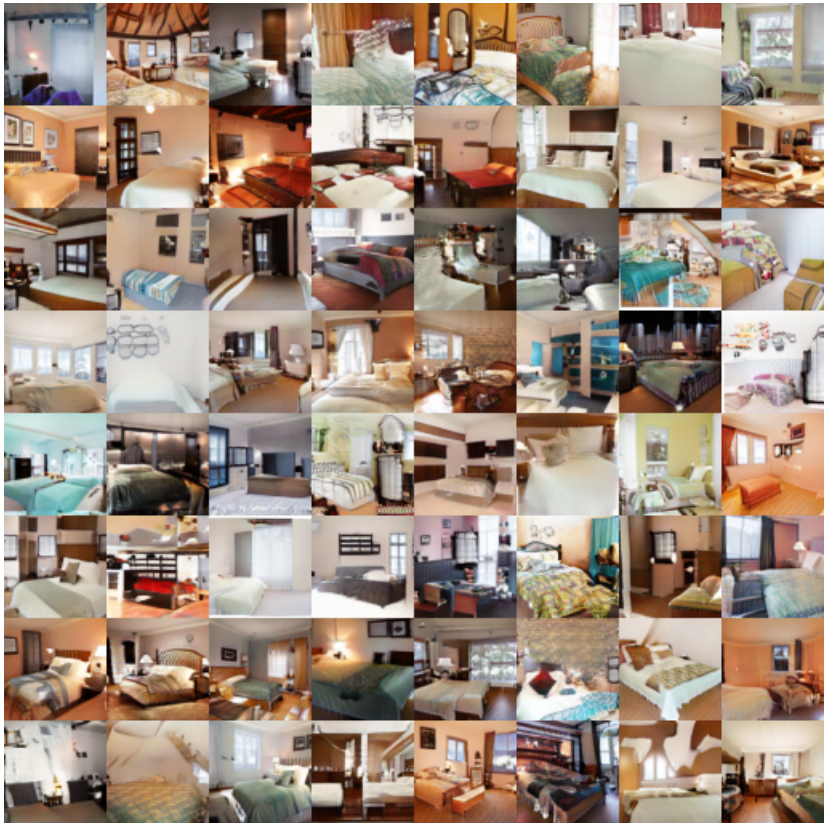
0.5 -



ANNEALING

- 2.0

Sample quality and diversity



Cross-Testing Protocol

Regularized $\gamma = 0.1$

Pred. \ True Cond.	Pos.	Neg.
Pos.	0.9688	0.0002
Neg.	0.0312	0.9998

Cross-testing: FP: 0.0

Unregularized

Pred. \ True Cond.	Pos.	Neg.
Pos.	1.0	0.0013
Neg.	0.0	0.9987

Cross-testing: FP: 1.0

- Regularized D classifies unregularized G's samples as fake
- Unregularized D classifies samples of regularized G as real

$\Rightarrow$ Regularized GAN generalizes better!

Outlook

- Applications: What to actually do with GANs?
- Evaluation: Convergence, generalization, mode collapse, sample quality, latent space semantics, WGAN vs. f-GAN, ...
- Optimization: Saddle-points in minimax problems, non-conservative vector-fields, ...

References

- Roth K., Lucchi A., Nowozin S., Hofmann T., Stabilizing Training of Generative Adversarial Networks through Regularization, NIPS '17
- Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y., Generative Adversarial Networks, NIPS '14
- Nowozin S., Cseke B., Tomioka R., f-GAN: Training generative neural samplers using variational divergence minimization, NIPS '16
- Nguyen X., Wainwright M.J., Jordan M.I. Estimating divergence functionals and the likelihood ratio by convex risk minimization, IEEE Transactions on Information Theory, 2010
- Bishop C.M., Training with noise is equivalent to Tikhonov regularization, Neural computation, 1995
- An G., The Effects of Adding Noise During Backpropagation Training on Generalization Performance, Neural computation, 1996