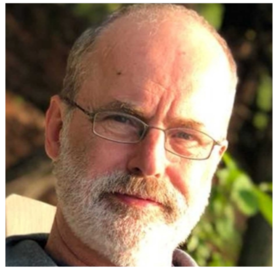
Samsung Research

GraN-GAN: Piecewise Gradient Normalization for Generative Adversarial Networks






Vin Bhaskara*[1] Tristan A. A.*[1,2,3] Allan Jepson[1] Alex Levinshtein[1]

* Equal Contribution

WACV 2022

[1] Samsung AI Center-Toronto

[2] Computer Science, UNIVERSITY OF TORONTO

[3] VECTOR INSTITUTE

Introduction

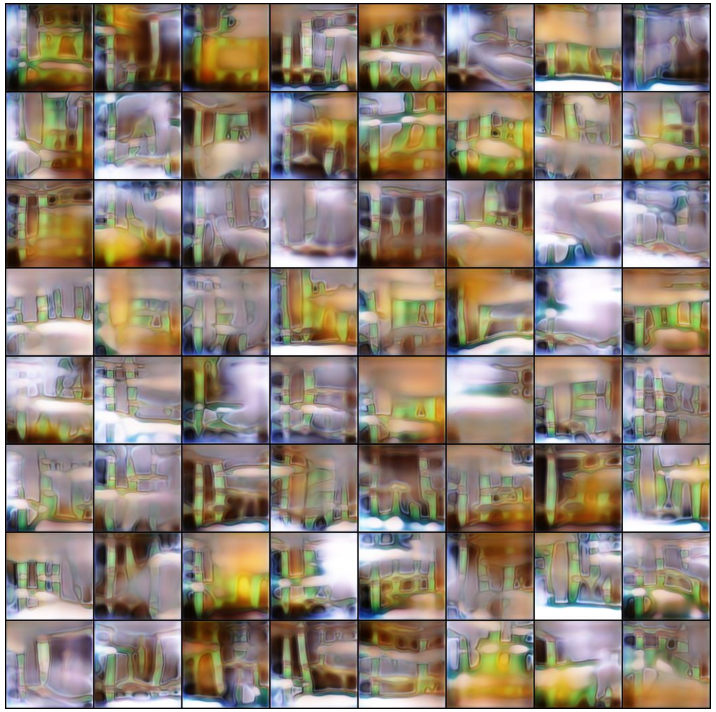
- Generative Adversarial Networks (GANs) are quite effective in unsupervised image generation.

Examples of generated images taken from StyleGAN [1] and BigGAN [2]:

[1] https://arxiv.org/pdf/1812.04948.pdf
[2] https://arxiv.org/pdf/1809.11096.pdf

Introduction

- Despite their effectiveness, GANs are hard to train.



Introduction

- The generated images and the training dynamics of the generator network $G(z)$ are dependent on the gradients of the loss function L_G:

$$\nabla_x L_G(D(x)) = \underbrace{\nabla_D L_G(D(x))}\ \underbrace{\nabla_x D(x)}$$

Loss function gradient:
defined by the GAN objective
(choice of distributional divergence)

Examples:
- Cross-entropy (CE) loss
- Non-saturating (NS) CE loss
- Wasserstein/Hinge losses

"Input-gradient" of Discriminator:
function of the network
architecture and constraints

Examples:
- Architectures (ResNet, DCGAN)
- Constraints on the Lipschitz constant and/or the input gradients $\nabla_x D(x)$:
  - Gradient Penalty
  - Spectral Norm
  - **GraN (Ours)**

Our focus in this work

Constraints to stabilize the gradients sent to G in training

Lipschitz Constant / Gradient Regularizations

Gradient Penalties (GPs)

- Soft-constrain the norm of the input-gradient of discriminator/critic
- Wasserstein-GANs enforce a Lipschitz Constant (LC) = 1. The P_1 penalty imposes a two-sided constraint on the grad norm
- R_1 Zero-centered gradient penalty:

$$\mathrm{P}_1(\mathrm{x}) = (||\nabla_x D(x)||_2 - 1)^2$$

$$\mathrm{R}_1(\mathrm{x}) = ||\nabla_x D(x)||_2^2$$

Spectral Normalization

- Per-layer 1-Lipschitz constraint on the discriminator/critic using an estimate of the largest singular value $\sigma(W_i)$ of weight matrix W_i:

$$W_i \leftarrow W_i / \sigma(W_i)$$

Downsides

- GPs do not guarantee exact enforcement and their domain must shift to catch-up to G in training.
- SN enforces layer-wise 1-Lipschitzness but can cause gradient attenuation due to progressively shrinking (smallest) LC with depth.

$$|D|_{\mathrm{Lip}} := \sup_{x_1, x_2} \frac{|D(x_2) - D(x_1)|}{|x_2 - x_1|}$$

$$|W_1 W_2 \cdot x|_{\mathrm{Lip}} \leq |W_1 \cdot x|_{\mathrm{Lip}} \; |W_2 \cdot x|_{\mathrm{Lip}}$$

Deep Piecewise Linear Networks (PLNs)

- Many modern deep neural networks $\widetilde{D}(x)$ with piecewise linear activation functions are piecewise linear networks (PLNs) in inputs x
- PLNs divide the input space into a set of convex polytopes
- Within each such segment, the network function is linear

$$\widetilde{D}(x) = w_x(\theta) \cdot x + b_x(\theta)$$

where $w_x(\theta)$ and $b_x(\theta)$ are the *effective* weights and biases of the overall linear function. (Note: $w_x(\theta)$ is *constant* in x, within a polytope.)

- The gradient therefore has a simple expression, per segment:

$$\nabla_x \widetilde{D}(x) = w_x(\theta)$$

Our method: Gradient Normalization (GraN)

Samsung Research

- When the discriminator/critic is a ReLU network, we can guarantee bounded gradients and piecewise K-Lipschitzness by defining the *normalized discriminator/critic* $D(x)$ as:

Discriminator output before normalization

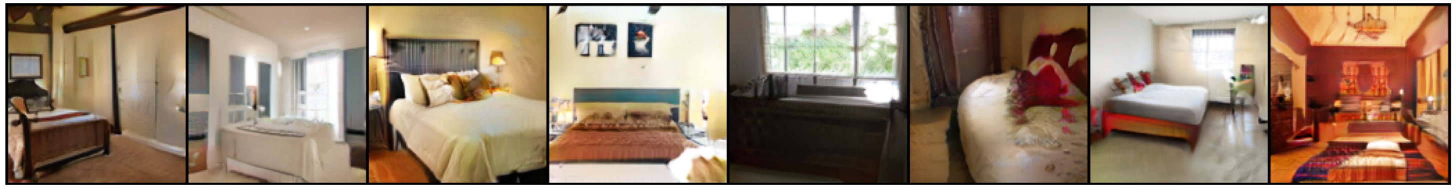
Normalizer $\approx \dfrac{K}{||\nabla_x \widetilde{D}(x)||}$

$$D(x) = \widetilde{D}(x)\ \frac{K\ ||\nabla_x \widetilde{D}(x)||}{||\nabla_x \widetilde{D}(x)||^2 + \epsilon}$$

- This guarantees a local K-Lipschitz constraint and bounds the gradient norm almost everywhere in x since

$$||\nabla_x D(x)|| = \frac{K\ ||\nabla_x \widetilde{D}(x)||^2}{||\nabla_x \widetilde{D}(x)||^2 + \epsilon} < K$$

Results on Unconditional Image Generation



GraND-GAN (FID: 10.8)

WGAN-GP (FID: 13.6)

SNGAN (FID: 13.2)

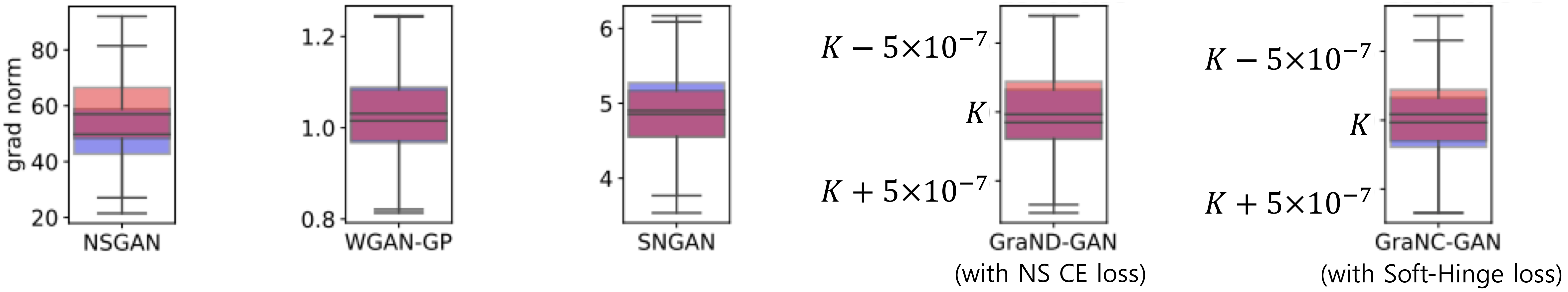
Method	FID ↓ LSUN	FID ↓ CelebA
NSGAN	–	–
NSGAN-GP	–	–
NSGAN-SN	74.926	14.33
NSGAN-GP†	**10.483**	**9.385**
NSGAN-SN†	12.635	9.644
GraND-GAN (Ours)	**10.795**	**9.377**
WGAN-GP	13.562	–
SNGAN	**13.237**	**13.466**
WGAN-GP†	16.884	–
SNGAN†	67.346	15.874
GraNC-GAN (Ours)	**12.533**	**12.000**

Best, **Second best** in FID

- **Please refer to the paper for more details and results on other datasets and metrics**

GraN-GAN: Empirical Analysis of LC

- Boxplots of **gradient norms** across real (blue) and fake (red) samples at 50K iterations (out of 100K) on CIFAR-10 with $K = 0.83$:



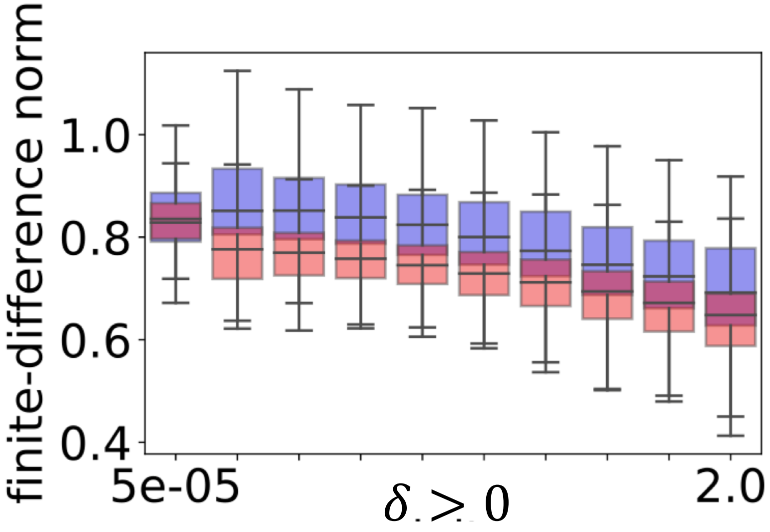
$\sim 10^{+1}$ $\sim 10^{-1}$ $\sim 10^{0}$ Ours: $\sim 10^{-7}$ ($K = 0.83$)

Order of magnitude of StdDev($||\nabla_x D(x)||$) across samples x

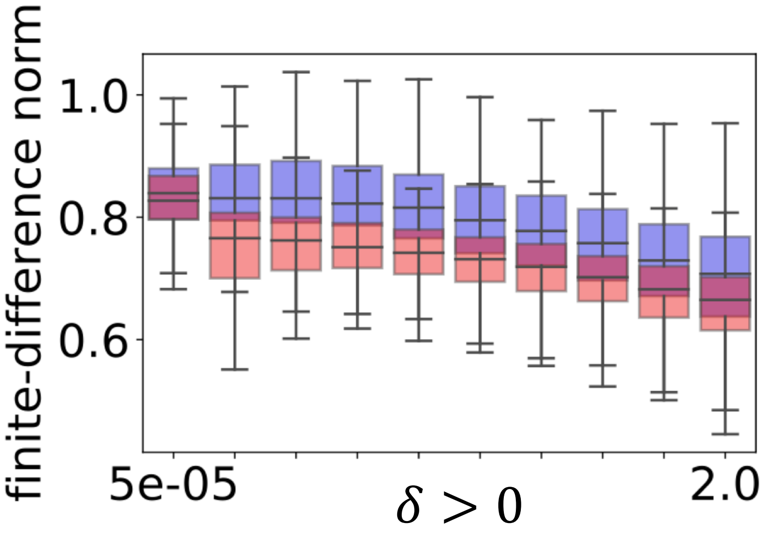
GraN-GAN: Empirical Analysis of LC

- GraN enforces a bounded gradient norm and, thus, a local K-Lipschitz constraint almost everywhere.
- However, due to the presence of discontinuities in the normalized discriminator at the polytope edges, GraN does not guarantee a global Lipschitz constraint
- Nevertheless, empirically the finite-difference grad norms are well-behaved even for large steps δ along $\hat{n} = \nabla_x D(x)$ on CIFAR-10 with $K = 0.83$:

$$\frac{|D(x+\delta\hat{n}) - D(x)|}{\delta}$$



GraND-GAN
(with NS CE loss)



GraNC-GAN
(with Soft-Hinge loss)

Conclusion

- We introduced GraN for piecewise linear discriminators/critics:
  - Ensures bounded input-gradients
  - Guarantees a tight local K-Lipschitz constraint almost everywhere
  - Does not constrain individual layers
- GraN results in improved GAN performance across datasets and loss types
- Despite discontinuities in D, we empirically observe a bounded global Lipschitz constant

Thank you for your attention!