Samsung AI Center-Toronto

Computer Science
UNIVERSITY OF TORONTO

GraN-GAN: Piecewise Gradient Normalization for Generative Adversarial Networks

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

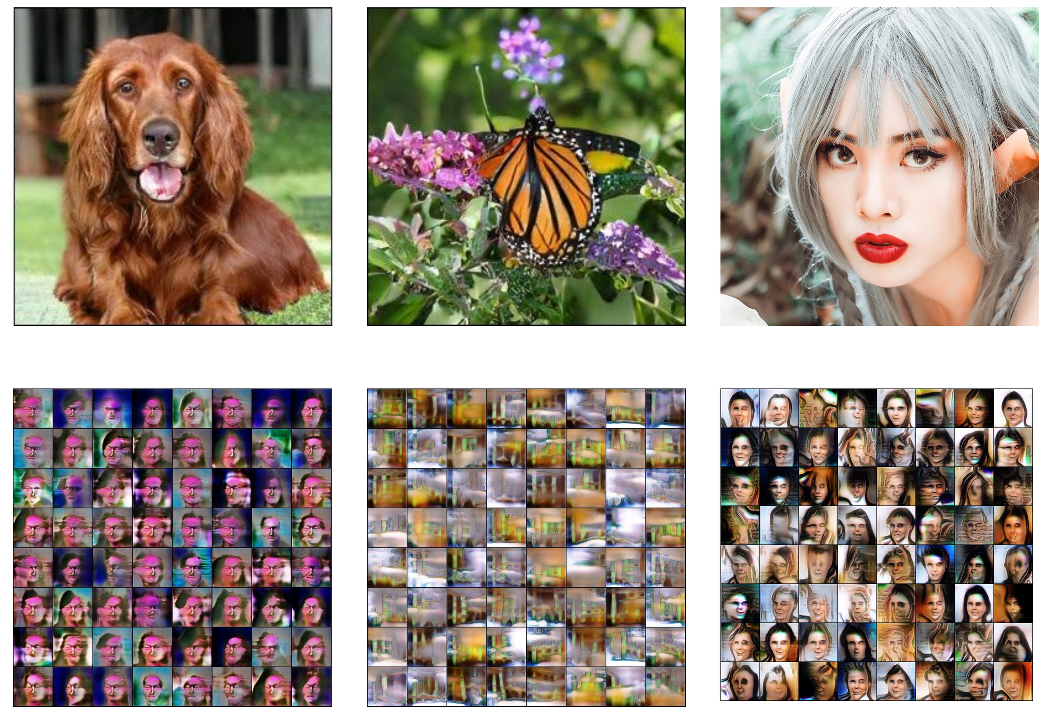
[*] Equal contribution, [1] SAIC-Toronto, [2] University of Toronto, [3] Vector Institute

VECTOR INSTITUTE

WACV WAIKOLOA, HI JAN 4-8

Introduction

(Images taken from StyleGAN and BigGAN)



- Generative Adversarial Networks (GANs) can be quite effective in unsupervised image generation.
- Despite their effectiveness, GANs are hard to train (e.g., mode collapse, training divergence).
- Recall: we wish to learn generator G, such that $x = G(z)$ looks like a *real* data sample $x \sim P_{data}$

E.g., with the classical minimax objective (Jensen-Shannon divergence):

$$\min_G \max_D E_{x\sim P_{data}}[\log D(x)] + E_{z\sim P_z}[\log(1 - D(G(z)))]$$

- 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)

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

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

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

Constraints to stabilize the gradients sent to G in training, ensure D does not get too far ahead of G by regularization, enforce theoretical properties (e.g., WGAN)

Our focus in this work

Previous works

- 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:

$$P_1(x) = (||\nabla_x D(x)||_2 - 1)^2$$

  - Zero-centered gradient penalty:

$$R_1(x) = ||\nabla_x D(x)||_2^2$$

- Spectral Normalization (SN):
  - 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:

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

- Drawbacks:
  - 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 (best) LC with depth.

Our method: GraN or Gradient Normalization

- 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 $\approx \frac{K}{||\nabla_x \tilde{D}(x)||}$

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

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

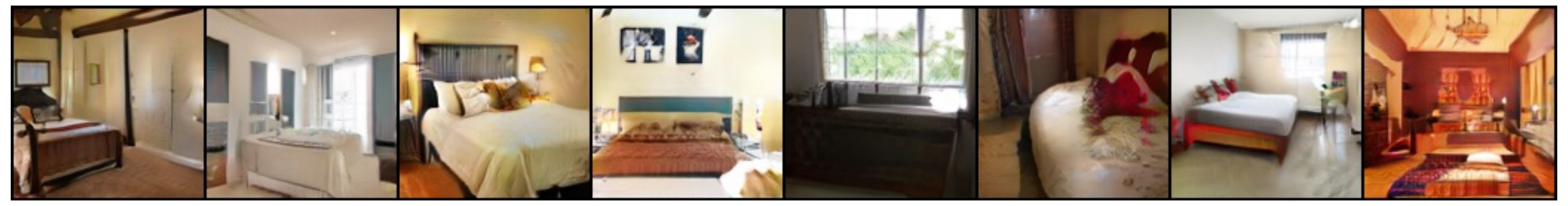
$$||\nabla_x D(x)|| = \frac{K\,||\nabla_x \tilde{D}(x)||^2}{||\nabla_x \tilde{D}(x)||^2 + \epsilon} < K$$

Results on Unconditional Image Generation

- We evaluate image generation across multiple datasets (CIFAR-10, CIFAR-100, STL-10, Celeb-A, LSUN Bedrooms) and GAN losses. Check our paper for more detailed results.

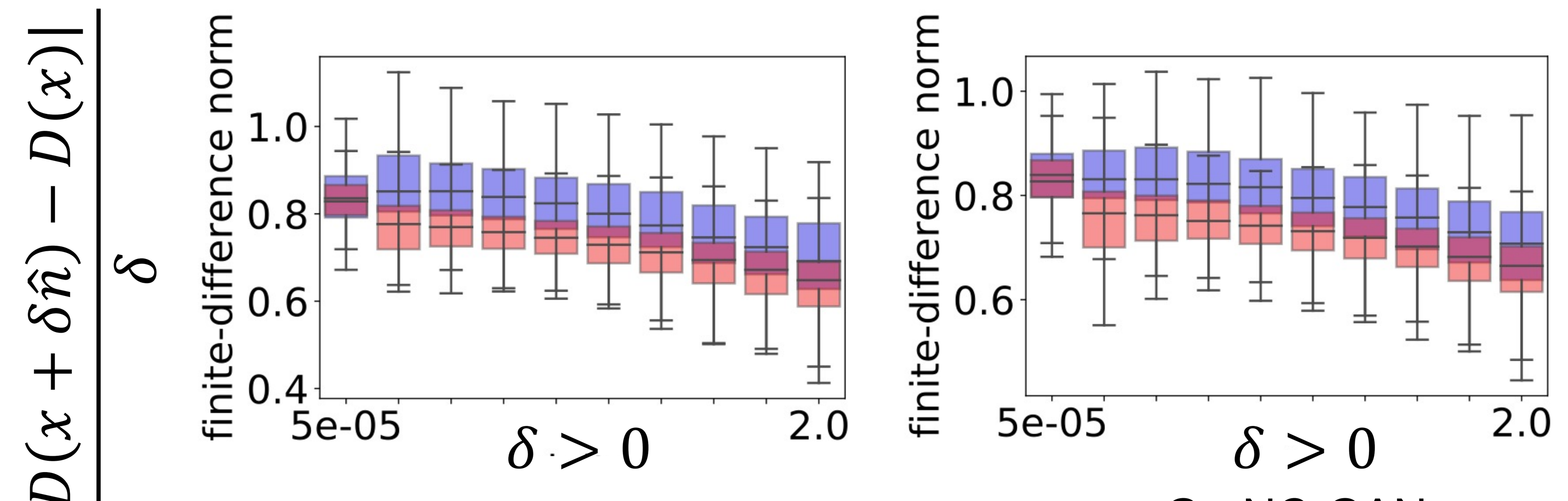
WGAN-GP (FID: 13.6)

SNGAN (FID: 13.2)



GraND-GAN (FID: 10.8)

- Despite our method enforcing local K-Lipschitzness in theory, 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 $K = 0.83$:



GraND-GAN (with NS CE loss)

GraNC-GAN (with Soft-Hinge loss)

Summary

- We introduced GraN for piecewise linear discriminators/critics, which ensures bounded input-gradients and guarantees a tight local K-Lipschitz constraint almost everywhere, yet does not constrain individual layers. GraN results in improved GAN performance across datasets and loss types.