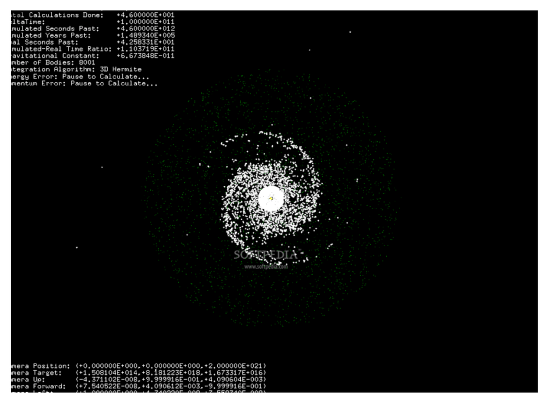
# Improved Weak Gravitational Lensing Using Generative Adversarial Networks

Amay Aggarwal, Andrew Ying, Michel Dellepere {amayagg, androo, mdelle1}@stanford.edu
*Department of Computer Science, Department of Physics, Stanford University*

## Background

### Gravitational Lensing

- *Gravitational lensing* occurs when massive objects create gravitational fields that distort photon trajectories in space.
- Weak Gravitational Lensing (WGL) is a tool that acts as a tracer for dark matter and allows physicists to make predictions about cosmic acceleration.



### Problem

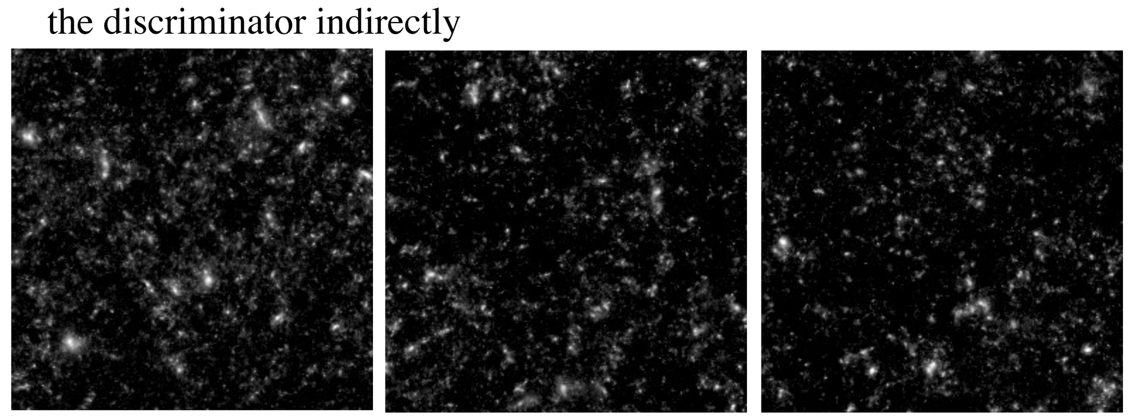
- Cosmological models known as N-body simulators are resource intensive as they track the movements of billions of particles through space.

### Our Approach and Objectives

- Generative Adversarial Networks have achieved state of the art performance in cosmological modeling.
- Investigate ways to prevent mode collapse
- **Goal**: To produce samples resembling baseline images with higher model stability.

## Data

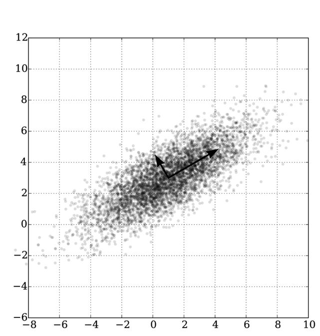
- 16,000 images (256 x 256) generated using Gadget2 [1] N-Body simulation code to create mock WL convergence maps.
- Sample validation from all maps because GAN's generator networks are not directly trained on the dataset but instead learn about it from the discriminator indirectly



Random sampled training data instances

## Metrics

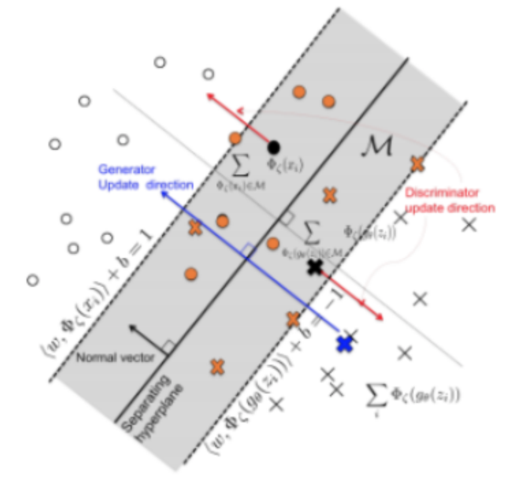
1. **Pixel Intensity:** traditional first-order statistic used in cosmological imaging
2. **Power Spectrum:** measure of the correlation in gravitational lensing at different distances
3. **Convergence using Principal Components:** assess convergence quality by plotting generated images along the top two principal components and examining the shape compared to the validation set.



## Framework / Methods

### Method 1: Geometric GAN

- GeoGAN uses an SVM separating hyperplane, composed of the following operations in feature space
1. Finding a separating hyperplane for a linear classifier
2. Discriminator parameter update away from hyperplane using gradient descent.
3. Generator parameter update along direction normal to hyperplane.
- Mathematically proven to converge to a Nash equilibrium as training examples $n \to \infty$



#### Discriminator Loss

$$\frac{1}{2Cn}\sum_{i=1}^{n}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1 - \langle w, 1 - \Phi_\zeta(x^{(i)})\rangle - b\right) +$$
$$\frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1 - \langle w, 1 + \Phi_\zeta(g_\theta(z^{(i)}))\rangle + b\right)$$

#### Generator Loss

$$L_{w,b,\zeta}(\theta) \;:=\; -\frac{1}{n}\sum_{i=1}^{n} D_{w,b,\zeta}(g_\theta(z_i))$$

#### Optimization Problem

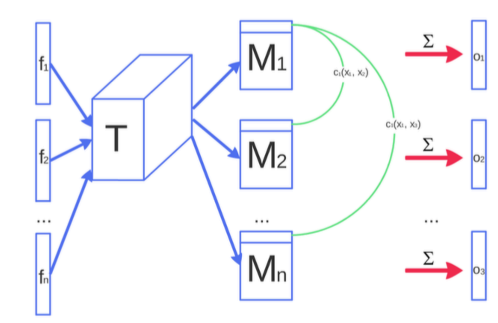
$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi i')$$

$$\langle w, \Phi(x^{(i)})\rangle + b \geq 1 - \xi_i, i = 1, ..., n$$
$$\langle w, \Phi(x^{(i)})\rangle + b \leq \xi_i' - 1, i = 1, ..., n$$
$$\xi_i, \xi_i' \geq 0, i = 1, ..., n$$

### Method 2: Mini-batch discrimination

- Technique to reduce mode collapse. Incorporates the *closeness* of an image with the others in its batch and feeds this result into next layer of discriminator.

$$c_b(x_i, x_j) = \exp(-\|M_{i,b} - M_{j,b}\|_1)$$



### Method 3: Relativistic Discriminator

- Modification: the discriminator now estimates the probability that the real data is more realistic than randomly sampled fake images.

#### Modified Discriminator Loss

$$\frac{1}{2Cn}\sum_{i=1}^{n}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1 - \langle w, 1 - \Phi_\zeta(x^{(i)}) + \Phi_\zeta(g_\theta(z^{(i)}))\rangle - b\right) +$$
$$\frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1 - \langle w, 1 + \Phi_\zeta(g_\theta(z^{(i)}) - \Phi_\zeta(x^{(i)}))\rangle + b\right)$$

#### Discriminator network architecture

layer types, activations, output shapes, and number of trainable parameters per layer. Note the inclusion of Minibatch discrimination to prevent mode collapse.

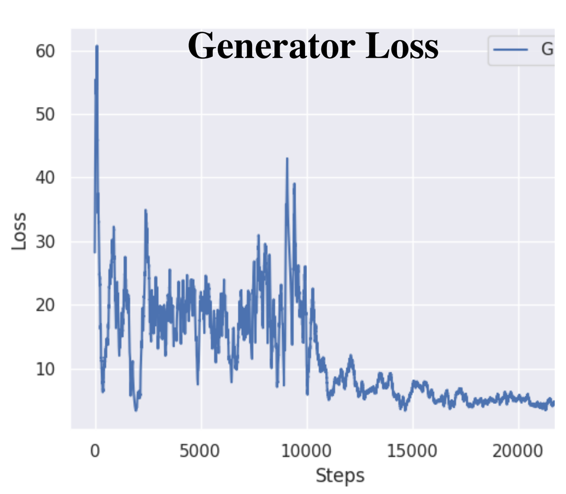
| | Activ. | Output Shape | Params. |
|---|---|---|---|
| Input map | - | $1 \times 256 \times 256$ | - |
| Conv $5 \times 5$ | LReLU | $64 \times 128 \times 218$ | 1664 |
| VirtualBatchNorm | LReLU | $128 \times 64 \times 64$ | 256 |
| Conv $5 \times 5$ | LReLU | $256 \times 32 \times 32$ | 819K |
| VirtualBatchNorm | LReLU | $256 \times 32 \times 32$ | 512 |
| Conv $5 \times 5$ | - | $512 \times 16 \times 16$ | 3.3M |
| VirtualBatchNorm | LReLU | $512 \times 16 \times 16$ | 1024 |
| Minibatch Discrimination | - | $256 \times 256$ | $256^2$ |
| Conv $5 \times 5$ | LReLU | $512 \times 16 \times 16$ | 4.4M |
| Linear | ReLU | 1 | 131K |

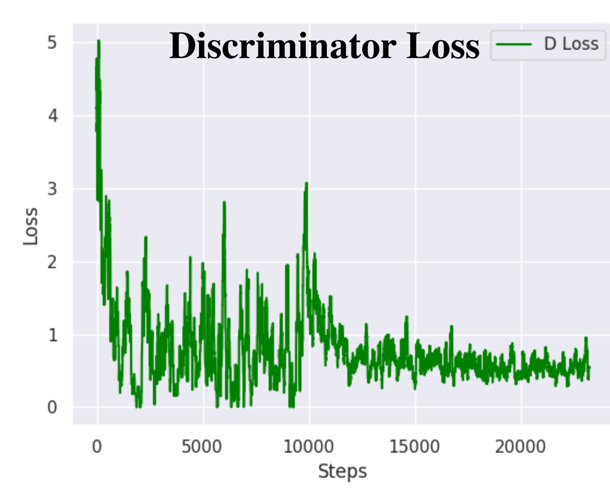
| | Activ. | Output Shape | Params. |
|---|---|---|---|
| Latent | - | 64 | - |
| Dense $5 \times 5$ | - | $512 \times 16 \times 16$ | 8.5M |
| VirtualBatchNorm | ReLU | $512 \times 16 \times 16$ | 1024 |
| TConv $5 \times 5$ | - | $256 \times 32 \times 32$ | 3.3M |
| VirtualBatchNorm | ReLU | $256 \times 32 \times 32$ | 512 |
| TConv $5 \times 5$ | - | $128 \times 64 \times 64$ | 819K |
| VirtualBatchNorm | ReLU | $128 \times 64 \times 64$ | 256 |
| TConv $5 \times 5$ | - | $64 \times 128 \times 128$ | 205K |
| VirtualBatchNorm | ReLU | $64 \times 128 \times 128$ | 128 |
| Conv $5 \times 5$ | Tanh | $1 \times 256 \times 256$ | 1601 |

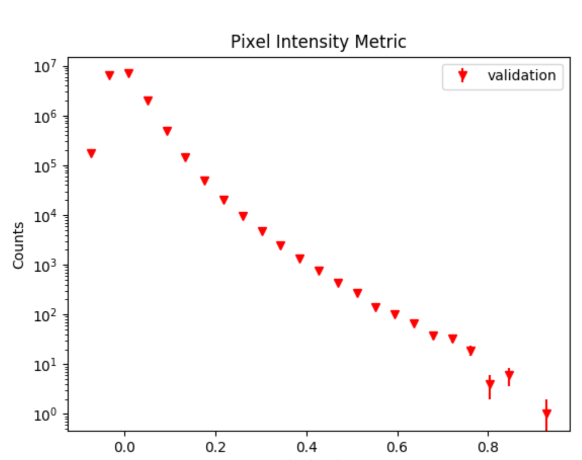
#### Generator network architecture

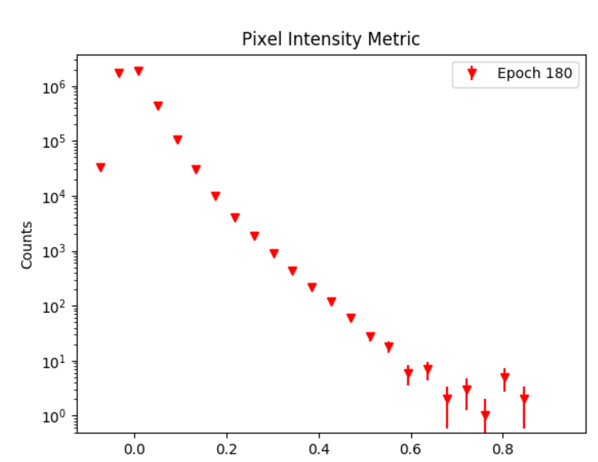
Note the inclusion of Virtual Batch Normalization to optimize neural networks.

## Plots



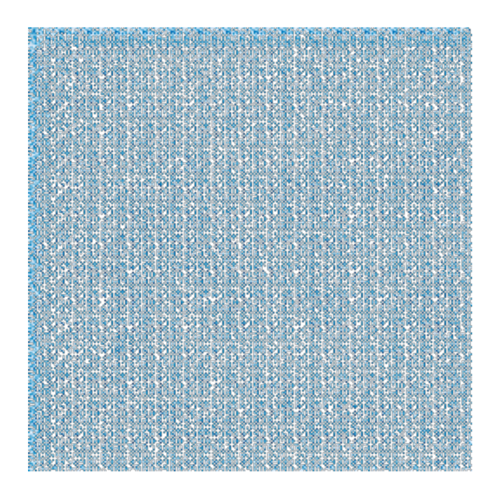



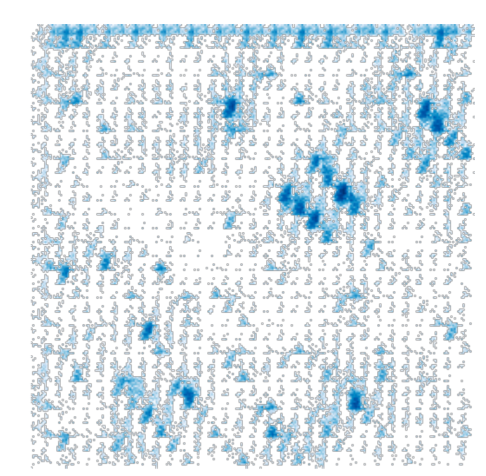




Generated pixel intensity metrics for validation (right) and generator model output (left)

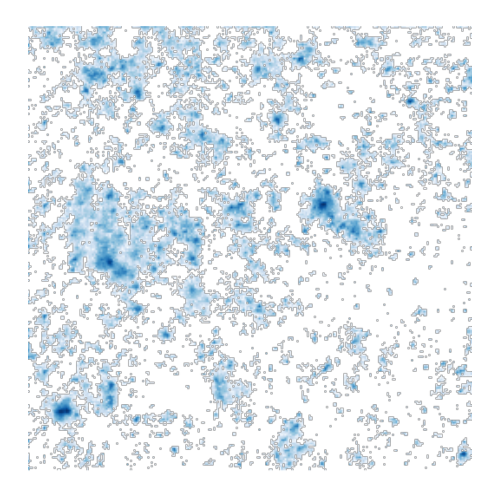
## Generated Images



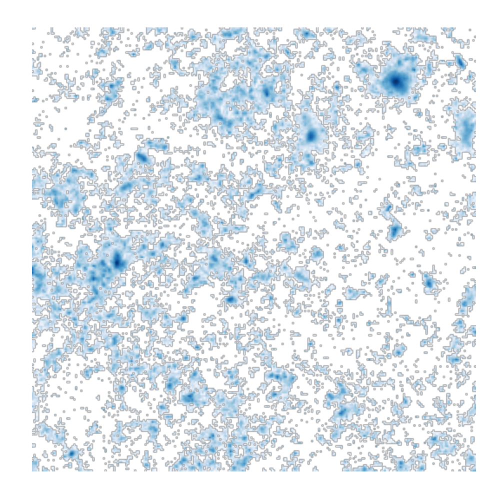
**GeoGAN Epoch 5**



**GeoGAN Epoch 80**

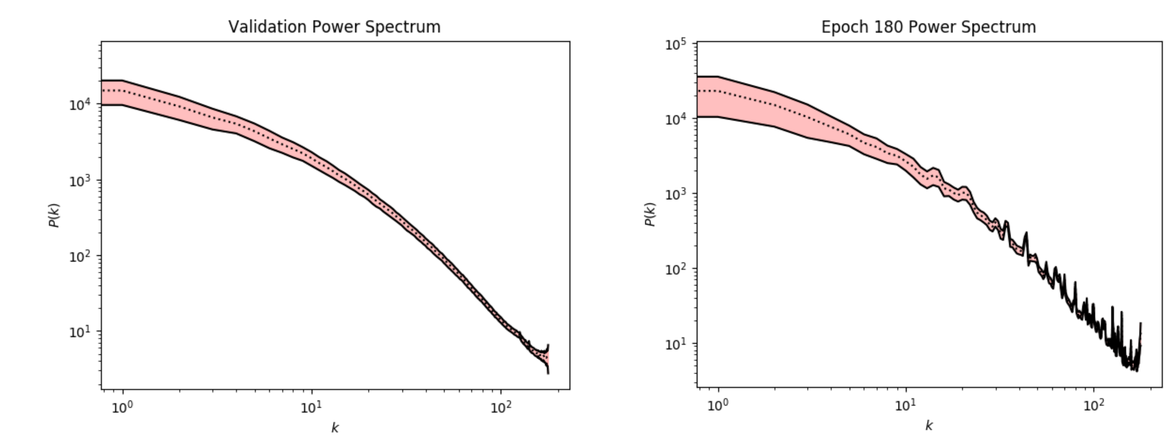


**GeoGAN Epoch 185**



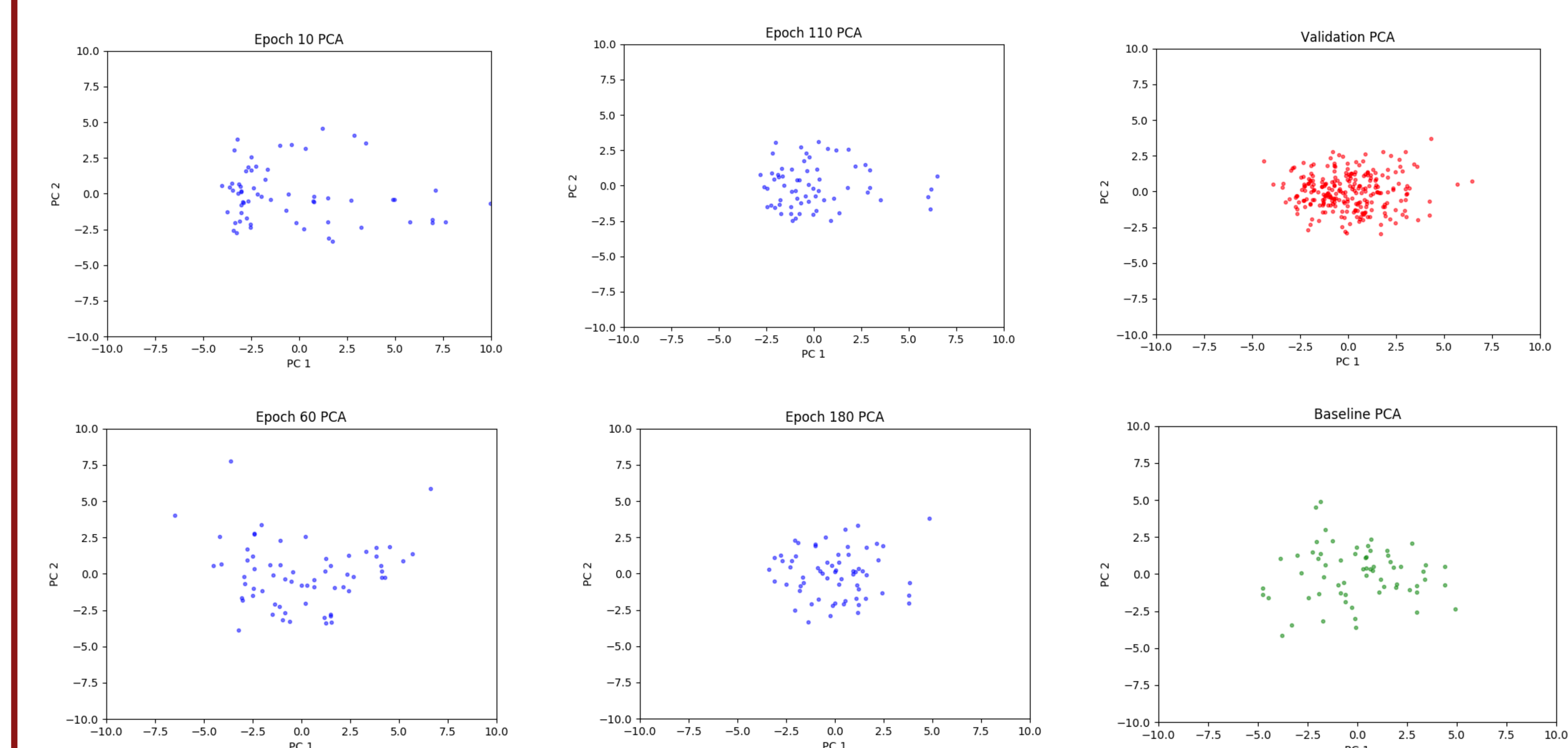
**Sample Baseline Image**

## Power Spectrum




**Power Spectrum**
A measure of correlation in gravitational lensing at different distances, evaluated at 248 Fourier modes.

## Principal Components Analysis




## Summary

- GeoGAN(normal + relativistic hybrid) converges to similar test statistics as baseline in far less time (~ 23.75 hours) than research paper (~4 days).
- No observed mode collapse behavior, which was a challenge in baseline paper.
- Both GeoGAN and Relativistic GeoGAN showed similar levels of training stability.

**Future Work**

- Explore more advanced metrics as test statistics, such as Minkowski functionals.
- Assess ability of GAN
- Extend approach from 2D to 3D mass simulations.
- Generate correlation matrices to analyze statistical independence of generated samples.

## References

[1] Springel, V. The cosmological simulation code GADGET-2. Monthly Notices of the Royal Astronomical Society, 364, 1105–1134 (2005). doi:10.1111/ j.1365-2966.2005.09655.x

[2] Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. CoRR, abs/1511.06434 (2015).

[3] Jae Hyun Lim and Jong Chul Ye. "Geometric GANs". In: (2017). eprint: arXiv:1705.02894.

[4] M. Mustafa, D. Bard, W. Bhimji, R. Al-Rfou, and Z. Lukic, "Creating ´ Virtual Universes Using Generative Adversarial Networks," ArXiv eprints, Jun. 2017.

[5] Martin Arjovsky and Léon Bottou. Towards Principled Methods for Training Generative Adversarial Networks. 2017. eprint: arXiv:1701.04862.

[6] Rodríguez, A.C., et al. Fast cosmic web simulations with generative adversarial networks. Computational Astrophysics and Cosmology, 5(1), 4 (2018). doi:10. 1186/s40668-018-0026-4.