Generative Adversarial Networks

Prasanna Balaprakash

Assistant Computer Scientist

Mathematics and Computer Science Division &

Leadership Computing Facility

Argonne National Laboratory

Outline

Generative Modeling

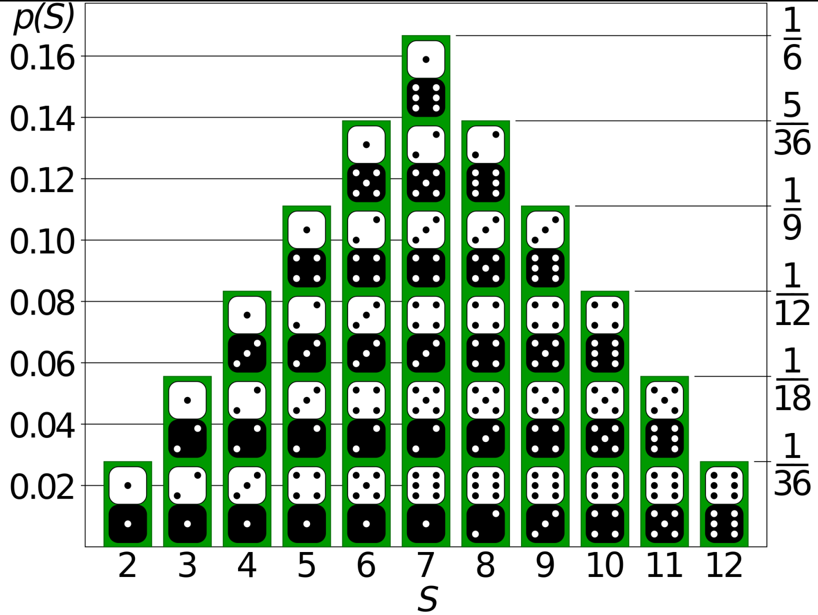
Generative Adversarial Networks

Tips and Tricks

Applications and Extensions

Generative Models

Sample data

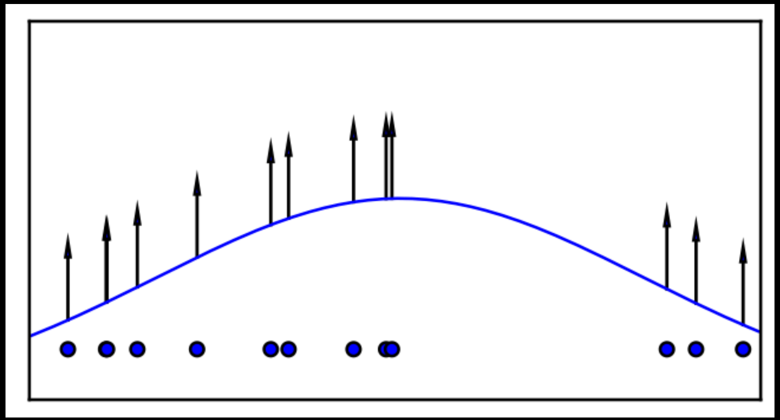


Distribution

Model how the data was generated

Generative Models

Sample data



Distribution

Model how the data was generated

Generative Models

Sample data

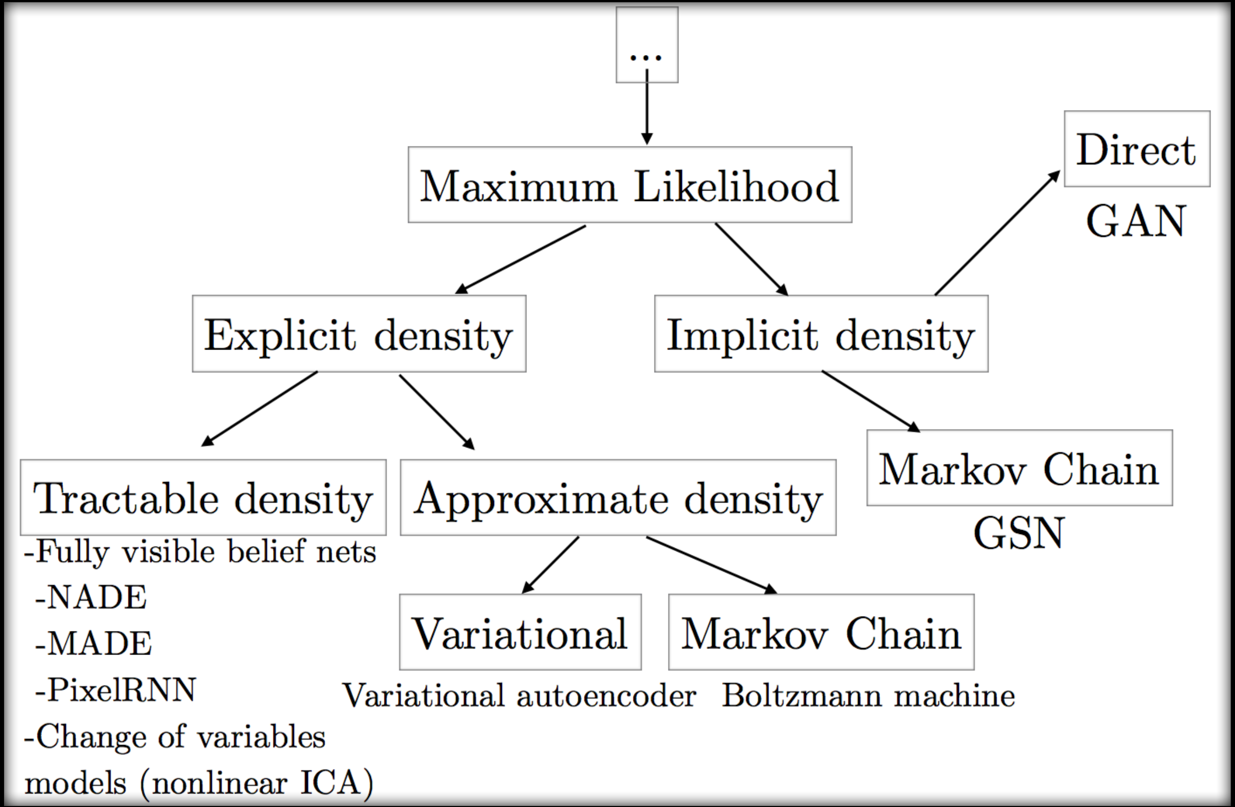
Distribution data

Generate data from the distribution

Why Generative Models?

- Ability to use high-dimensional, complicated probability distributions
- Simulate possible futures or simulated reinforcement learning
- Missing data
  - semi-supervised learning
- Multi-modal outputs (many outputs for a single input)
- Realistic generation of samples

Taxonomy of Generative Models



Outline

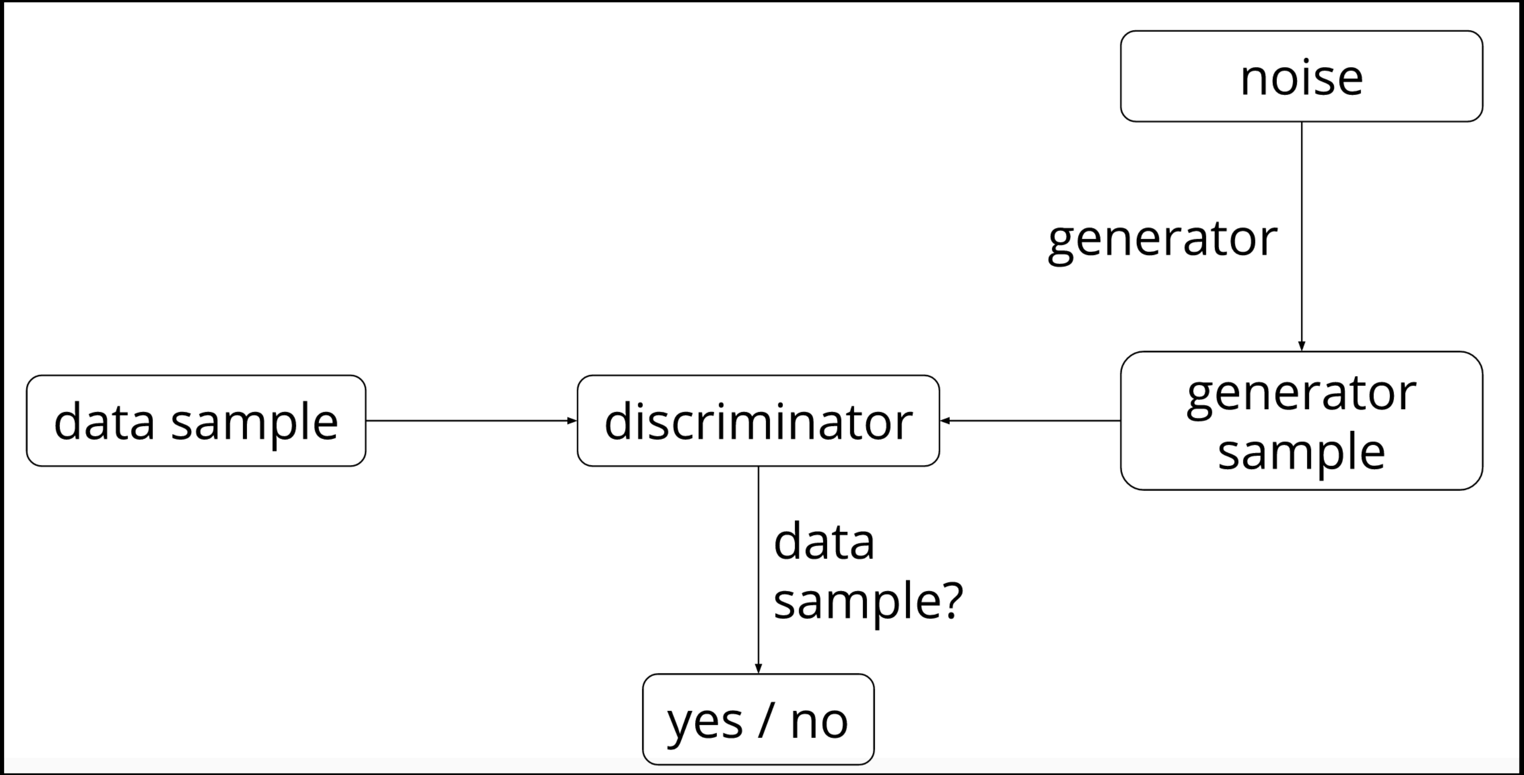
Generative Modeling

Generative Adversarial Networks

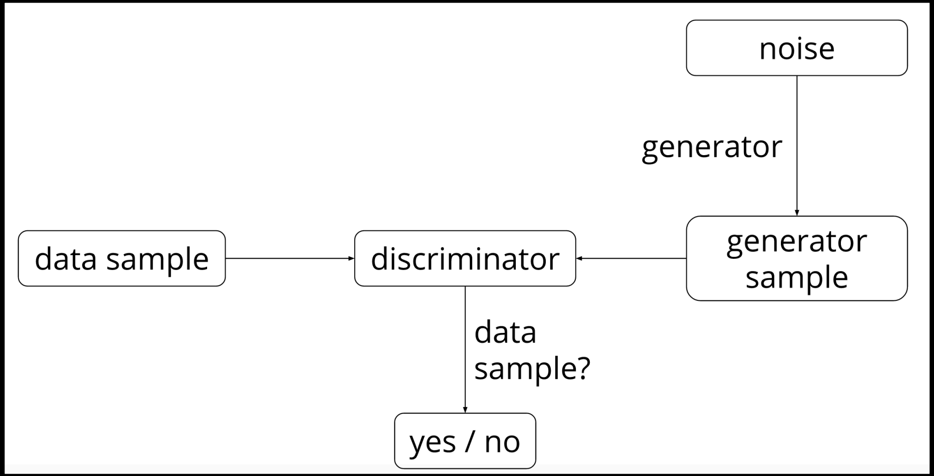
Tips and Tricks

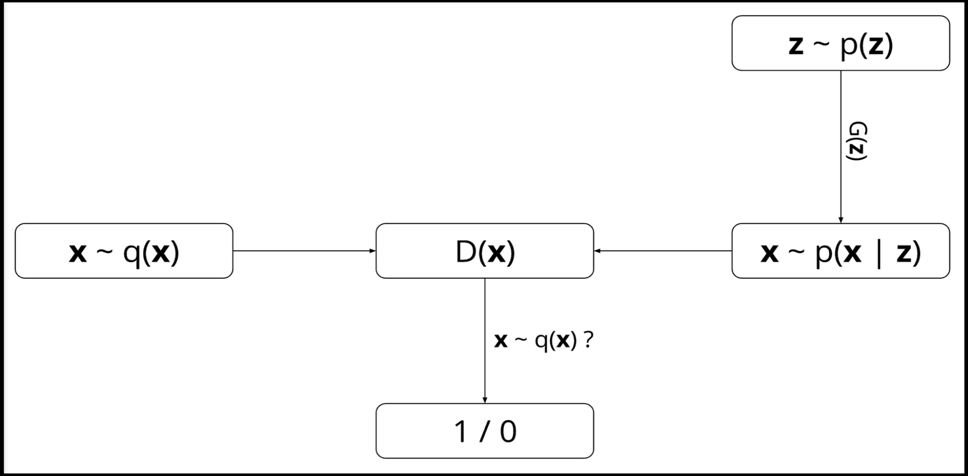
Applications and Extensions

Generative Adversarial Networks



Generative Adversarial Networks





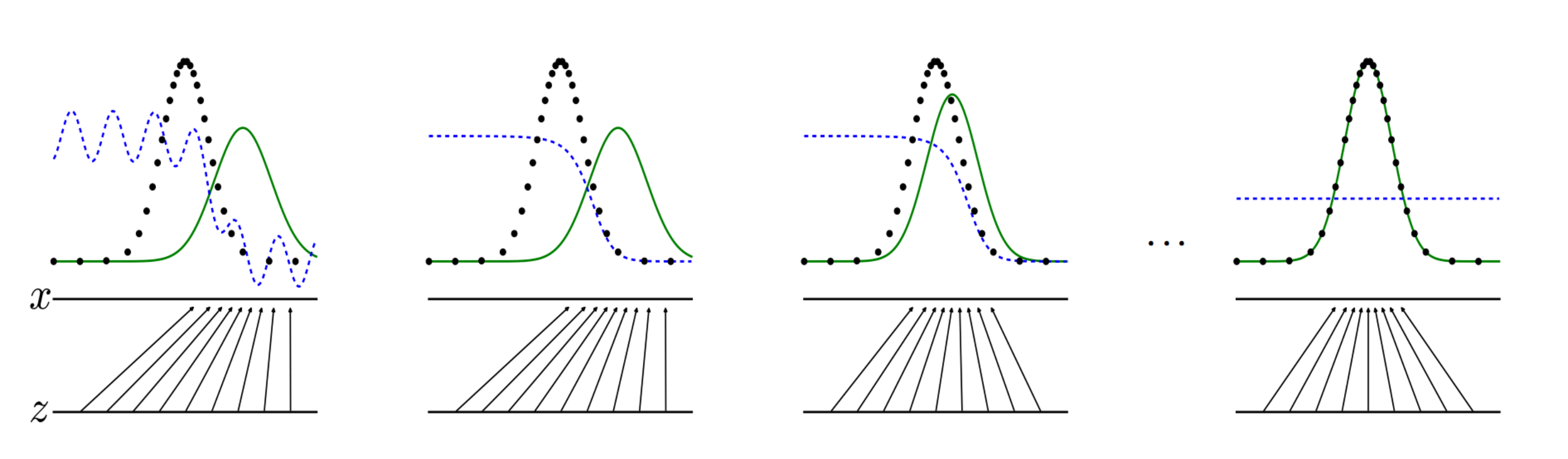
- GAN pits two neural networks against each other:
  - a discriminator network D(x)
  - a generator network G(z)

Generative Adversarial Networks

$$\min_G \max_D V(D,G) = \mathbb{E}_{q(\mathbf{x})}[\log(D(\mathbf{x}))] + \mathbb{E}_{p(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))]$$

Maximize discrimination

Minimize getting caught

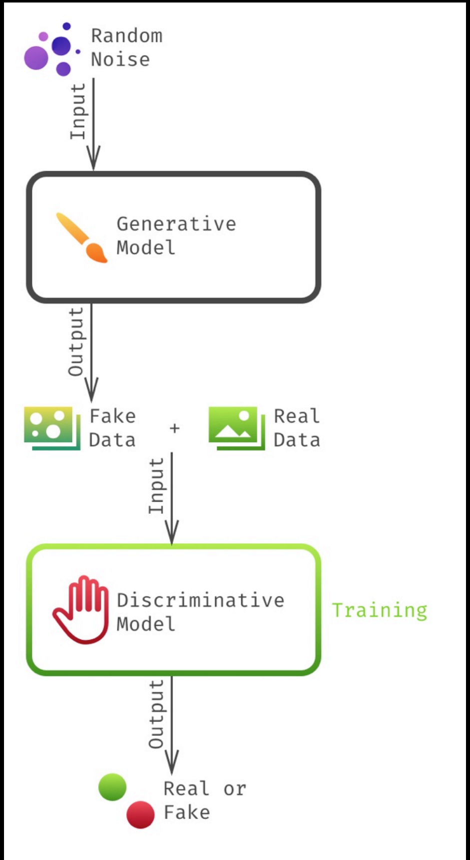


$$D^*(\mathbf{x}) = \frac{q(\mathbf{x})}{q(\mathbf{x}) + p(\mathbf{x})}$$

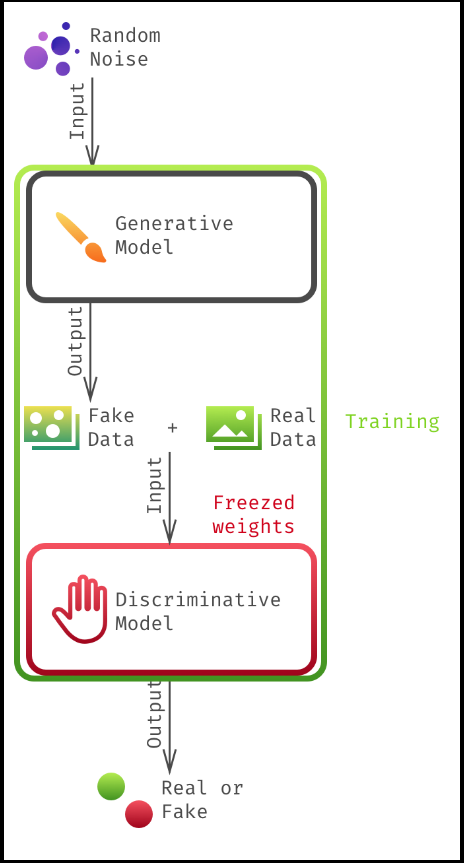
$$D(\boldsymbol{x}) = \frac{1}{2}$$

Generative Adversarial Networks

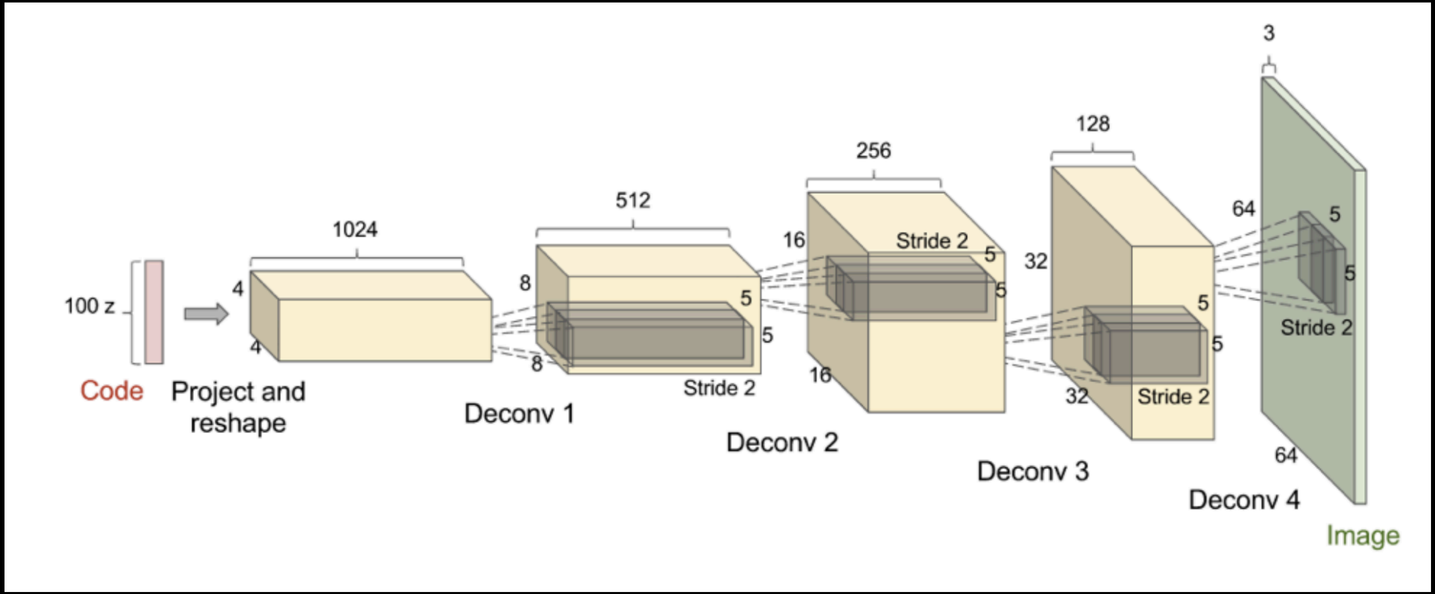
Step1:
Train the discriminator



Step2:
Train the generator via chained models



DCGAN Architecture

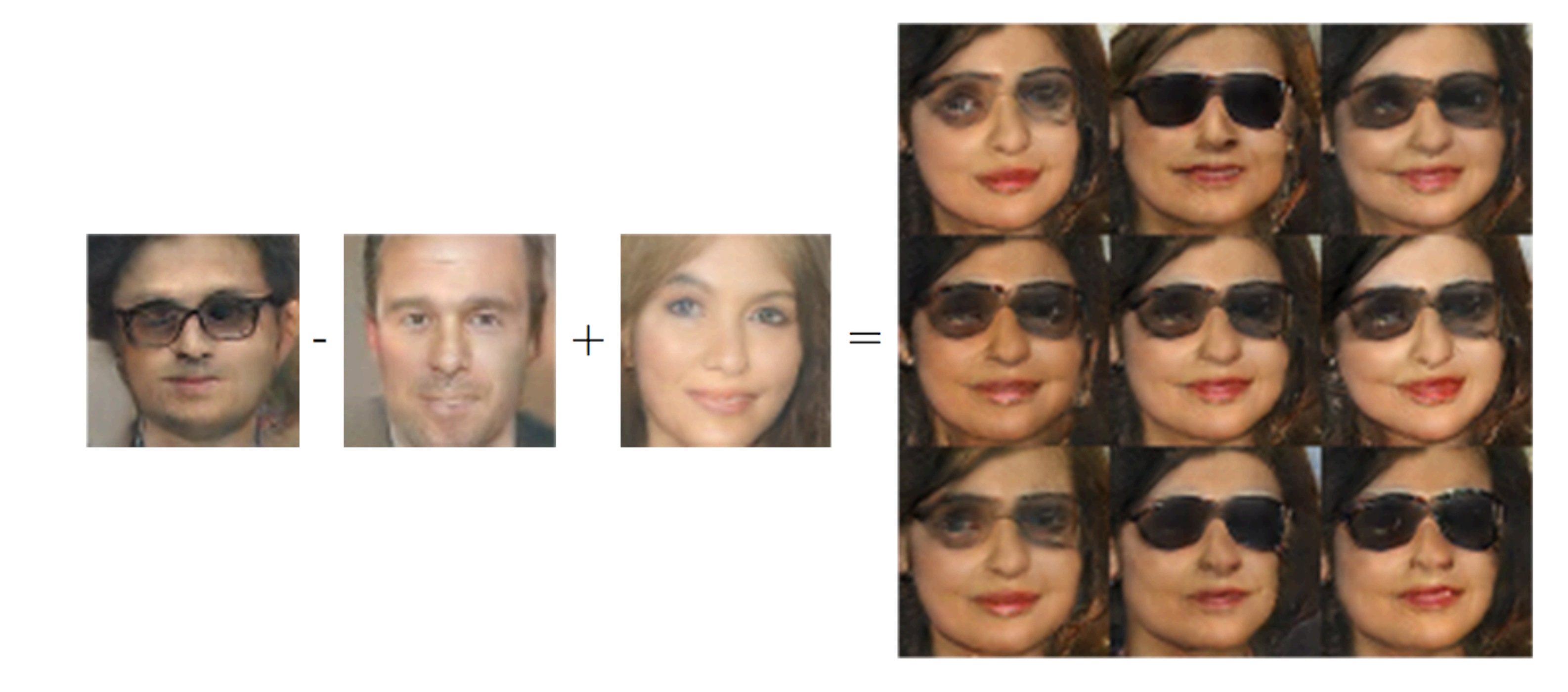


De-convolution: convolution run backwards (with stride > 1)
Batch normalization: faster and stable learning
No pooling layers: because pooling operation is not invertible

DCGAN Architecture

Sample images of bedrooms generated by a DCGAN trained on the LSUN dataset

Vector Space Arithmetic



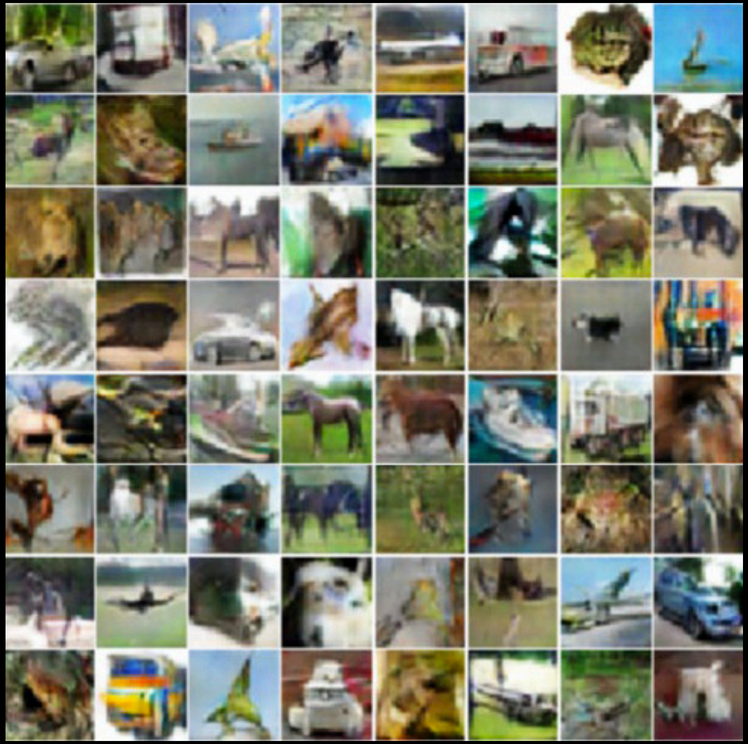
Outline

- Generative Modeling
- Generative Adversarial Networks
- <u>Tips and Tricks</u>
- Applications and Extensions

Mode collapse

- Generator produce same image
- Maximize discriminator and minimize generator
  - fully maximize the discriminator first is OK
  - fully minimize the generator first is NOT OK
    - maps the noise to the argmax of the discriminator
- Use mini-batch with image diversity

Imagenet and CIFAR10



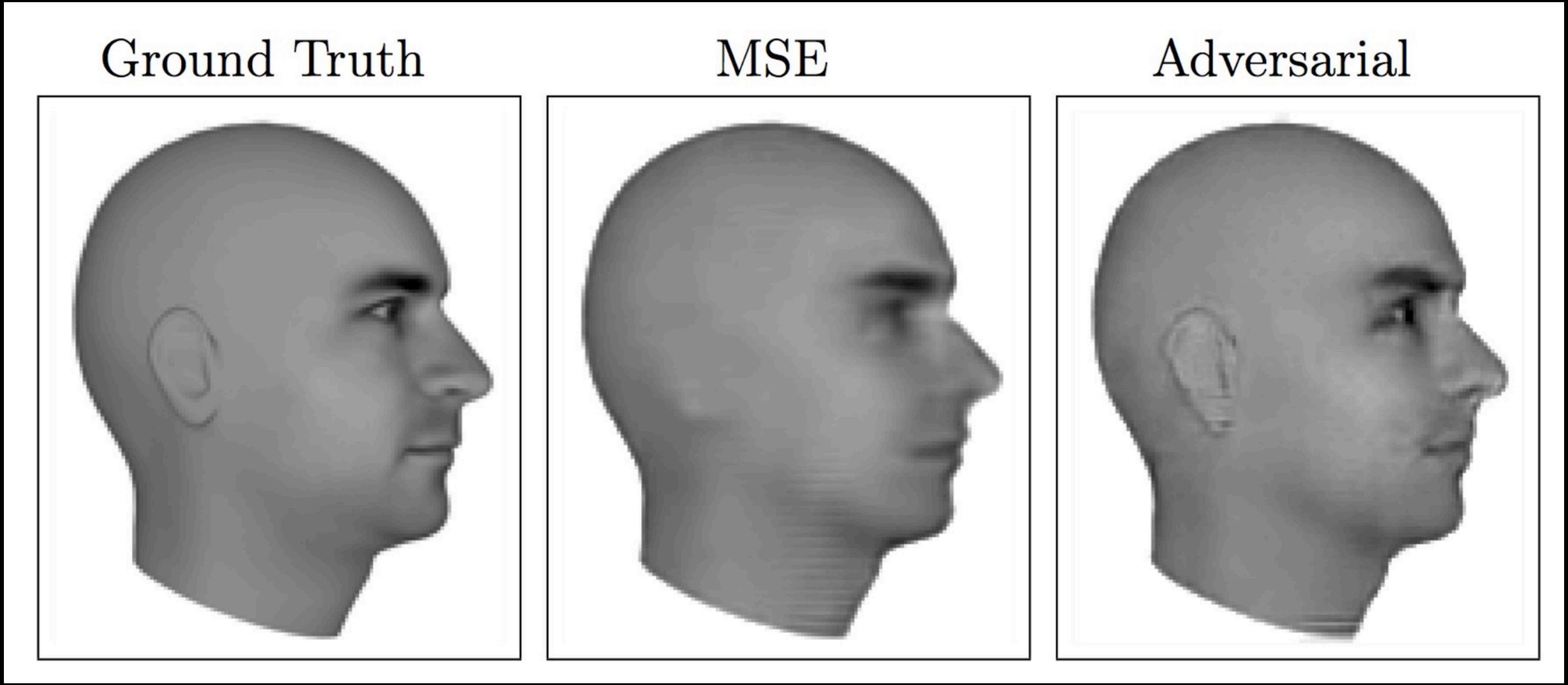
Tips and tricks to make GANs work

- Normalize the inputs
  - normalize the images between -1 and 1
  - Tanh as the last layer of the generator output
- Loss function to optimize G is min (log 1-D), but in practice use max log D
- Sample Z from a Gaussian distribution
- Construct different mini-batches for real and fake
- Don't GAN without reading
  - https://github.com/soumith/ganhacks

Outline

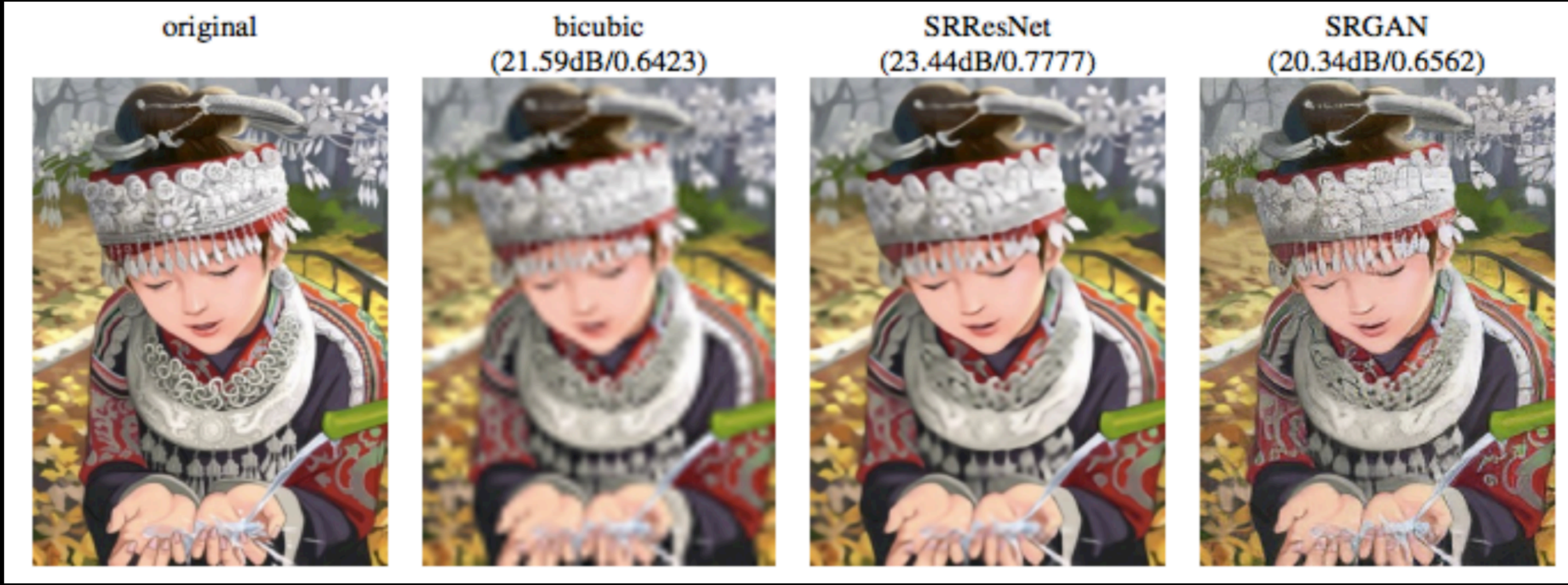
- Generative Modeling
- Generative Adversarial Networks
- Tips and Tricks
- <u>Applications and Extensions</u>

Next video frame prediction



(Lotter et al 2016)

Image super resolution

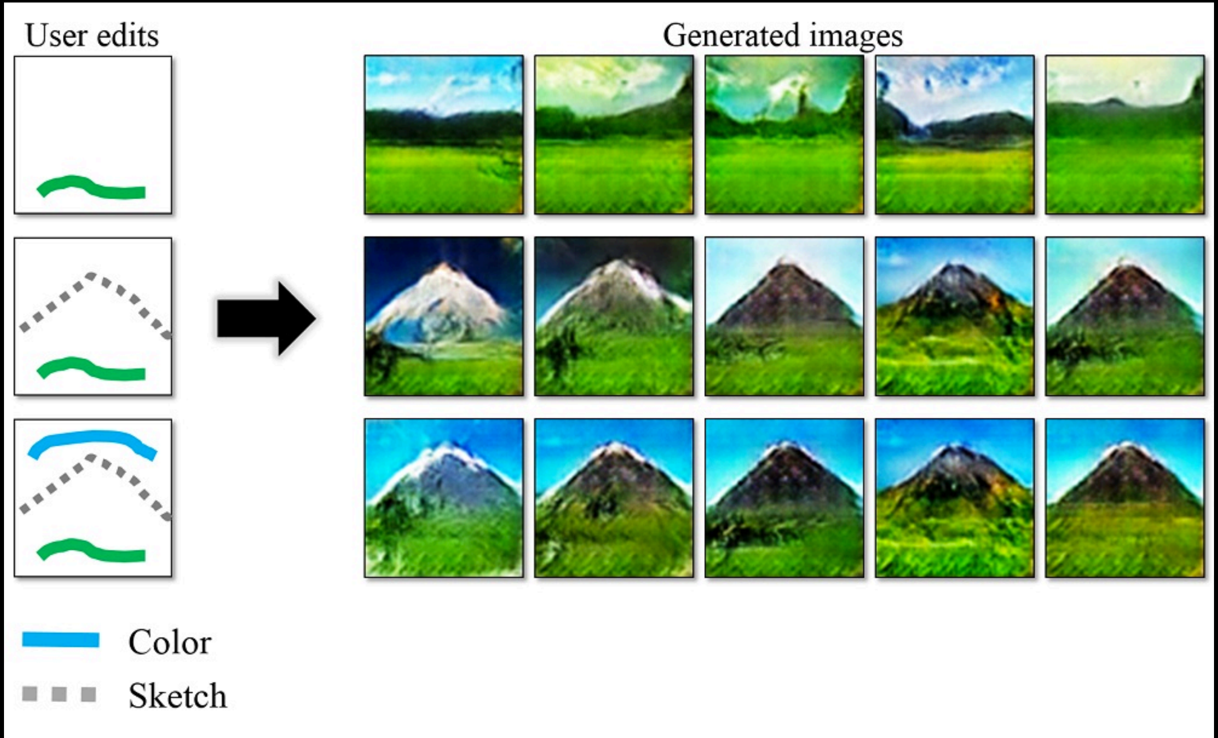


(Ledig et al 2016)

Image to image translation

(Isola et al 2016)

iGAN



(Zhu et al 2016)

Extensions

- Adversarial auto encoders
- Semi-supervised learning
- Transfer learning
- Reinforcement learning
- Inverse problems
- …

Outline

Generative Modeling

Generative Adversarial Networks

Tips and Tricks

Applications and Extensions

Conclusion and summary

- GANS
  - approximate probability density
  - use supervised learning to approximate an intractable cost function
  - simulate many cost functions, including the one used for maximum likelihood
  - generate high resolution samples from diverse image classes
- Successful in several (image-based) applications
- Training is still an art than science
- Many interesting research directions and extensions

“Adversarial training is the coolest thing since sliced bread.”

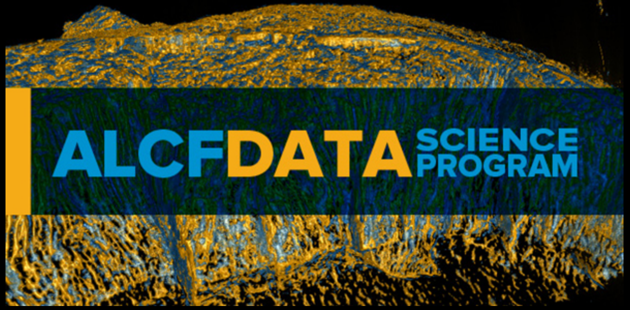
Yann LeCun, Director of AI Research at Facebook and Professor at NYU

Acknowledgements

- I. Goodfellow, et al. “Generative adversarial nets.” Advances in neural information processing systems. 2014.
- T. Salimans et al. “Improved techniques for training gans.” Advances in Neural Information Processing Systems. 2016.
- I. Goodfellow. "NIPS 2016 Tutorial: Generative Adversarial Networks." arXiv preprint arXiv:1701.00160 (2016)
- http://www.iangoodfellow.com/presentations.html
- https://ishmaelbelghazi.github.io/ALI/
- https://github.com/soumith/ganhacks
- http://www.rricard.me/machine/learning/generative/adversarial/networks/2017/04/05/gans-part1.html
- https://blog.openai.com/generative-models/

ALCF Data Science Program (ADSP) Proposals due June 15

- "Big Data" science problems that require the scale and performance of leadership computing resource such as Theta and Aurora, and will enable new science and novel usage modalities on these systems.
- Projects will cover a wide variety of application domains that span computational, experimental and observational sciences.
- Focus on data science techniques including but not limited to statistics, machine learning, deep learning, UQ, image processing, graph analytics, complex and interactive workflows
- Two-year proposal period and will be renewed annually. Proposals will target **science** and **software technology** scaling for data science. In 2016, four projects were funded spanning material science, neuroscience, imaging and high-energy physics.
- Projects receive ALCF staff support in Data and Computational Science. Tier-1 projects will be supported in part with postdoctoral scholars.
- Yearly call for proposal.
  Next deadline – June 15, 2017 (5 PM CST)
  https://www.alcf.anl.gov/alcf-data-science-program



ADSP System Resources

Mira – IBM BG/Q

Cooley - Cray/NVIDIA

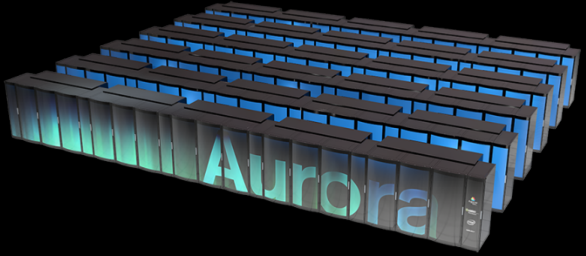
- 126 nodes (Haswell)
- 1512 cores
- 126 Tesla K80
- 48 TB RAM (3 TB GPU)

Theta- Cray/Intel

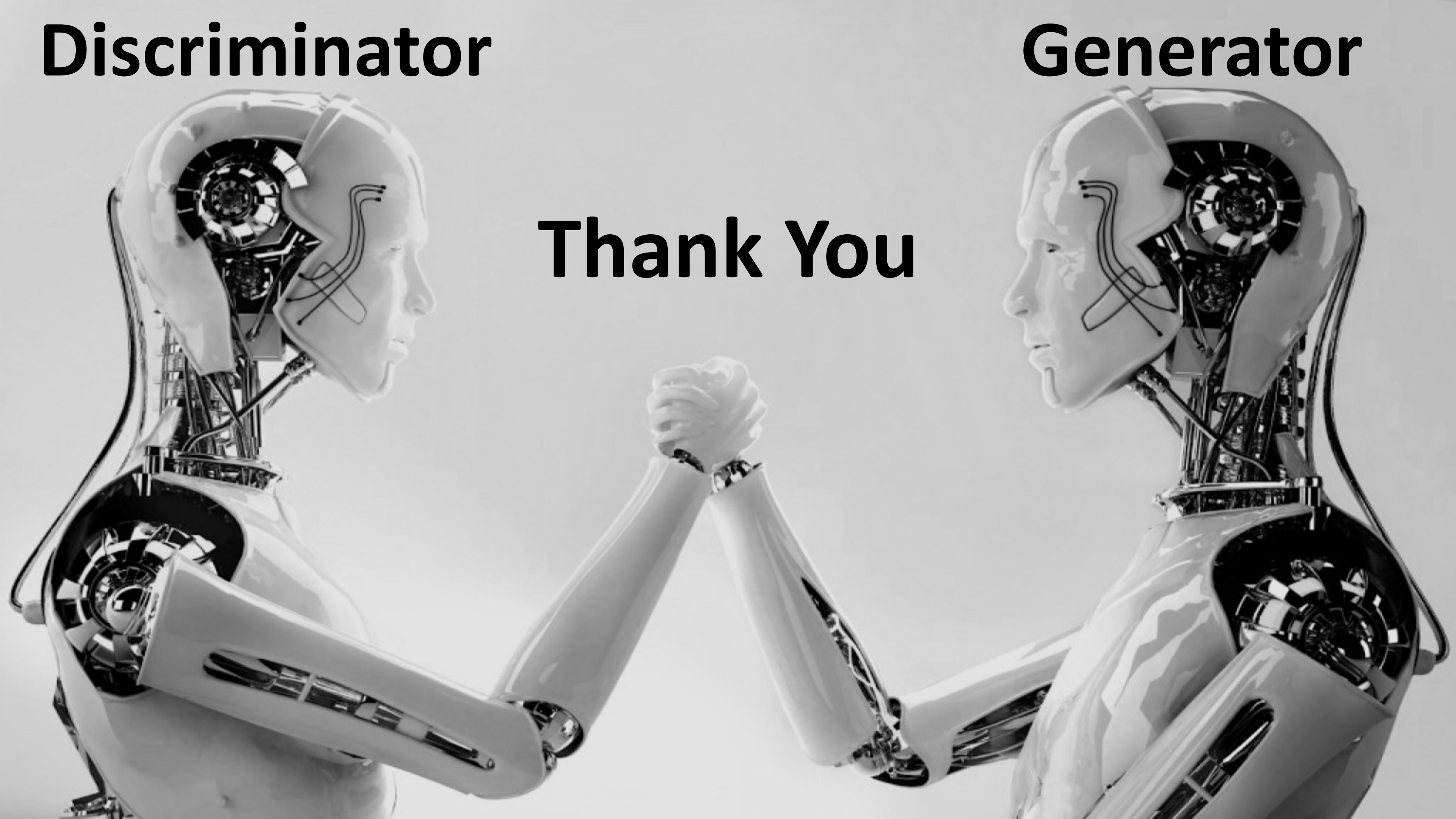
- 3240 nodes (KNL)
- 829,440 cores
- 50.6 TB MCDRAM
- 607.5 TB DDR4 RAM
- 414.7 TB SSD

Sage- Cray Urika-GX

- 32 nodes (Haswell)
- 8 TB DRAM
- 25.6 TB NVMe SSD
- BigData Analytics Stack



Over 180 PF peak performance
> 50,000 nodes with 3rd Generation Intel® Xeon Phi™ processor codename Knights Hill, > 60 cores
Over 7 PB total system memory


Discriminator
Generator
Thank You