Advances of Generative Adversarial Networks: A Survey

Dirk Holscher [*], Christoph Reich [*], Martin Knahl [*], Frank Gut [*] and Nathan Clarke[†]
[*] Institute for Data Science, Cloud Computing and IT Security, Furtwangen University, 78120 Furtwangen, Germany
Email: {dirk.hoelscher, christoph.reich, martin.knahl, frank.gut}@hs-furtwangen.de
[†] Centre for Security, Communications and Network Research, Plymouth University, Plymouth, U.K.
Email: n.clarke@plymouth.ac.uk

IARIA

HOCHSCHULE FURTWANGEN UNIVERSITY HFU

UNIVERSITY OF PLYMOUTH

Presenter Introduction: Dirk Holscher

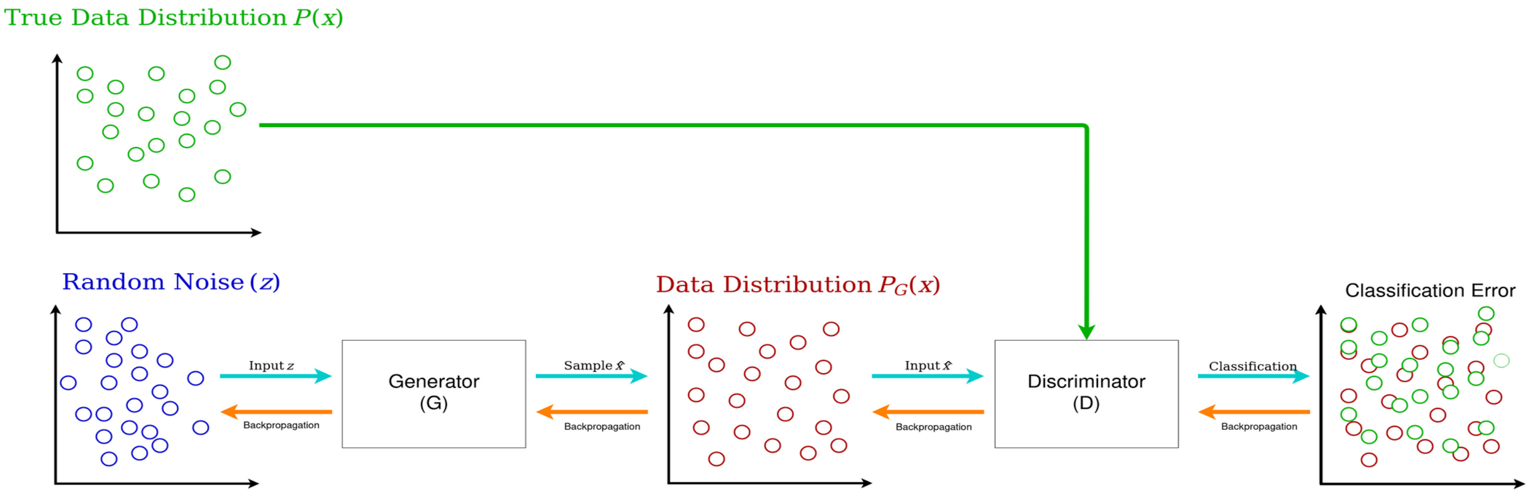
- 2013 – 2015: Furtwangen University (Germany)
- 2015 – Present: Member of Institute for Data Science, Cloud Computing and IT Security, Furtwangen University
- 2016 – 2019: Data Quality Assessment for Autonomous Driving Vehicles – Furtwangen University and Daimler AG
- 2019 – Present: PhD Student Furtwangen University (Germany) / Plymouth Univeristy (UK)

Outline

- Generative Adversarial Networks (GANs)
- Research Questions
- GAN-Architectures
- Evaluation Methods
- Existing Problems
- Conclusion

Generative Adversarial Networks (GANs)

- Generic GAN architecture
  - Generator (G) takes a random noise vector z as input to generate new samples $P_G(x)$, matching true data P(x)
  - Discriminator (D) knows P(x) and takes $P_G(x)$ as input to classify if a samples from $P_G(x)$ are real or fake
  - Error is backpropagated to G

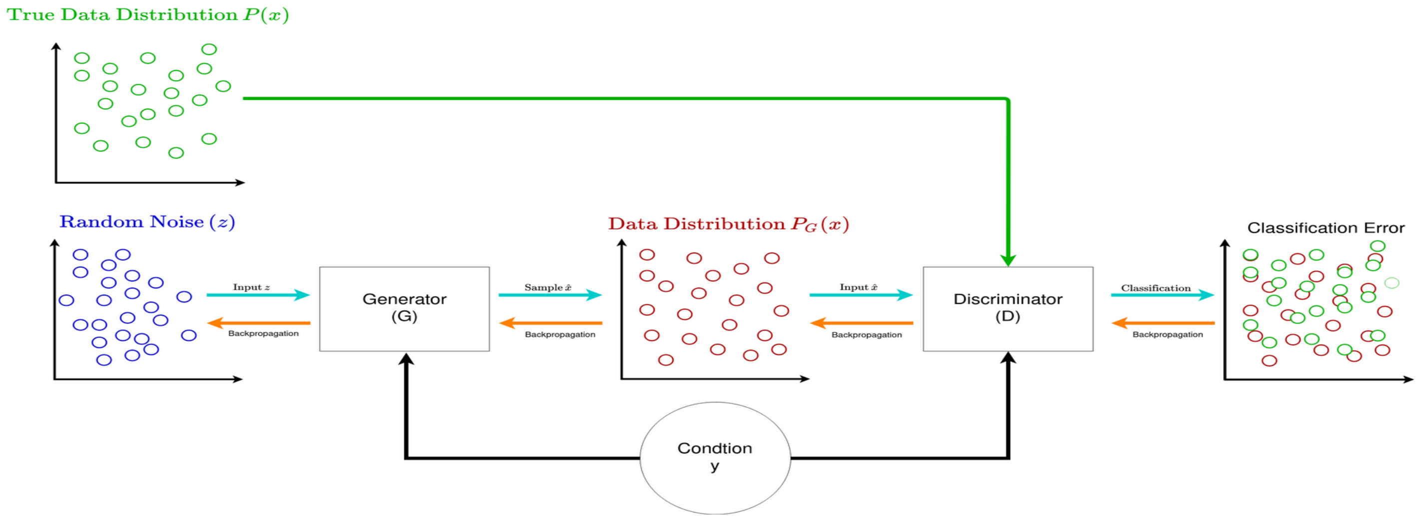


Research Questions

- The survey is based on general research questions to give an overview of the evolution of GANs and their applicability.
    - What GAN architectures exist?
    - In what domains are GANs utilized?
    - What datasets are used to evaluate GAN performance?
    - What metrics are deployed to validate GANs?
    - What challenges exist when working with GANs?
    - Advances in research?
- The survey includes publications of IEEE Explore, ACM Digital Library, NIPS Proceedings and arXiv

Conditional GAN (cGAN)

- Adding another input (y) the condition to the GAN
- y is a restricting condition (e.g., class labels) limiting sample creation to the condition
- D is using y to determine if a sample is real or fake by knowing the real data distribution P(x) as well as y



Deep Convolutional Generative Adversarial Networks (DCGAN)

- Connects GANs and Convolutional Neural Networks (CNN)
- It mainly consists of convolution layers without fully connected or max pooling layers
- Reversed process instead of receiving a list of probabilities (CNN) as the output the DCGAN inputs random numbers and receives an image
- Two models are trained; a CNN to classify real and fake images and DCGAN to generate images classified as real by the CNN

Image-to-Image Translation GANs

- Unpaired and Paired
  - Unpaired: Source domain x and target domain y, no matching x_i for every y_i
  - Paired: Source and target domain have matching pairs in both domains, for each x_i there is a corresponding y_i
- Unpaired Cycle consistent GAN
  - Simultaneously training two Generators and Discriminators (one pair for each domain)
  - G_x generates images for domain y and G_y for domain x
  - The output of G_x is used as input for D_x (trained on images of domain y) and vice versa, this is to achieve cycle consistency, where one image of domain is translated in an image of domain y and the the translated image is used to create a image of domain x and should yield the original input image

StyleGAN

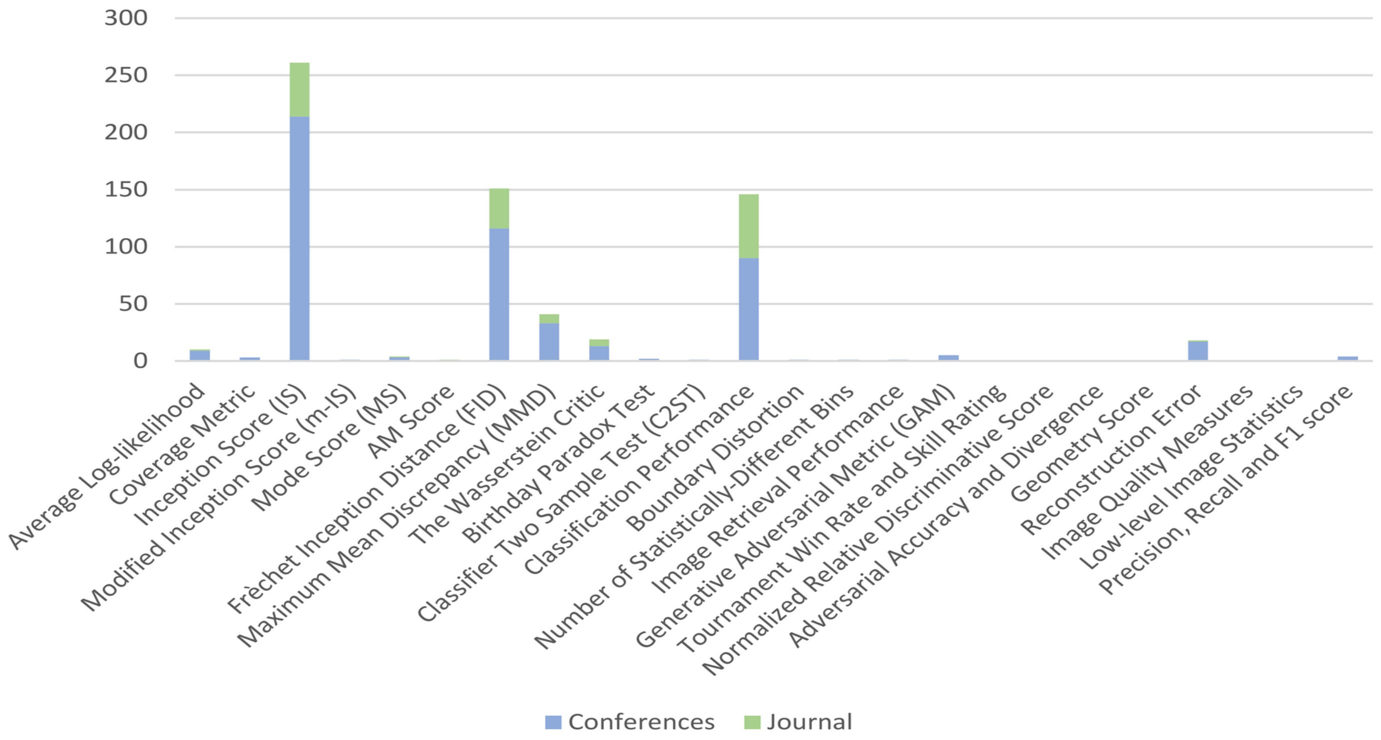
- Two sources of randomness to generate images
    - Noise Layer and Mapping Network, outputs a style vector used in each layer of the generator using an adaptive instance normalization layer
    - Noise is added at every point in G and is injected into whole feature maps for more fine grained style interpretation
    - A style is created from each learned distribution and based on the collected styles new images a created
- Progressively growing GAN with G starting to create 4x4 images until training is stable and then doubling the size till 1024x1024 is reached. The last layer converts the image into RGB via a 1x1 convolution.

StackGAN

- GANs are stacked in a tree like shape
- The first GAN is used for text-to-image synthesis creating low resolution images with rough outlines and colors
- The second GAN is conditioned on low resolution images and text embedding and tries to create high resolution images. The low resolution images of the first GAN are used to train G_2 and D_2 s
- The second GAN does not receive any additional random noise and is able to learn new previously omitted information
- The second GAN is constructed as an encoder-decoder network

Evaluation Methods

- In literature there are 24 quantative and qualitative measurements commonly used to evaluate GAN performance



Evaluation Methods

- Commonly used performance evaluation methods are: Inception Score, Frechet Inception Distance, Classification Performance
- Inception Score:
  - Utilizes pre-trained Inception classification model
  - Generated images are classified with the model (probability of an image belonging to each class)
  - Focus on image quality (how does an generated image look) and diversity (number of object which are generated)
  - Calculated score 1.0 lowest score insufficient quality and diversity and max number of classes as highest score

Evaluation Methods

- Frechet Inception Distance:
    - Enhancement of the Inception Score
    - Using pre-trained inception v3 for classification
    - Calculates the distance between two Gaussian (real image feature vectors and generated image feature vector)
    - Lower scores show higher similarities between fake and real images (fake images of high quality)
- Classification Performance:
    - Fake images are as an unsupervised feature extractor to label additional datasets and measure their performance

Evaluation Datasets

- Different GAN architectures use various datasets to compare and measure performance
- Dataset categorisation is based on architecture similarities

Architecture	Cityscape	CMP Facades	ImageNet
CycleGAN	X	X	X
Pix2Pix	X	X	X

Architecture	CIFAR-10	CLEVR	Flying Chairs	CUB	MS COCO	Oxford-102	LSUN	ImageNet
StackGAN	X	X	X					X
CP-GAN				X	X	X	X	

Evaluation Datasets

Architecture	MNIST	CIFAR-10	TFD	SVHN	FFHQ	ImageNet	MIR Flickr 25k	YFCC100M2	CelebA	LSUN
GAN	X	X	X							
cGAN	X					X	X	X		
InfoGAN	X			X					X	
Wasserstein										X
DCGAN	X	X		X		X			X	X
StyleGAN					X					X

GANs Existing Problems

- Mode Collapse:
  - Lack of diversity in generated samples
  - G finds a sweet-spot mode and concentrates on it
  - G will abuse the sweet-spot to produce more plausible output in order to fool D
  - D learns the pattern and always classify it to be fake, this can lead to D getting stuck which an updated version of G will abuse
- Non Convergence:
  - With G improving due to backprogation, D’s performance will get worse
  - At one point D will randomize its output and become negligible
  - If G is trained past the point of convergence the generated samples will get worse

Conclusion

- GANs show your adaptability and suitability to be deployed various domains
- Various evaluation methods making it hard to compare different models and require the implementation of said method
- New approaches will further stabilize training, minimize,mode collapse and non-convergence
- Synthesized samples are still distinguishable from real samples