

Fayyaz Minhas

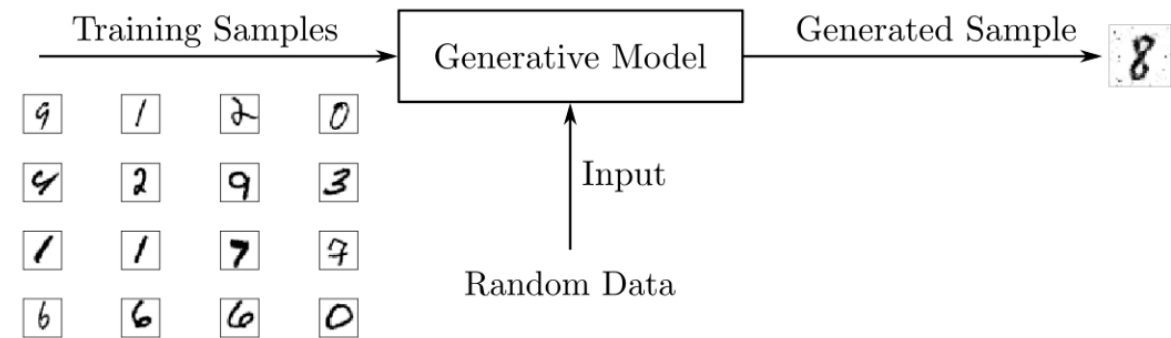
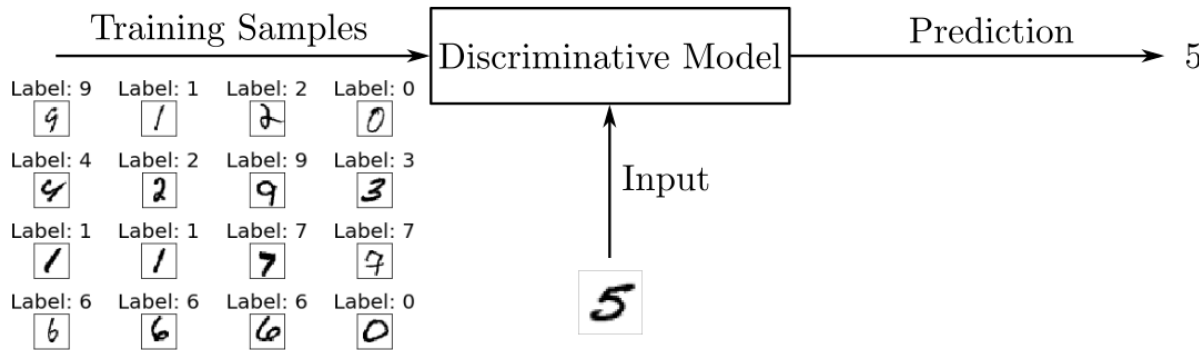
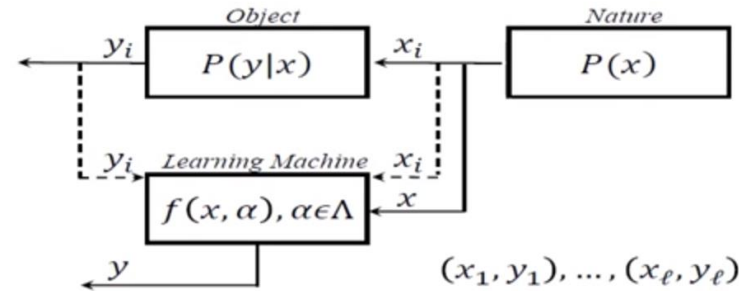
GENERATIVE MACHINE LEARNING

Introduction to Sampling

- Making a distribution from observations
 - Assume you are given a bunch of samples, can you make a probability distribution out of it?

- Sampling from a distribution
 - Assume you are given a probability distribution $p(x)$, then if you “sample” from it, you will be generating samples x which when observed will give you $p(x)$

A generative look at Machine Learning



Lex Fridman. *Complete Statistical Theory of Learning (Vladimir Vapnik)* | MIT Deep Learning Series, 2020.
<https://www.youtube.com/watch?v=Ow25mjFjSmg>.

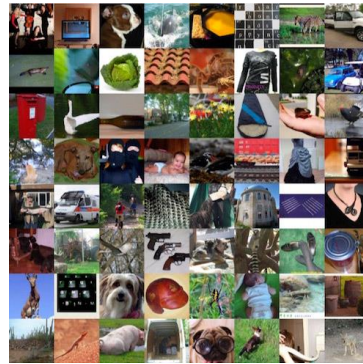
Generating data with machine learning

- Can we generate examples?
- How?
 - Density Modelling
 - Modelling the Probability of observing a given point $p(x)$
 - Once I have an explicit or implicit $p(x)$, I can sample from that distribution to generate an example
 - How to learn $p(x)$?
 - We can use a neural network!
 - $p_{model}(x; \theta)$
 - Relation to partitioning of the feature space



Generative Models

- Can we build a model to approximate a data distribution?



Real image (training data) $\sim p_{\text{data}}(x)$



Generated samples $\sim p_{\text{model}}(x)$

Want to learn $p_{\text{model}}(x)$ similar to $p_{\text{data}}(x)$

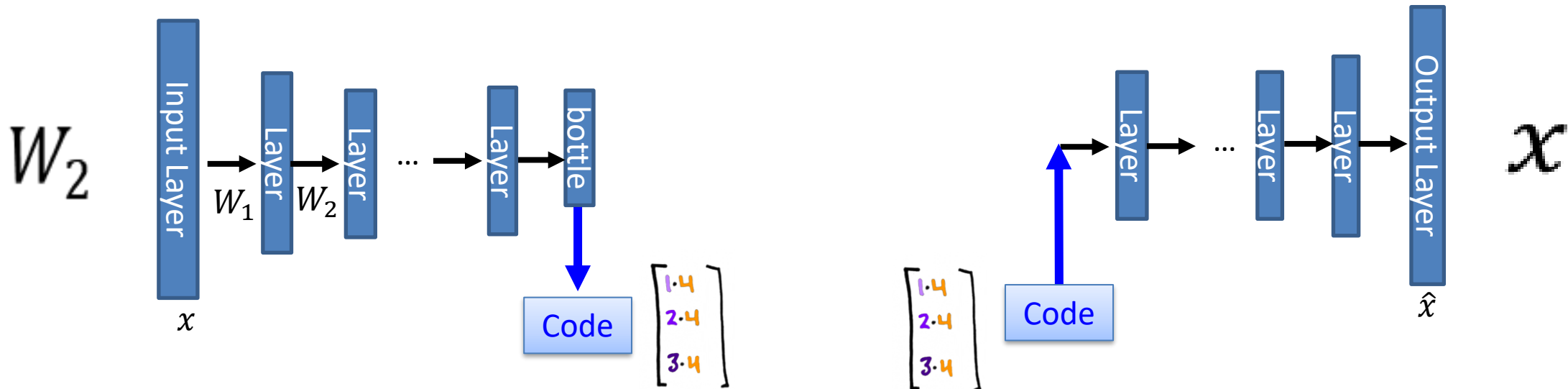
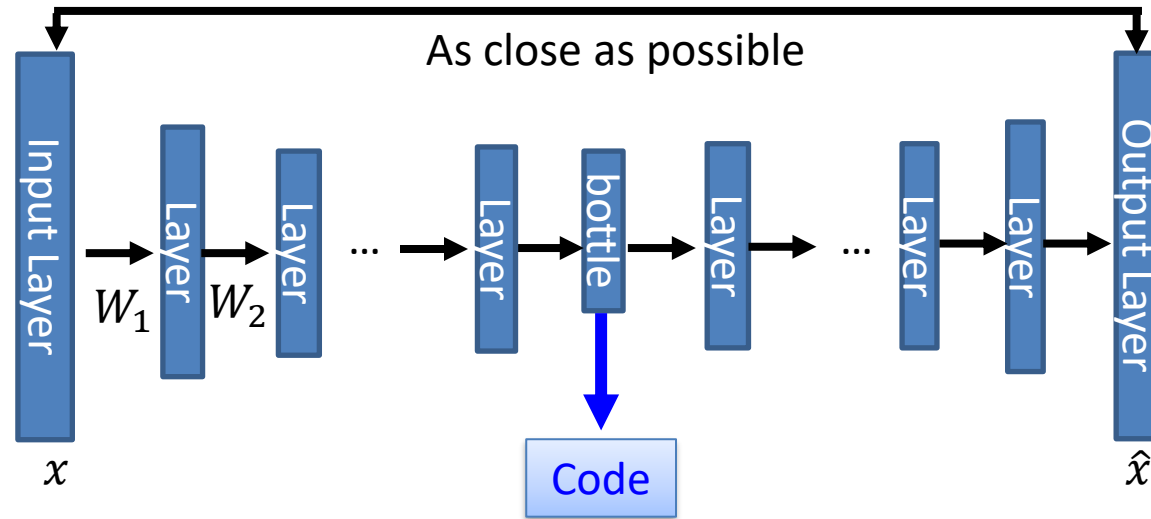
Density estimation, a core problem in unsupervised learning

Several flavors:

- Explicit density estimation: explicitly define and solve for $p_{\text{model}}(x)$
- Implicit density estimation: learn model that can sample from $p_{\text{model}}(x)$ w/o explicitly defining it

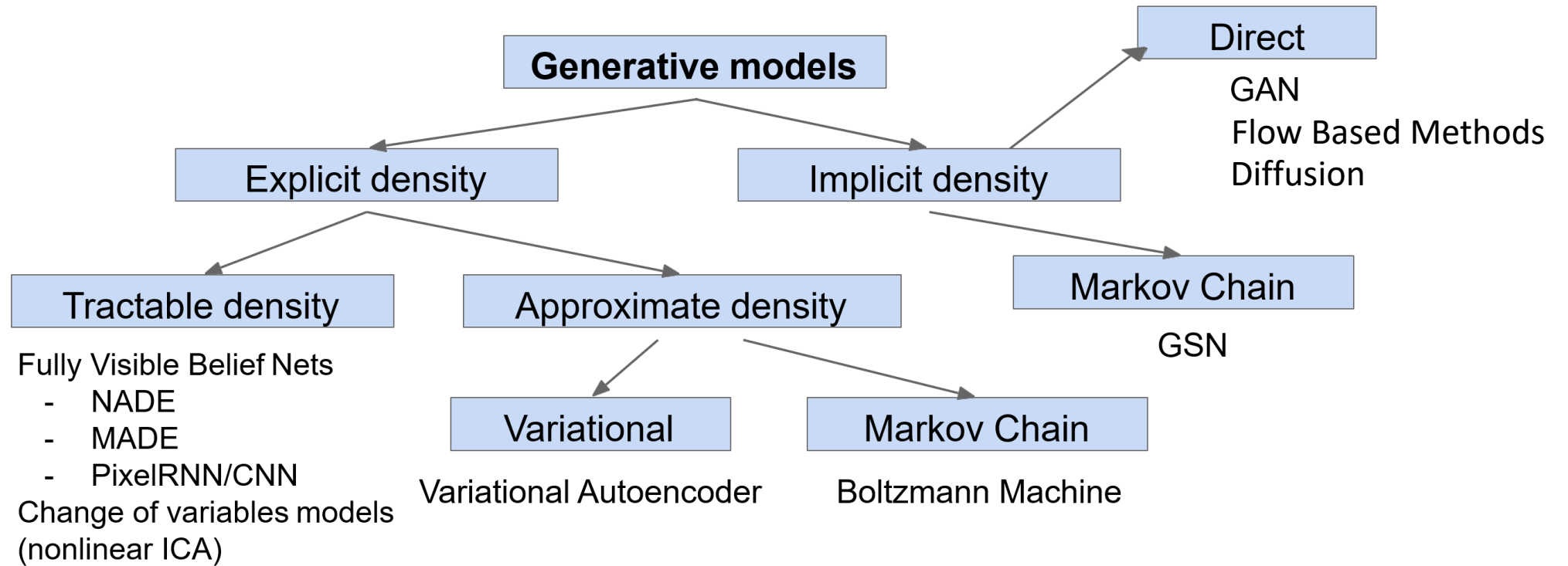
<https://openai.com/blog/generative-models/>

Generating Data with Autoencoders



Generative Models

- Can generate data



Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

A Simple Generative Machine Learning Example

- Nature
 - A coin with $p(x=H)=0.7$ and $p(x=T)=0.3$
 - Generates data
- Data
 - {H,H,H,T,T,H,T,H,H,T}
- Goal of Generative Learning
 - Make a machine learning model that can generate data (heads or tails) that follows the same distribution as the real world or natural process
 - Assume you have a learning machine G , which generates data then difference between the probability distributions of real and generated samples should be small

Density Modelling in Generative Models

– Explicit Density Modelling

- The model should be able to
 - Generate a sample x
 - tell what is the probability of $p(x=H)$
- Examples
 - Variational Autoencoders

– Implicit Density Modelling

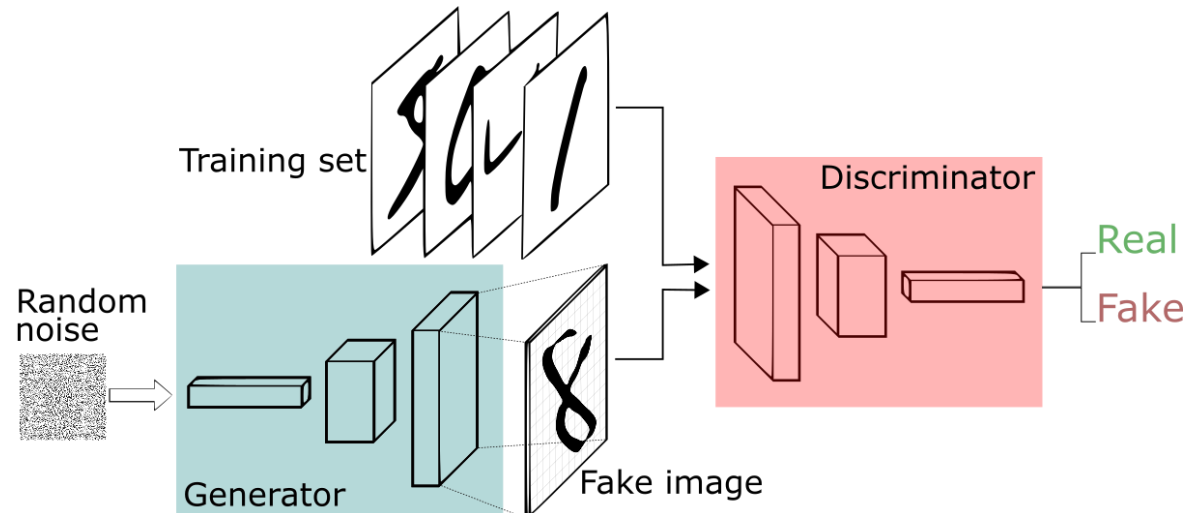
- The model should be able to
 - Generate a sample x
- Examples
 - Generative Adversarial Networks

REO for Generative Models

- Goal
 - Learn parameters of the model so that the generated examples follow the same distribution as the real world examples
- Representation: $x = f(z; \theta)$
- Evaluation: Differences between the probability distribution of x in nature, $p(x)$ and of the generated samples $p_{\theta}(x)$ from $f(z; \theta)$ should be minimized
 - So that if I sample from $p(x)$ or if I sample from $p_{\theta}(x)$, the real and generated samples are similar
 - KL Divergence
- Optimization

Generative Adversarial Networks

- A generator learns to generate data that “fits” the unknown underlying probability distribution of a training set based on whether a “discriminator” can distinguish between the samples from the wild and the generated ones



GAN Tutorial

 Open in Colab

A Barebones GAN in PyTorch for generating coin flips

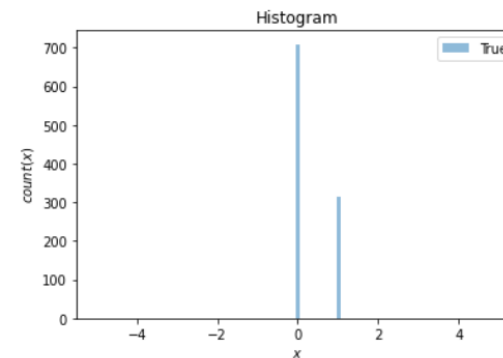
By Fayyaz Minhas

Let's consider a very simple coin toss as a process that generates coin flips with a probability of 0.3 of producing heads. We can describe the underlying probability distribution for this generative process (coin toss) as $p(x)$ where $x \in \{H = 1, T = 0\}$ is sampled from $p(x)$, i.e., $x \sim p(x)$. We would like to use a Generative Adversarial Network (GAN) to model this process using a number of data samples or observations from the original process for training. Specifically, we would like to have a GAN with such a generator that you (and its discriminator) wouldn't be able to tell if a series of coin tosses has been generated using the GAN or the underlying true process! In more mathematical terms, we would like to train a generative model $x = G(z; \theta_G)$ that can generate samples x using Normally distributed random input ($z \sim \mathcal{N}(0, 1)$) such that the probability distribution of these generated samples $p_G(x)$ is close to $p(x)$ without knowing $p(x)$ in advance or explicitly modelling $p_G(x)$.

Using a GAN is an overkill for this simple task and there are much simpler and more effective ways of modelling this simple problem. However, this GAN based solution is intended to help you understand how GANs can model complex densities implicitly and can be used to generate samples that mimic the true or natural generative process.

We first simulate the coin toss and generate 1024 training samples below. The histogram shows the (sample estimate of) the true density.

```
]:  
"""  
A toy GAN to generate coin tosses  
"""  
  
# Let's model the natural density and generate some data using that  
  
import torch  
from torch import nn  
  
import math  
import matplotlib.pyplot as plt  
import numpy as np  
train_data_length = 1024  
def cointoss(t):  
    phead = 0.3  
    return 1.0*(t
```

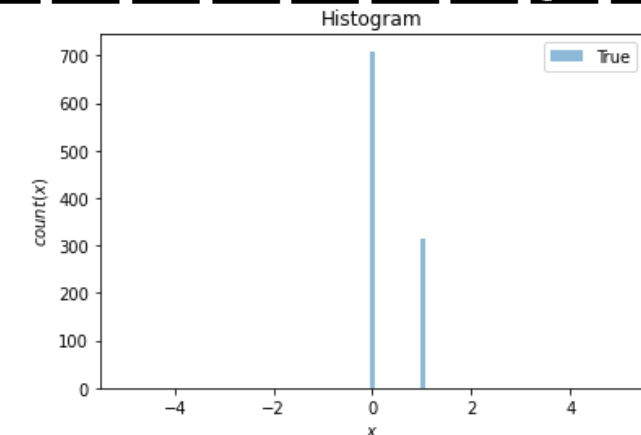
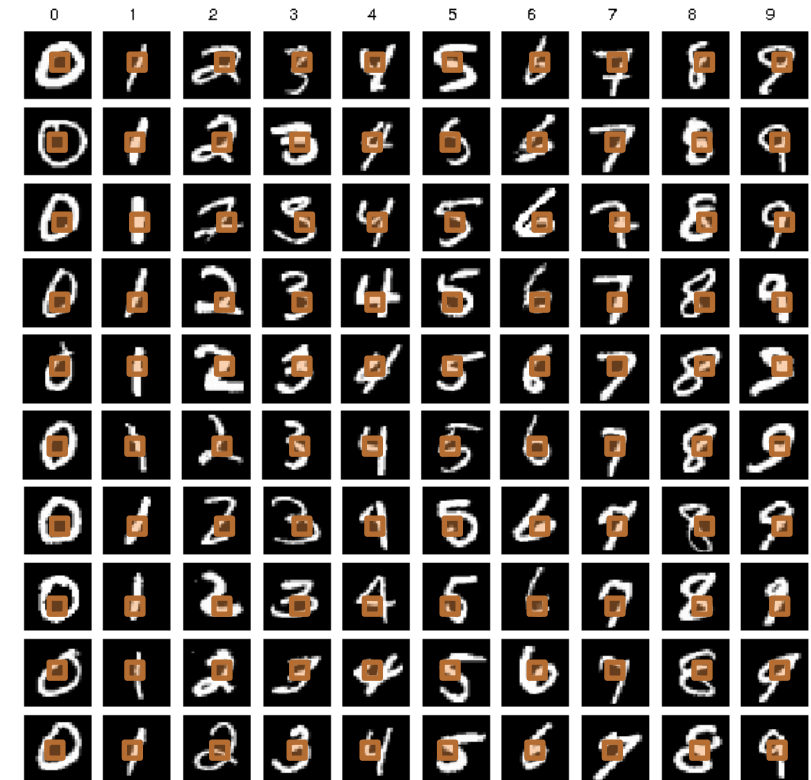


Example Data tensor([[0., 0., 0., ..., 0., 1., 1.]])

<https://github.com/foxtrotmike/CS909/blob/master/simpleGAN.ipynb>

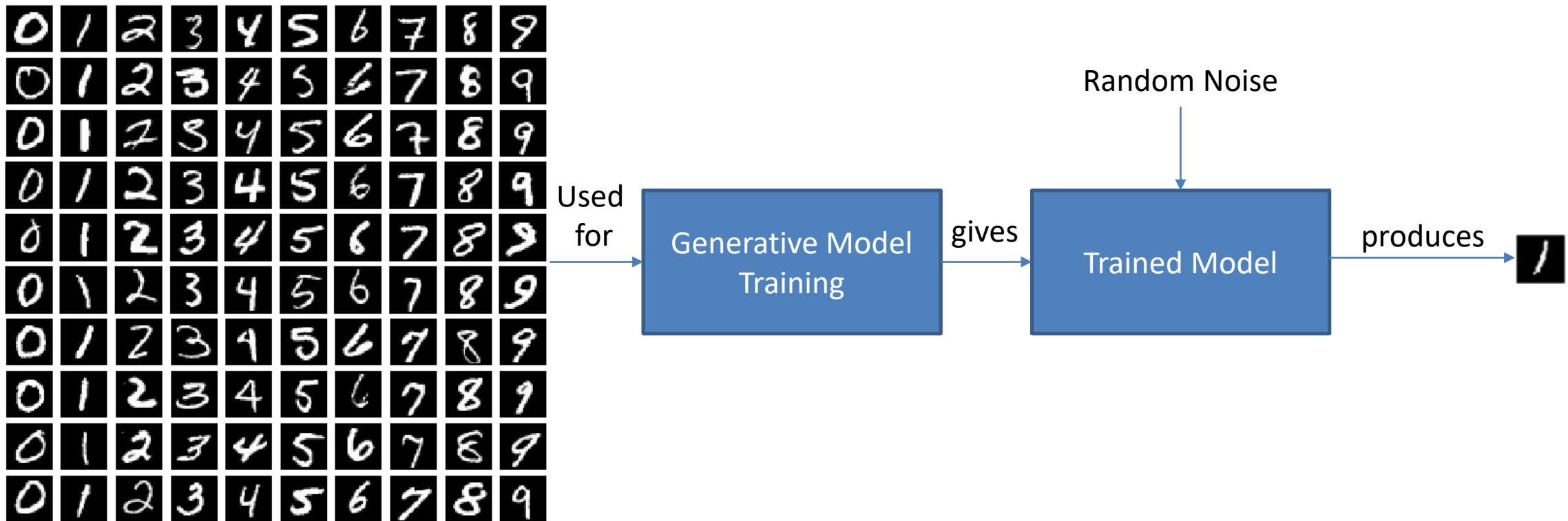
How to go from generating coin flips to images?

- Assume you are given a bunch of B&W images for training that you would like to use to train a GAN to generate more images like that.
- Let's look at a single pixel location in each image
 - We have a distribution of pixel values across all images at that location
 - We would like our GAN to generate data according to that distribution at that pixel location
 - Naïve idea: Have multiple GANs – one for each pixel location
 - Assumes each pixel is independent of the other
 - Computationally intensive
 - We can train a single GAN to generate a multi-dimensional probability distribution by using a multi-output generator.



Unconditional vs Conditional Generation

- Unconditional Generative Modelling
 - Simple model the probability distribution of the data $p(x)$
 - Example: Generating images without paying any regard to the digit

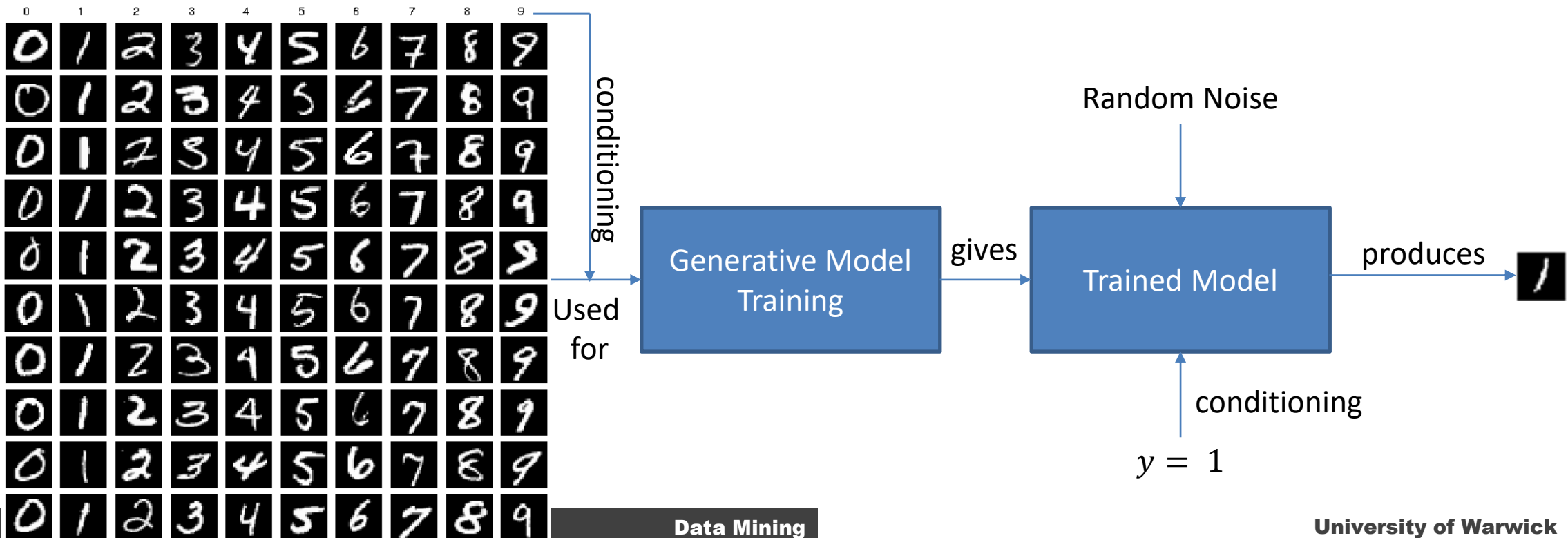


Unconditional vs Conditional Generation

- Conditional Generative Modelling

- Simple model the probability distribution $p(x|y)$ of the data x conditioned on a variable y

- Example: Generating images for a given digit



GANs Applications

- GANs have some impressive applications
 - Synthetic Image Generation
 - Speech Generation
 - Image to Image Translation
 - Style Transfer
 - Deep Fakes

Barebones GAN

<https://github.com/foxtrotmike/CS909/blob/master/simpleGAN.ipynb>

Raevskiy, Mikhail. "Write Your First Generative Adversarial Network Model on PyTorch." Medium, August 31, 2020.

<https://medium.com/dev-genius/write-your-first-generative-adversarial-network-model-on-pytorch-7dc0c7c892c7>.



Ian Goodfellow. NIPS 2016 Tutorial: Generative Adversarial Networks. <https://arxiv.org/abs/1701.00160>
<https://github.com/eriklindernoren/PyTorch-GAN>
<https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/>
<https://affinelayer.com/pixsrv/>

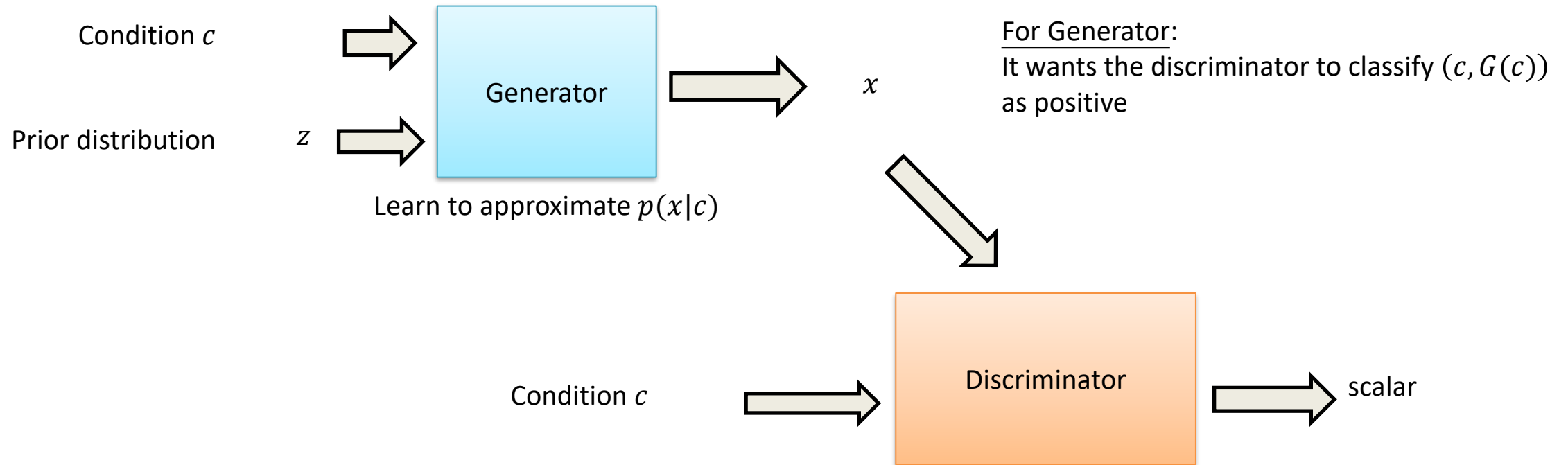
The GAN Zoo

- GAN - Generative Adversarial Networks
- 3D-GAN - Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling
- acGAN - Face Aging With Conditional Generative Adversarial Networks
- AC-GAN - Conditional Image Synthesis With Auxiliary Classifier GANs
- AdaGAN - AdaGAN: Boosting Generative Models
- AEGAN - Learning Inverse Mapping by Autoencoder based Generative Adversarial Nets
- AffGAN - Amortised MAP Inference for Image Super-resolution
- AL-CGAN - Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts
- ALI - Adversarially Learned Inference
- AM-GAN - Generative Adversarial Nets with Labeled Data by Activation Maximization
- AnoGAN - Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery
- ArtGAN - ArtGAN: Artwork Synthesis with Conditional Categorical GANs
- b-GAN - b-GAN: Unified Framework of Generative Adversarial Networks
- Bayesian GAN - Deep and Hierarchical Implicit Models
- BEGAN - BEGAN: Boundary Equilibrium Generative Adversarial Networks
- BiGAN - Adversarial Feature Learning
- BS-GAN - Boundary-Seeking Generative Adversarial Networks
- CGAN - Conditional Generative Adversarial Nets
- CaloGAN - CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks
- CCGAN - Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks
- CatGAN - Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks
- CoGAN - Coupled Generative Adversarial Networks
- Context-RNN-GAN - Contextual RNN-GANs for Abstract Reasoning Diagram Generation
- C-RNN-GAN - C-RNN-GAN: Continuous recurrent neural networks with adversarial training
- CS-GAN - Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets
- CVAE-GAN - CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training
- CycleGAN - Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks
- DTN - Unsupervised Cross-Domain Image Generation
- DCGAN - Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks
- DiscoGAN - Learning to Discover Cross-Domain Relations with Generative Adversarial Networks
- DR-GAN - Disentangled Representation Learning GAN for Pose-Invariant Face Recognition
- DualGAN - DualGAN: Unsupervised Dual Learning for Image-to-Image Translation
- EBGAN - Energy-based Generative Adversarial Network
- f-GAN - f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization
- FF-GAN - Towards Large-Pose Face Frontalization in the Wild
- GAWWN - Learning What and Where to Draw
- GeneGAN - GeneGAN: Learning Object Transfiguration and Attribute Subspace from Unpaired Data
- Geometric GAN - Geometric GAN
- GoGAN - Gang of GANs: Generative Adversarial Networks with Maximum Margin Ranking
- GP-GAN - GP-GAN: Towards Realistic High-Resolution Image Blending
- IAN - Neural Photo Editing with Introspective Adversarial Networks
- iGAN - Generative Visual Manipulation on the Natural Image Manifold
- IcGAN - Invertible Conditional GANs for image editing
- ID-CGAN - Image De-raining Using a Conditional Generative Adversarial Network
- Improved GAN - Improved Techniques for Training GANs
- InfoGAN - InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets
- LAGAN - Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis
- LAPGAN - Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks

<https://github.com/hindupuravinash/the-gan-zoo>

Conditional GAN

Training data: (c, x) , (condition, desired output), e.g., (text, image)



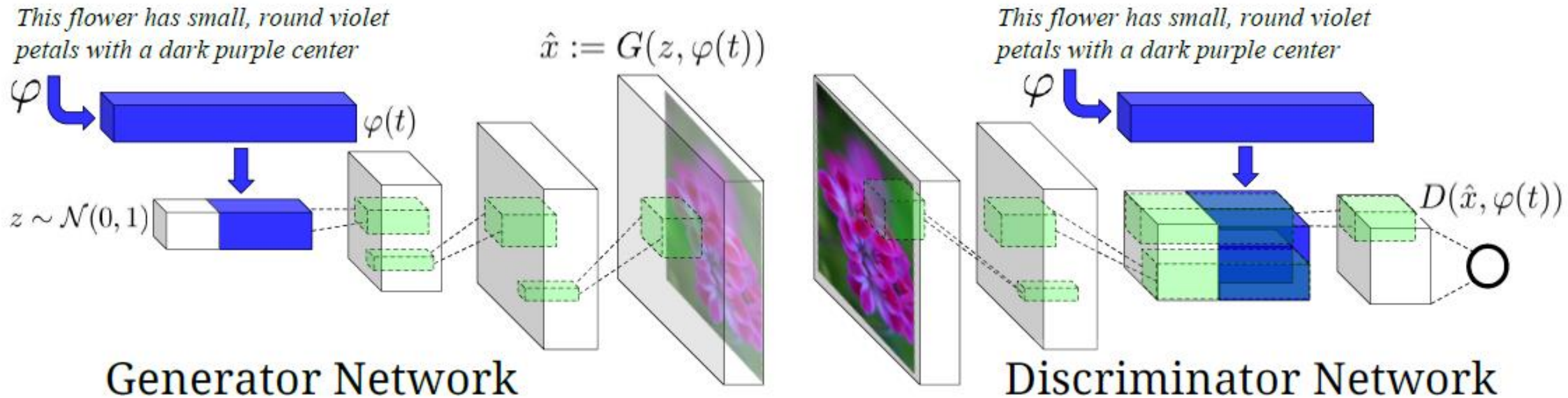
For Discriminator:

Positive example: (c, x) , e.g., the original (text, image) pair

Negative examples: $(c, G(c))$, e.g., (text, generated image) pair

(c', x) , e.g., (arbitrary text, original image) pair

Text-to-Image Synthesis



S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, H. Lee, “Generative Adversarial Text-to-Image Synthesis”, ICML 2016

H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, D. Metaxas, “StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks”, arXiv preprint, 2016

S. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, H. Lee, “Learning What and Where to Draw”, NIPS 2016

Text to Image – Results




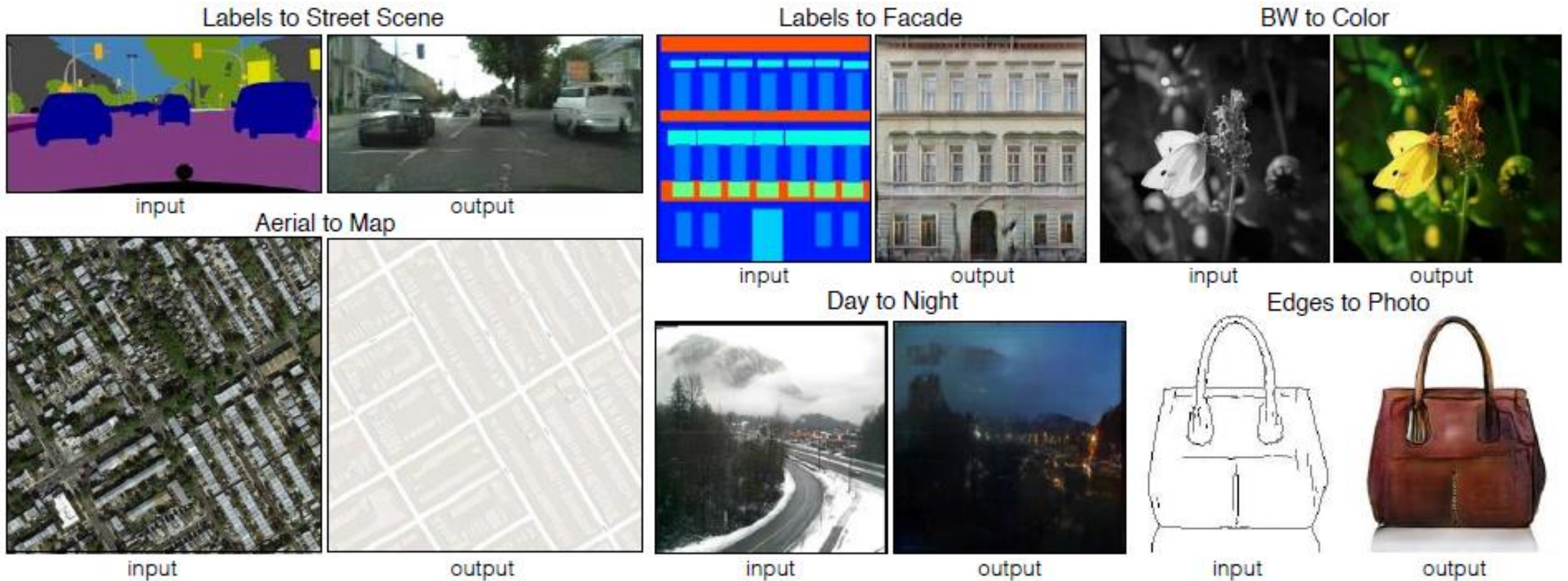
Caption	Image
a pitcher is about to throw the ball to the batter	
a group of people on skis stand in the snow	
a man in a wet suit riding a surfboard on a wave	

Image-to-image Translation



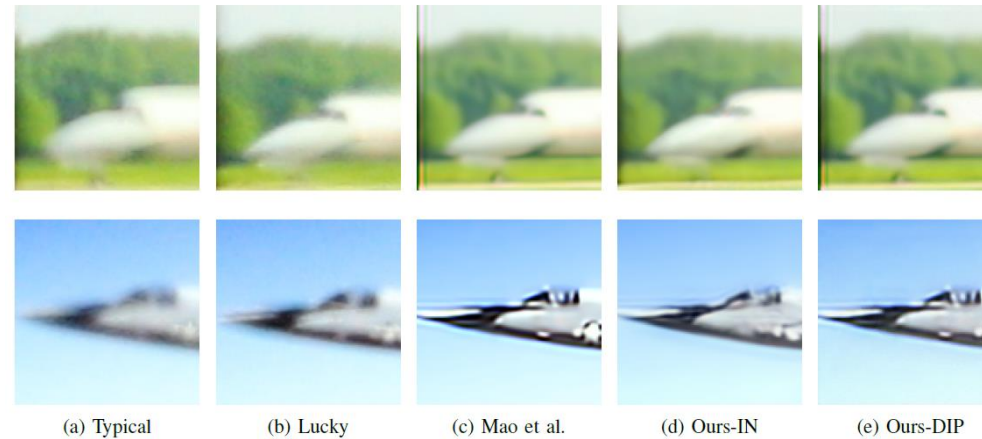
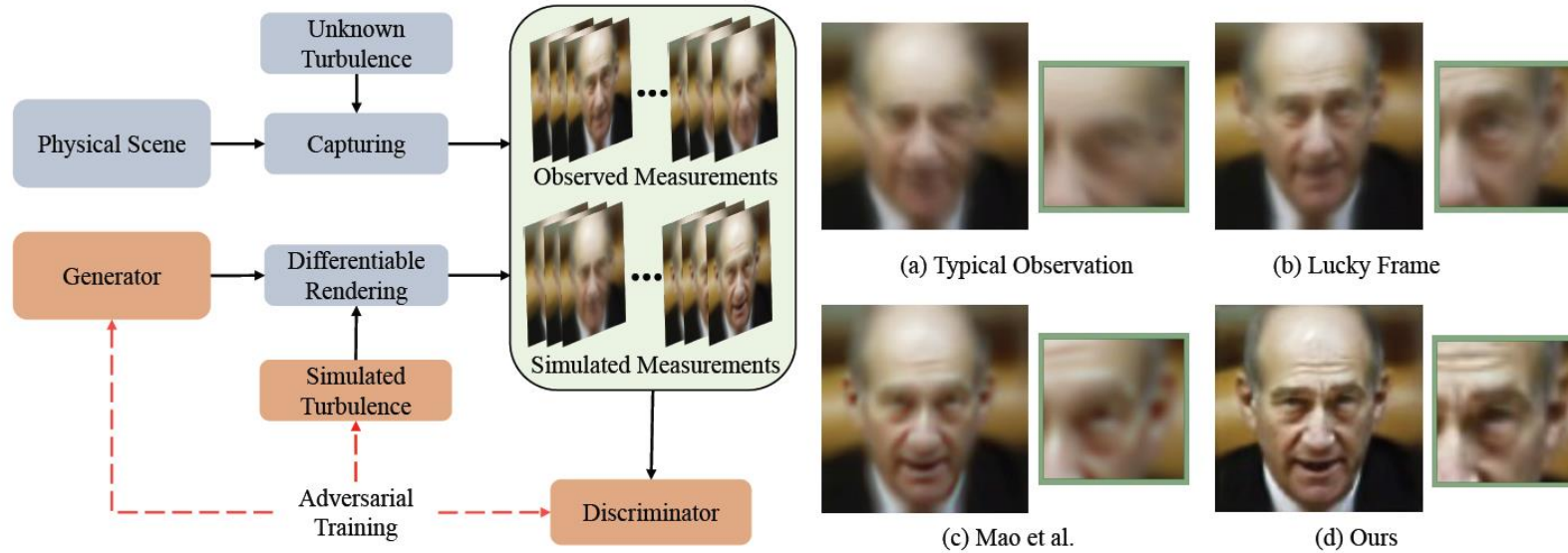
P. Isola, J.-Y. Zhu, T. Zhou, A.A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks", arXiv preprint, 2016

Unpaired Transformation – Cycle GAN, Disco GAN

Transform an object from one domain to another without paired data

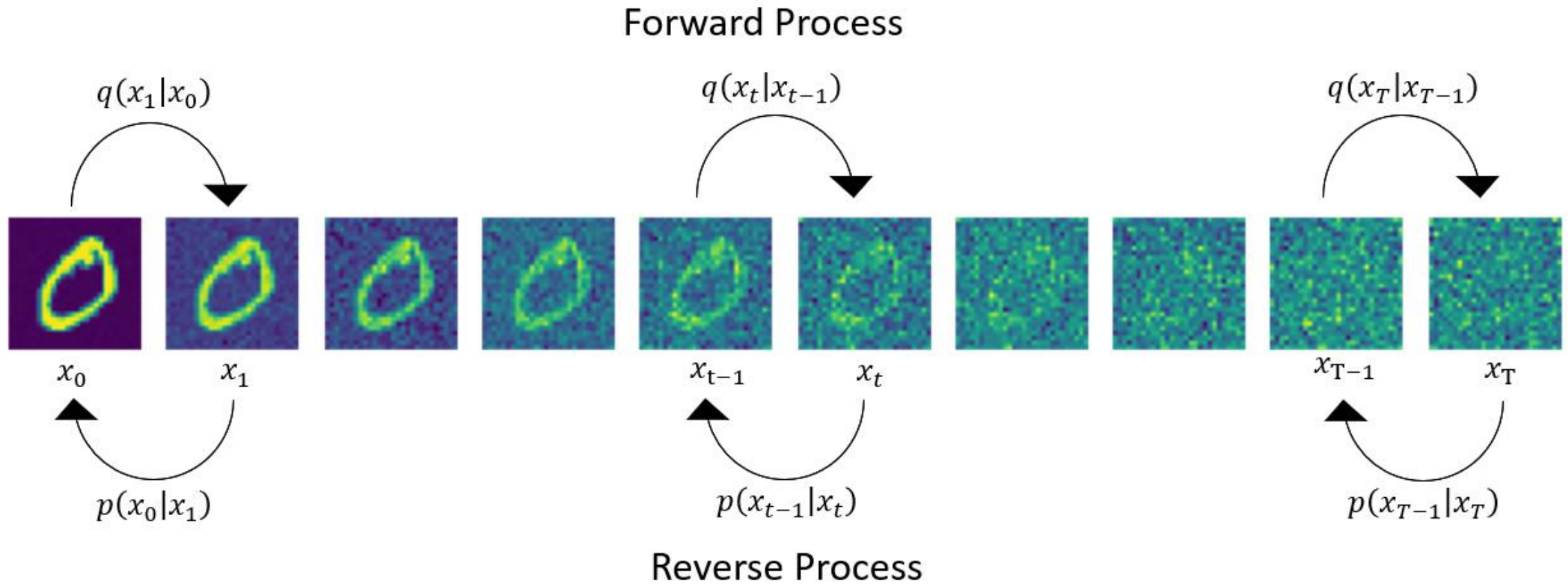


TurbuGAN



Feng, Brandon Yushan, Mingyang Xie, and Christopher A. Metzler. "TurbuGAN: An Adversarial Learning Approach to Spatially-Varying Multiframe Blind Deconvolution with Applications to Imaging Through Turbulence." arXiv, August 22, 2022. <https://doi.org/10.48550/arXiv.2203.06764>.

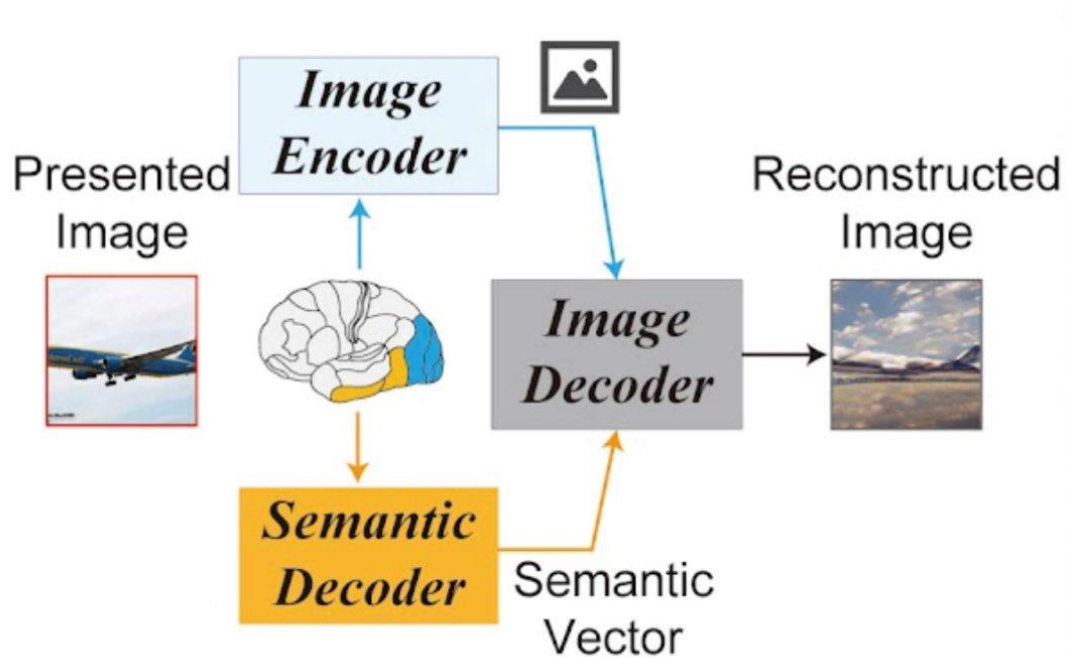
Diffusion Models (Tutorial)



<https://github.com/wrgwrgh/Simple-Diffusion/blob/main/SimpleDiffusion.ipynb>

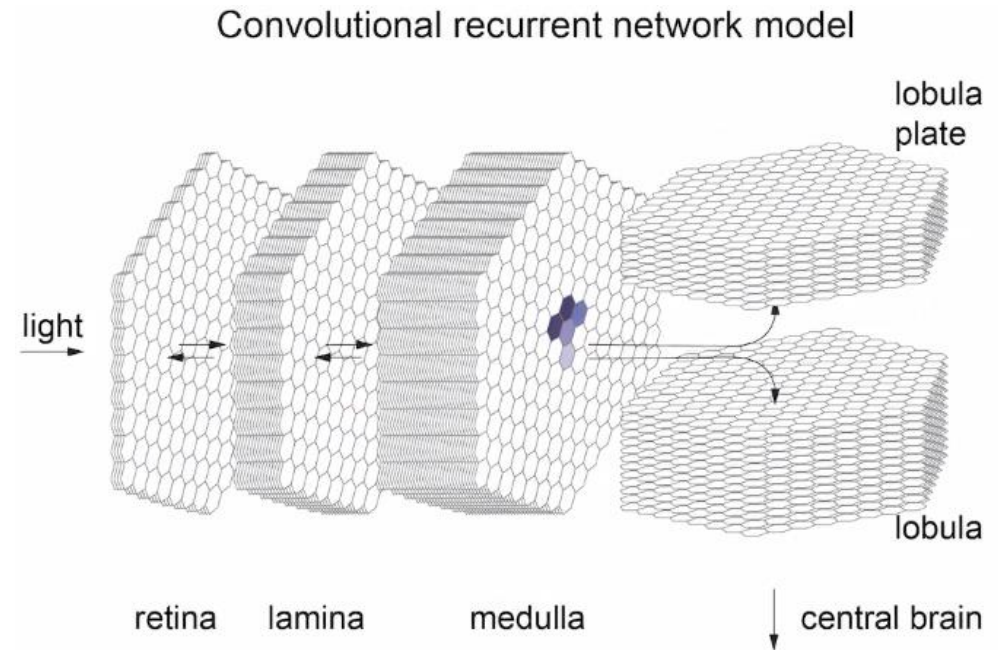
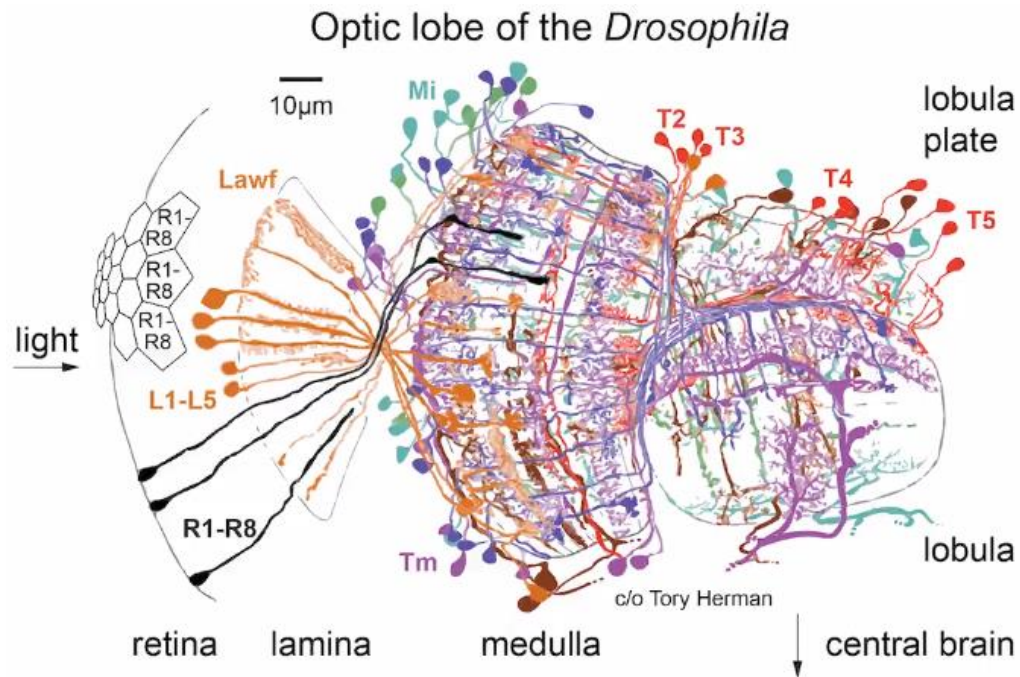
Seeing without seeing

- Takagi, Yu, and Shinji Nishimoto. "High-Resolution Image Reconstruction with Latent Diffusion Models from Human Brain Activity." bioRxiv, December 1, 2022. <https://doi.org/10.1101/2022.11.18.517004>.



Do NNs see like us?

- built a neural network constrained by anatomical connectivity, and showed that with task-training it develops realistic neural activity observed in single cells!



<https://twitter.com/srinituraga/status/1635352596956467209>