CS 6748: Advanced Topics in Machine Learning

Deep Generative Models

Volodymyr Kuleshov

Cornell Tech

URL: https://kuleshov.github.io/cornell-deep-generative-models-course/

Introduction

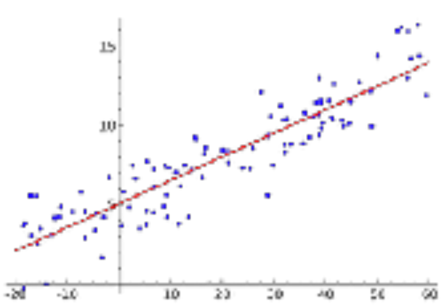
Volodymyr Kuleshov

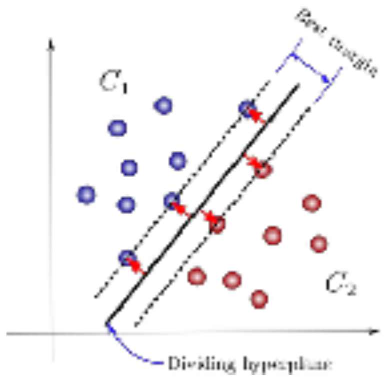
Assistant Professor
Department of Computer Science
Cornell Tech, starting Spring 2020

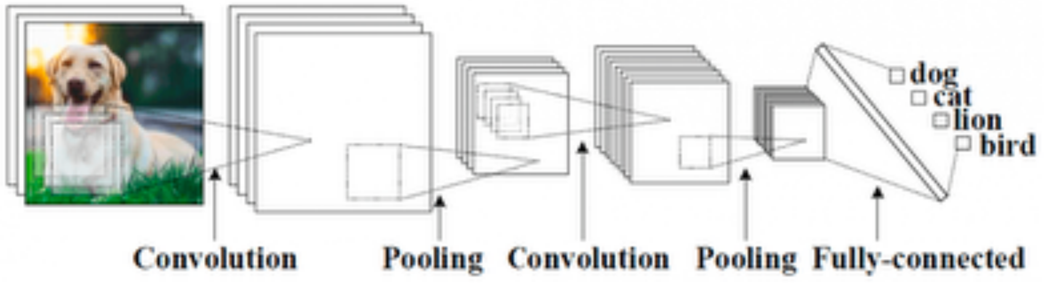
- PhD & Post-Doc from Stanford in 2018. Co-founder Afresh.AI.
- Research in Machine Learning
  - Probabilistic methods, deep learning, reasoning under uncertainty
- Applications in Health, Science, Sustainability
  - Cheap genomics assays with ML, machine reading of the scientific literature, fighting food waste using AI

Introduction

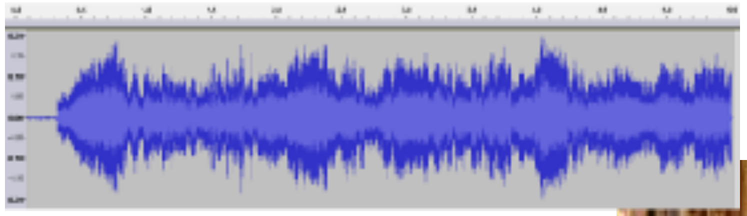
Generative modeling is the study of machine learning models over complex, high-dimensional inputs.







Discriminative Models



Speech

Language

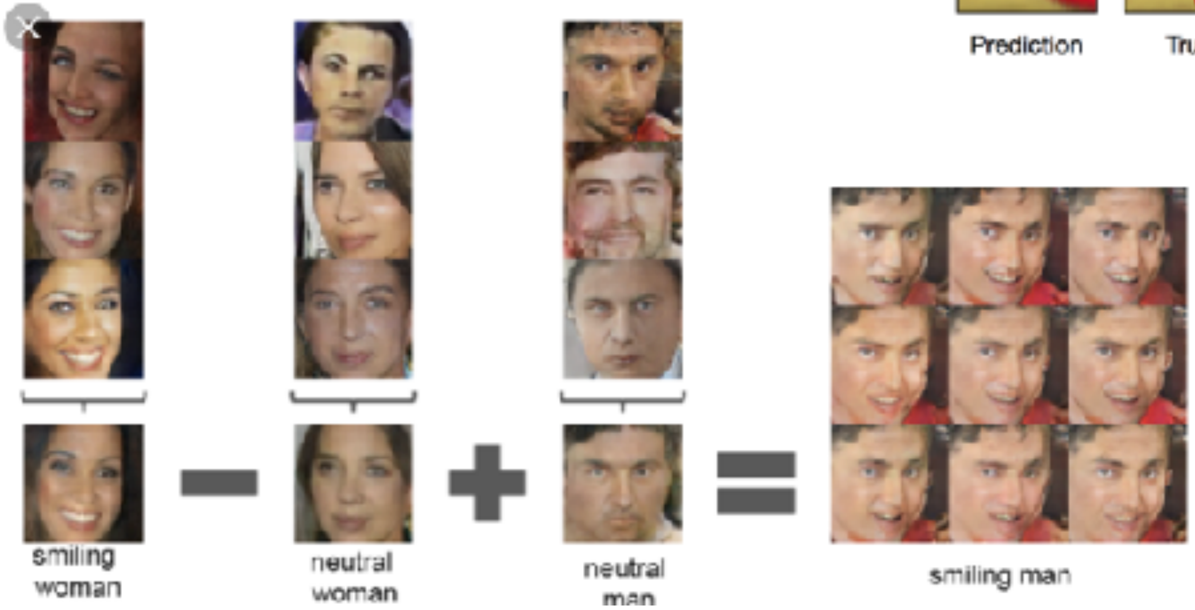
Images

Generative Models

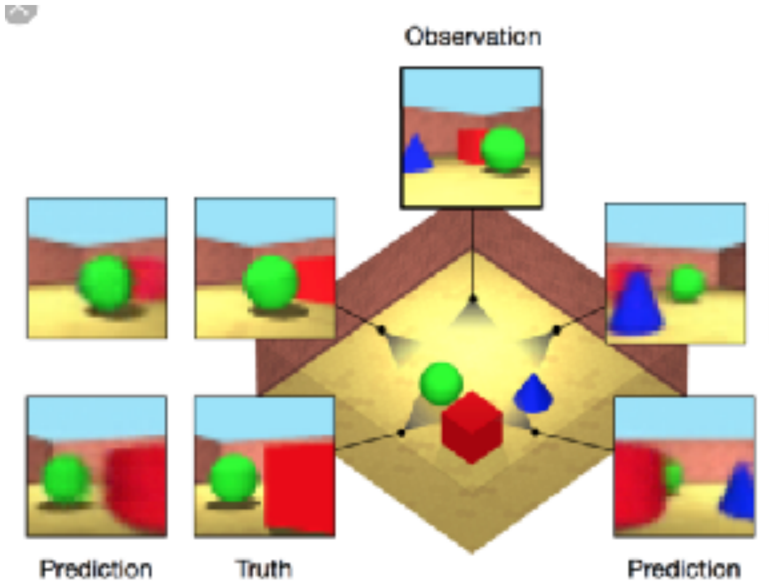
Introduction

What are generative models useful for?

Generation



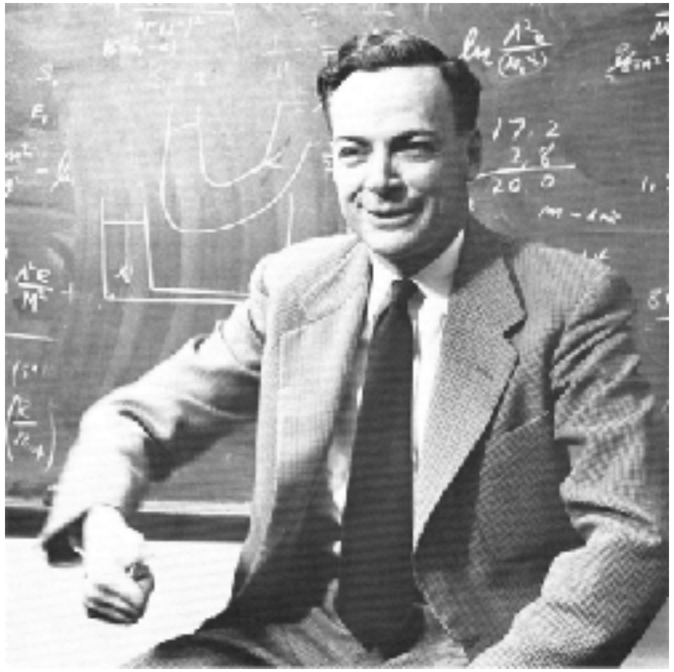
Representation Learning



Decision Making / RL

Introduction

Generative modeling is a step towards building artificially intelligent systems



Richard Feynman: “*What I cannot create, I do not understand*”

An Analogy to Computer Graphics and Vision

Computer graphics is one way to do generative modeling.

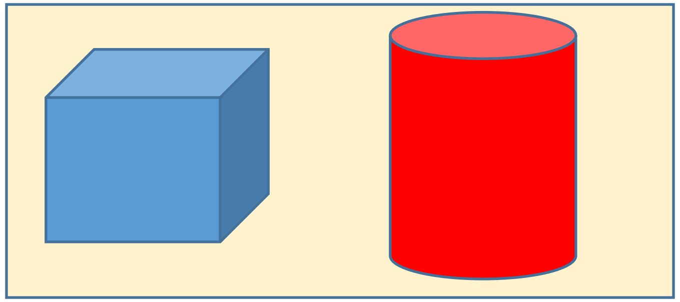
Formal description

Cube(color=blue, position=, size=, …)
Cylinder(color=red, position=, size=,..)

Generation (graphics)

Inference (vision)

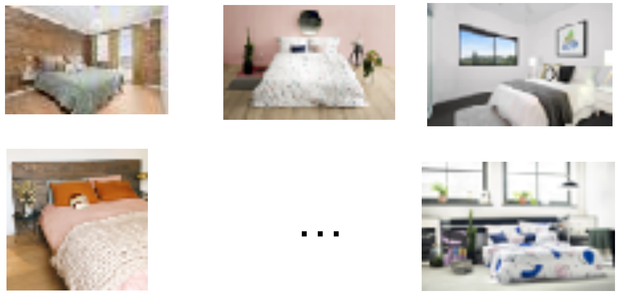
Raw sensory data



Our models will have **similar structure (generation + inference)**

Statistical Generative Models

Statistical generative models are **learned from data**

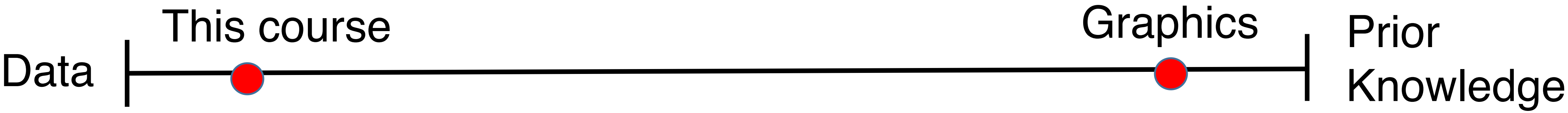


+

Data
(e.g., images of bedrooms)

Prior Knowledge
(e.g., physics, materials, ..)

Priors are always necessary, but there is a spectrum



Statistical Generative Models

A statistical generative model is a **probability distribution** p(x)

- **Data:** samples (e.g., images of bedrooms)
- **Prior knowledge:** parametric form (e.g., Gaussian?), loss function (e.g., maximum likelihood?), optimization algorithm, etc.

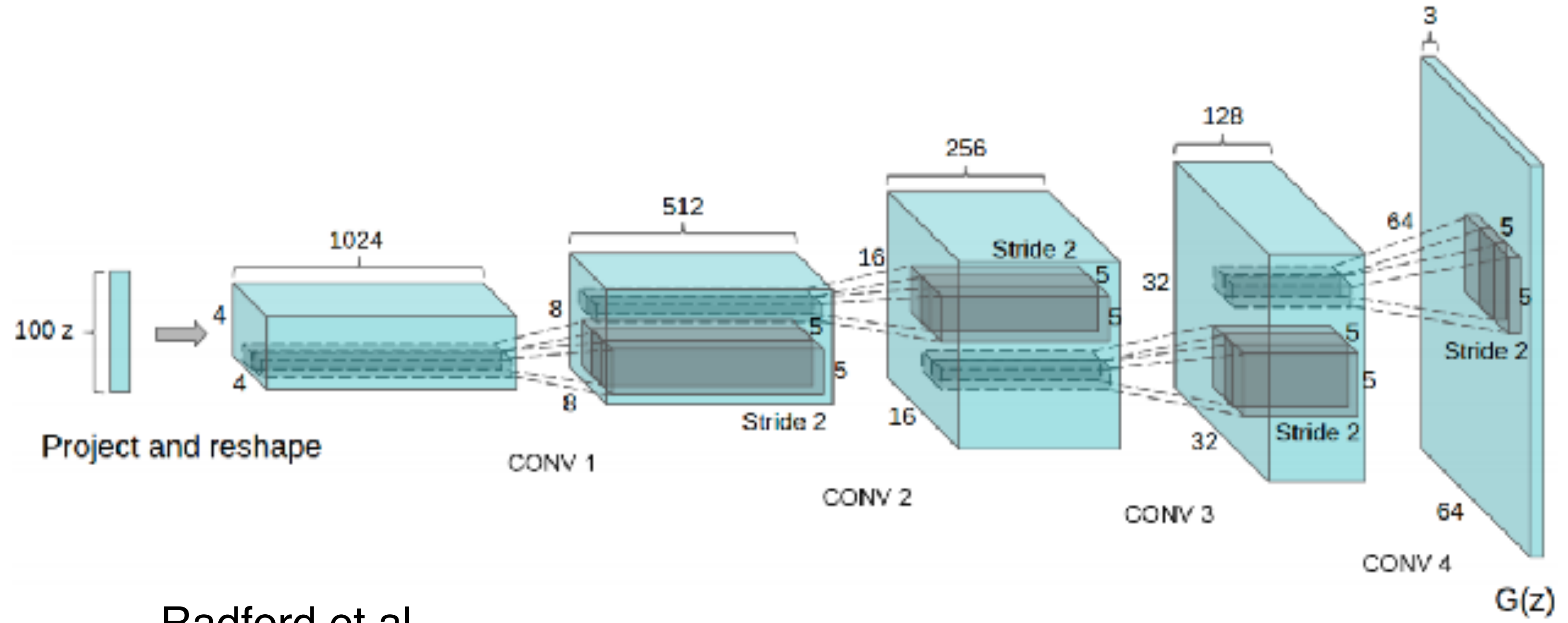
Image x

→ A probability distribution p(x) →

Sampling from p(x) generates new images

Deep Generative Models

The probability distribution p(x) will be specified using **deep neural networks**



Radford et al.

Discriminative vs. Generative Models

Discriminative: classify bedroom vs. dining room

Generative: generate X

Y=B , X=  Y=D , X= 

Y=B , X=  Y=D , X= 

… …

Y=B , X=  Y=D , X= 

The input image X is always given. **Goal**: a good decision boundary, via **conditional distribution**

P(Y = Bedroom | X= ) = 0.0001

Ex: logistic regression, convolutional net, etc.

The input X is **not** given. Requires a model of the **joint distribution**

P(Y = Bedroom , X= ) = 0.0002

Discriminative vs. generative

Joint and conditional are related via **Bayes Rule**:

P(Y = Bedroom | X = ) = P(Y = Bedroom, X= ) / P(X = )

Discriminative: X is always given, does not need to model P(X= )

Therefore it cannot handle missing data P(Y = Bedroom | X = )

Conditional Generative Models

Class **conditional generative models** are also possible:

P(X=  Y = Bedroom)

It's often useful to condition on rich side information Y

P(X=  | Caption = "A black table with 6 chairs")

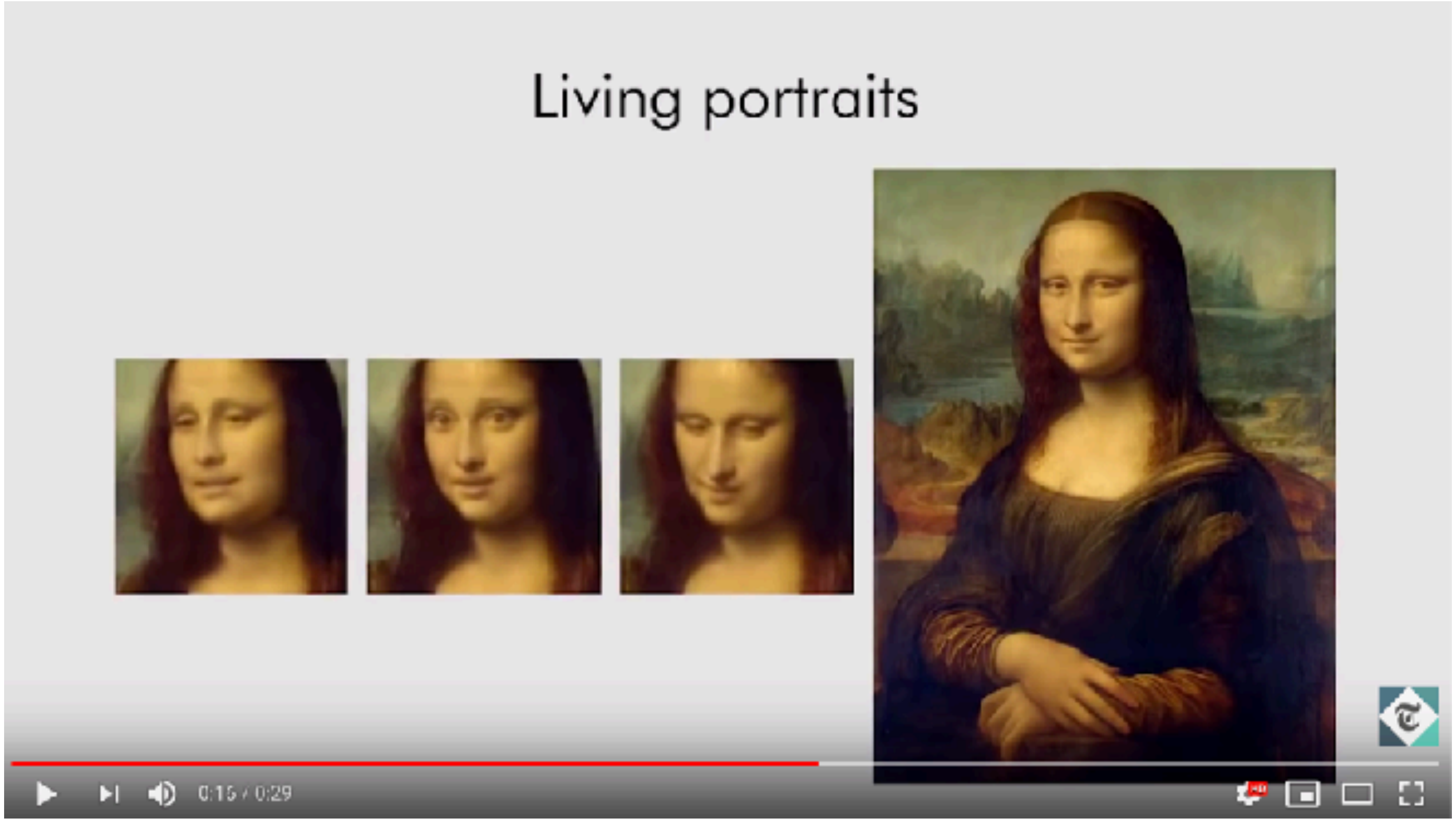
A discriminative model is a very simple conditional generative model of Y:

P(Y = Bedroom | X= )

Progressive Growing of GANs

Karras et al., 2018

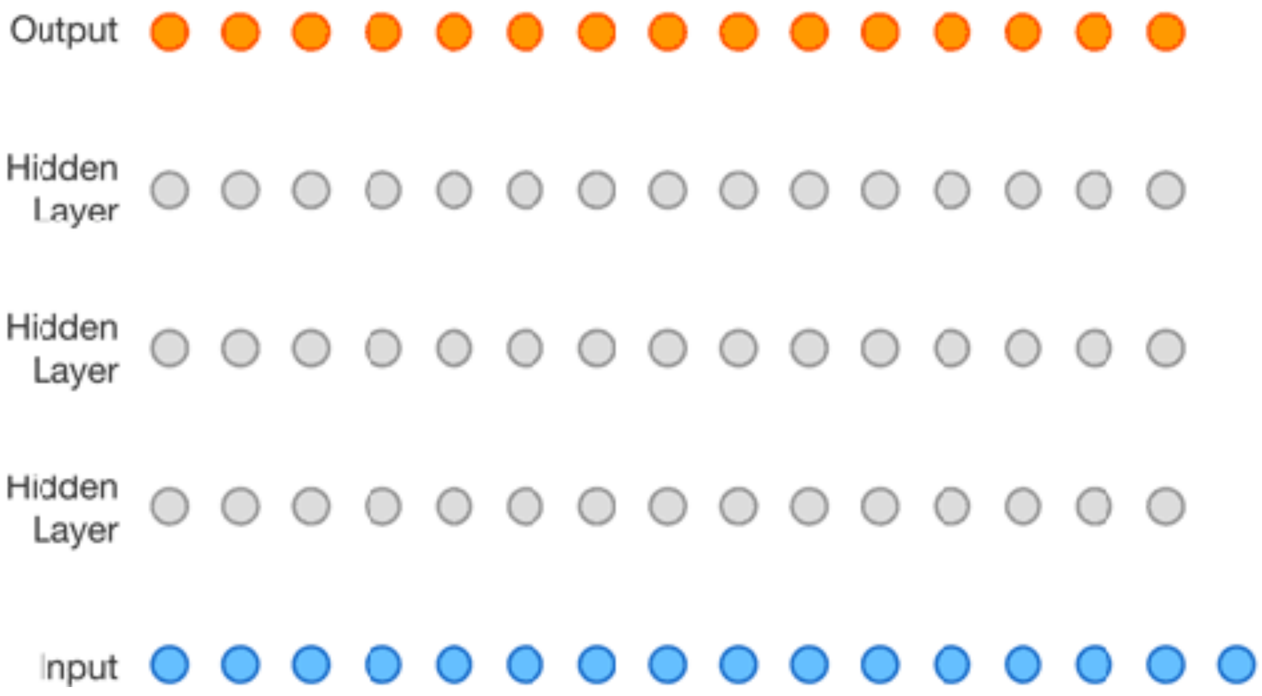
Living Portraits



https://www.youtube.com/watch?v=P2uZF-5F1wI

WaveNet

Generative model of speech signals



Text to Speech

Parametric

Concatenative

WaveNet

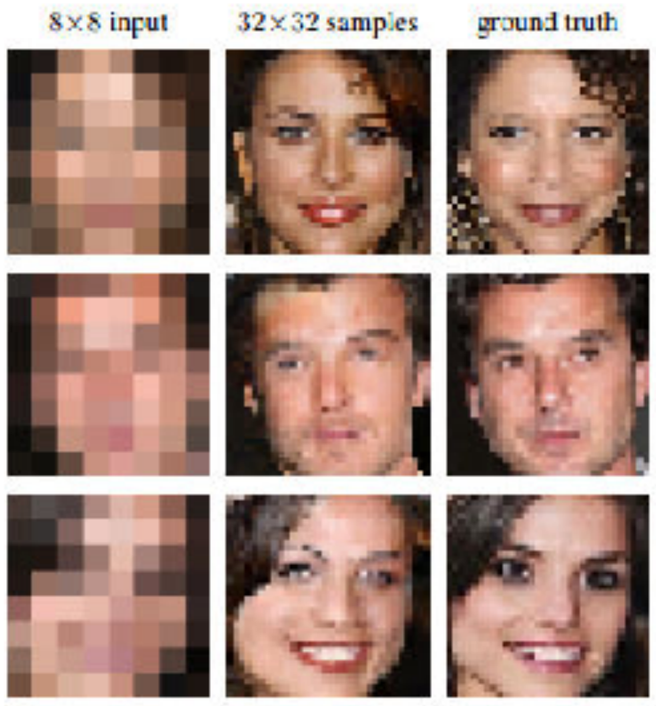
Unconditional

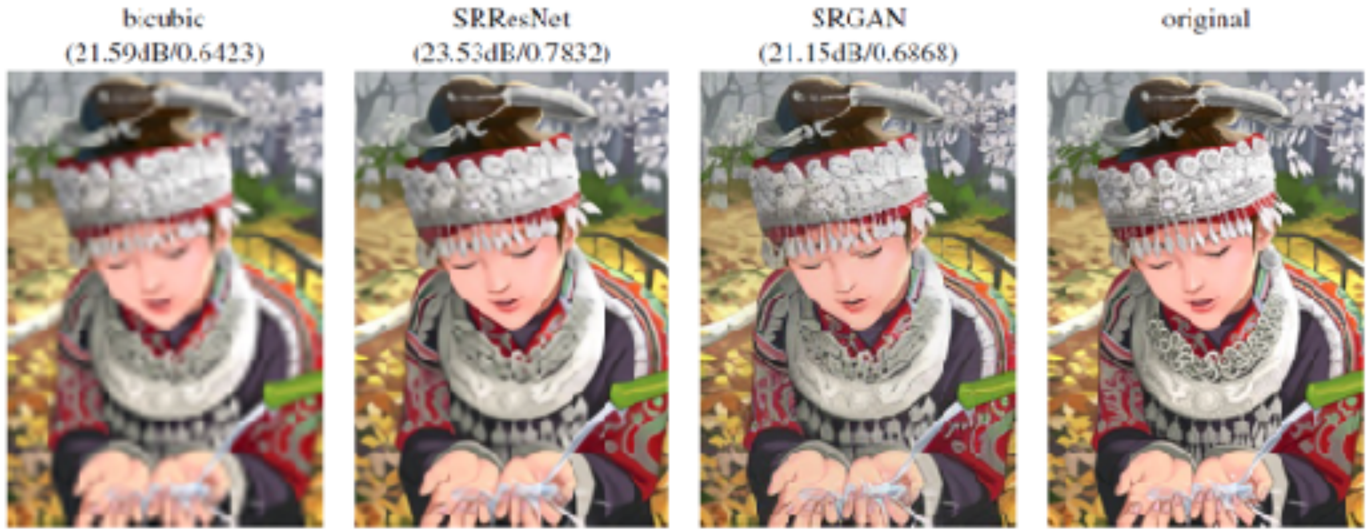
Music

van den Oord et al, 2016c

Image Super Resolution

Conditional generative model P(high res image | low res image)

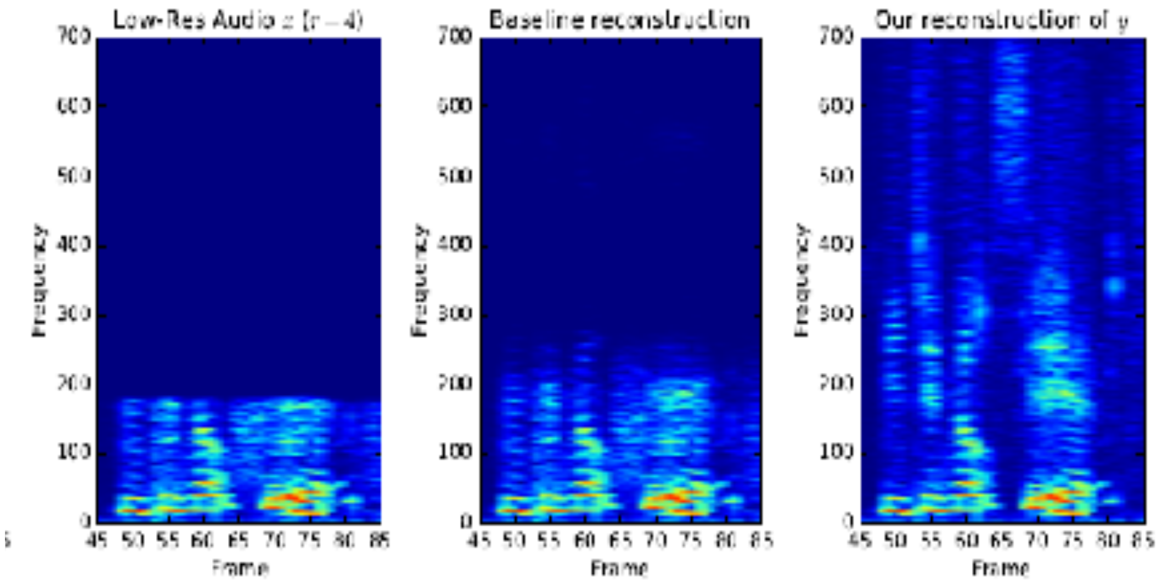




Ledig et al., 2017

Audio Super Resolution

Conditional generative model P(high-res signal I low-res audio signal)



Low res signal

High res audio signal

Kuleshov et al., 2017

Machine Translation

Conditional generative model P(English text | Chinese text)

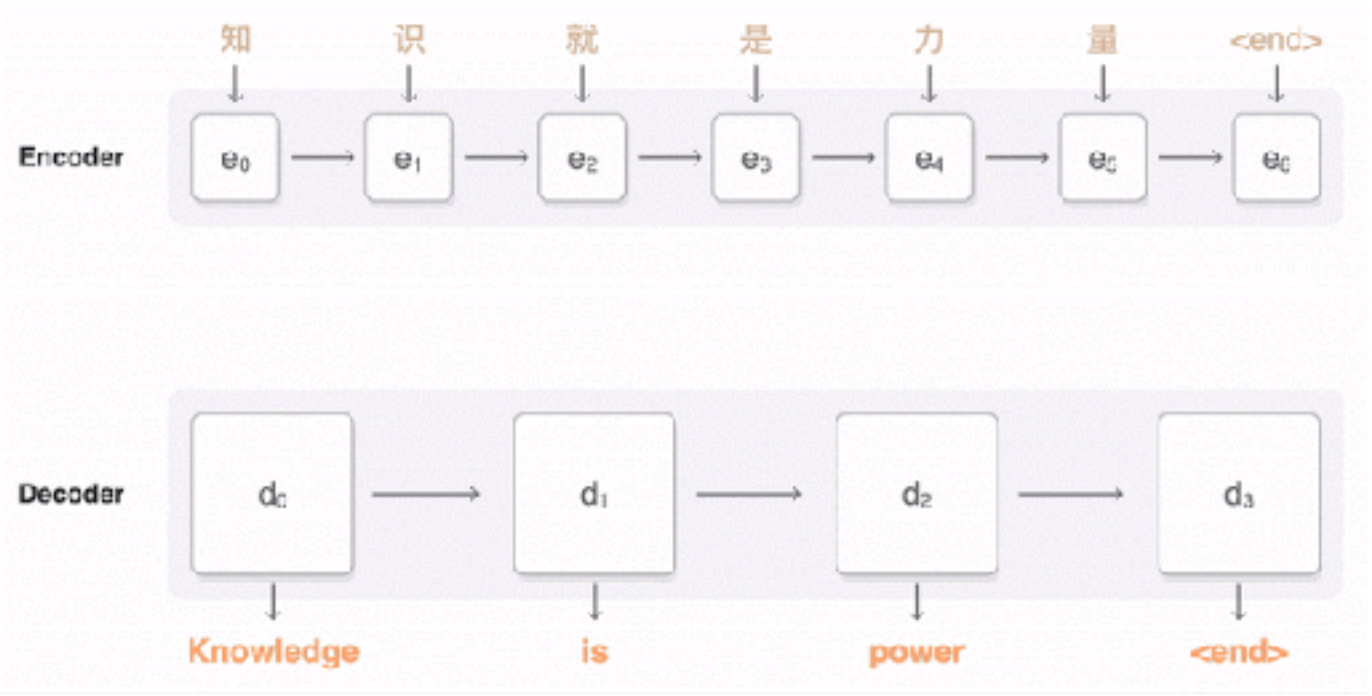


Figure from Google AI research blog.

Language Generation

Custom prompt

To get an A+ in deep generative models, students have to

Completion

To get an A+ in deep generative models, students have to be willing to work with problems that are a whole lot more interesting than, say, the ones that most students work on in class. If you're a great student, the question above can be avoided and you'll be able to do great work, but if you're not, you will need to go beyond the basics before getting good.

Now to be clear, this advice is not just for the deep-learning crowd; it is good advice for any student who is taking his or her first course in machine learning.

The key point is that if you have a deep, deep brain of a computer scientist, that's just as important to you.

Radford et al., 2019
Demo from talktotransformer.com

Image Translation

Conditional generative model P(zebra images | horse images)

Zhu et al., 2017

Image Translation

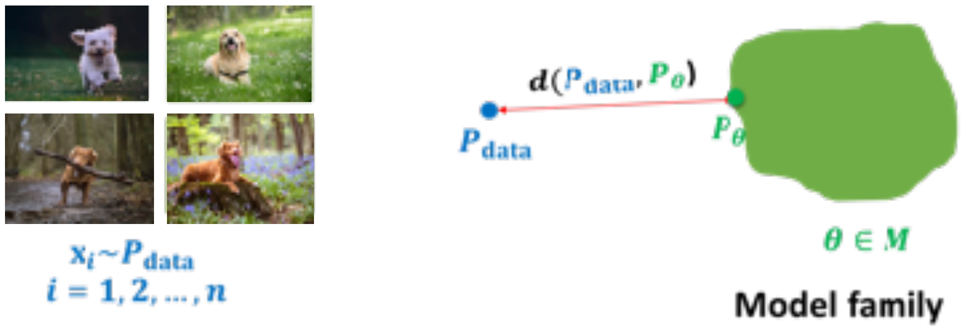
Conditional generative model P(zebra images I horse images)

Image Translation

Conditional generative model P(cat images | dog images)

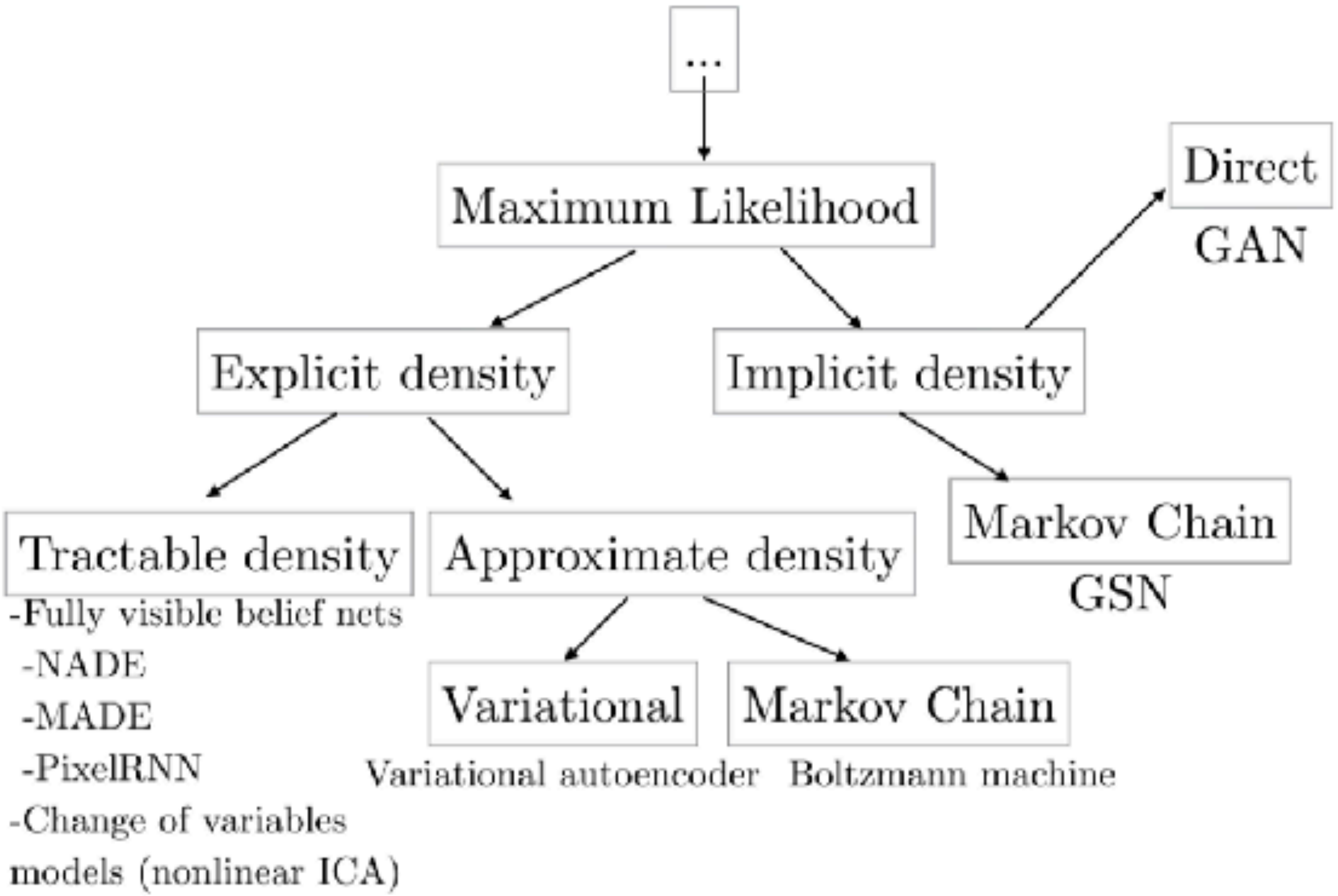
Roadmap and Key Challenges

- **Representation**: how do we model the joint distribution of many random variables?
  - Need compact representation
- **Learning**: what is the right way to compare probability distributions?



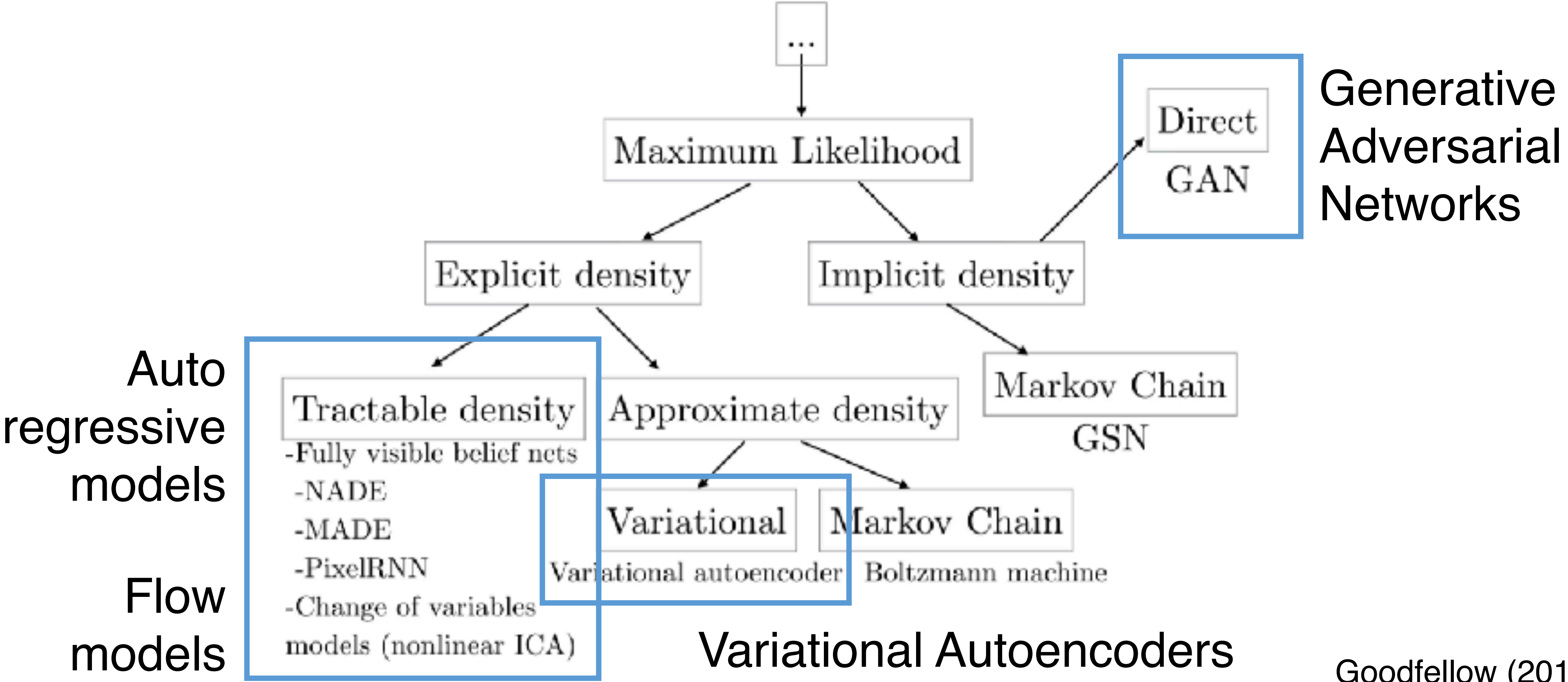
- **Inference**: how do we invert the generation process (e.g., vision as inverse graphics)?
  - Unsupervised learning: recover high-level descriptions (features) from raw data

Syllabus



Goodfellow (2016)

Syllabus



Goodfellow (2016)

Prerequisites

- Basic knowledge about machine learning from at least one of CS4780, CS4701, CS5785
- Basic knowledge of probabilities and calculus:
  - Gradients, gradient-descent optimization, backpropagation
  - Random variables, independence, conditional independence
  - Bayes rule, chain rule, change of variables formulas
- Basic knowledge of deep neural networks (CNNs, RNNs; CS5787).
- Proficiency in some programming language, preferably Python, required.

Logistics

- Class webpage: https://kuleshov.github.io/cornell-deep-generative-models-course/index.html
- There is no required textbook. Reading materials and course notes will be provided.
- Suggested Reading: *Deep Learning* by Ian Goodfellow, Yoshua Bengio, Aaron Courville. Online version available free here.
- Lecture notes: https://deepgenerativemodels.github.io/notes/index.html
- Office hours: After class. See website for more details.

Grading

- Three homeworks (15% each): mix of conceptual and programming based questions
- Student Presentations: 15%
- Course Project: 40%
  - Proposal: 5%
  - Progress Report: 10%
  - Final Report: 25%

Presentations

- Presentations can be done individually or in groups of two.
- Should cover 1-2 research papers out of a list posted on the course website (“Additional readings” on the syllabus page)
- Expected length is 1 hour, followed by 30min discussion
- Email the instructor to reserve presentation slots.
    - Presentations held in class from 03/17-04/23.
- Requirements
    - Send presentation outline at least two weeks before talk
    - Send presentation slides at least two days before talk

Projects

- Course projects will be done in groups of up to 3 students and can fall into one or more of the following categories:
  - Application of deep generative models on a novel task/dataset
  - Algorithmic improvements into the evaluation, learning and/or inference of deep generative models
  - Theoretical analysis of any aspect of existing deep generative models
- Teaching staff will suggest possible projects