CSCI 699 - ProbGen

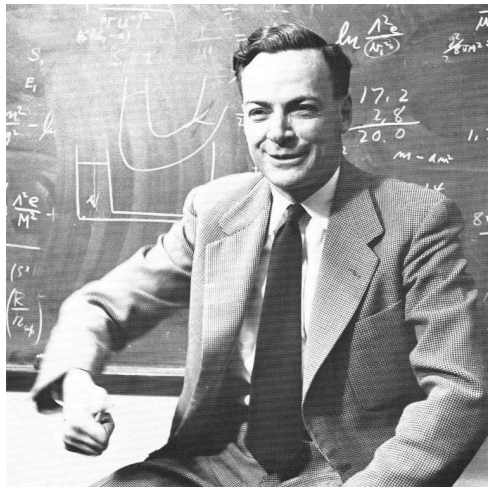
Probabilistic and Generative Models

Willie Neiswanger

Lecture 1 - Introduction

Generative Models

Richard Feynman

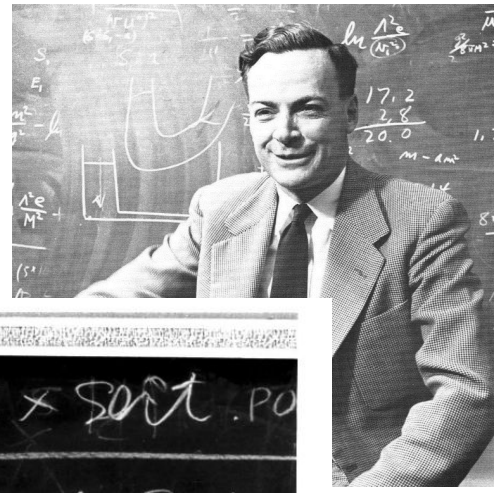


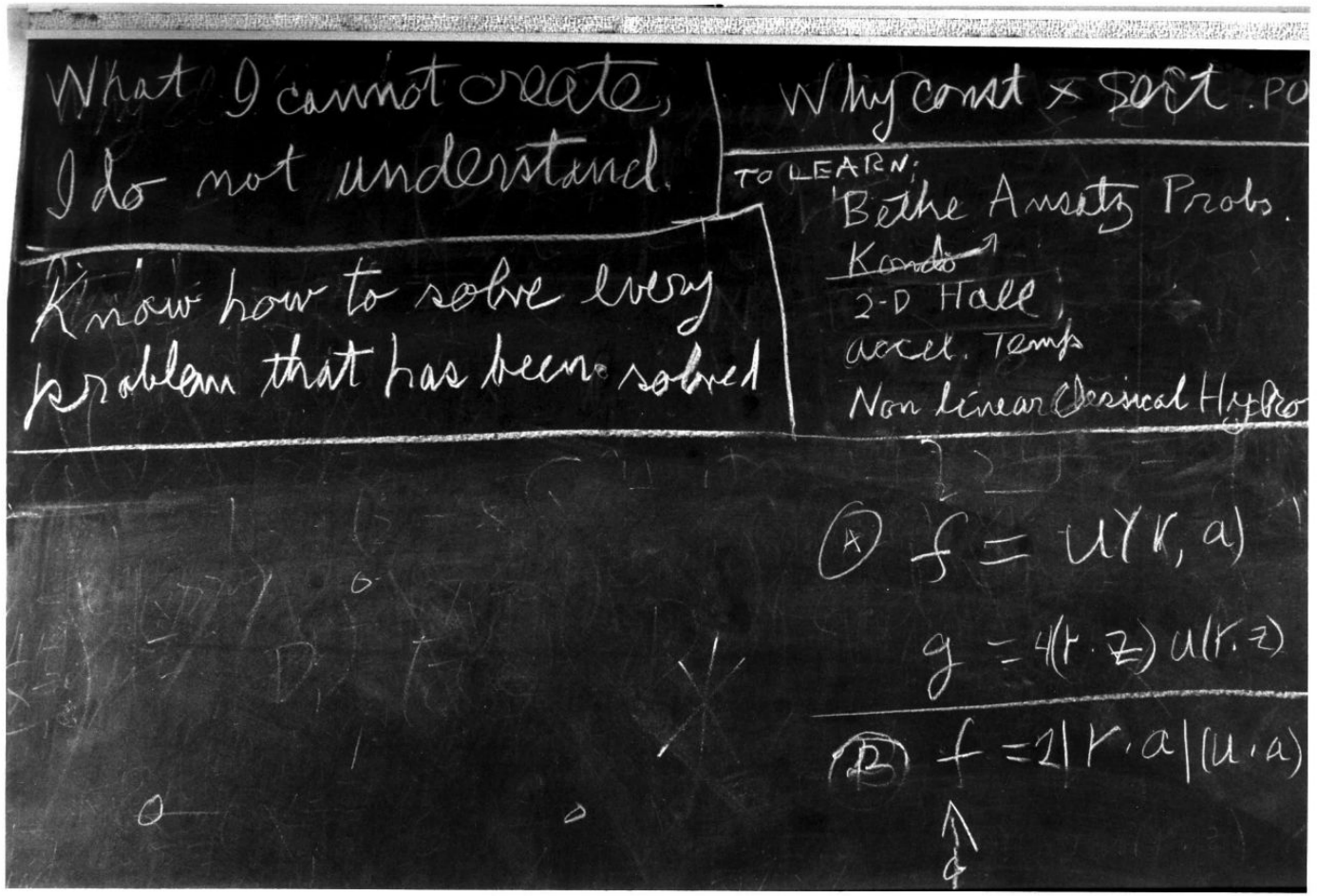
Source: Stefano Ermon, Deep Generative Models Course

(famous physicist)

Generative Models

Richard Feynman's blackboard at the time of his death.

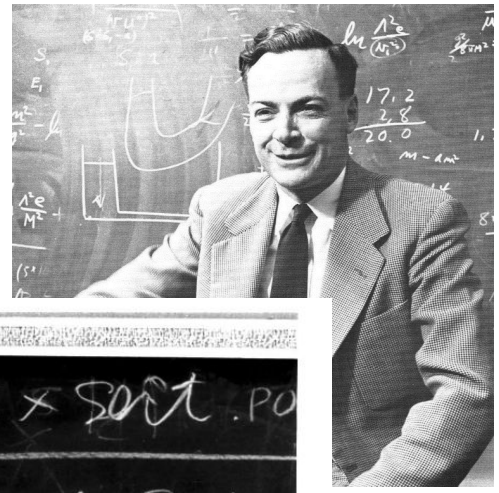


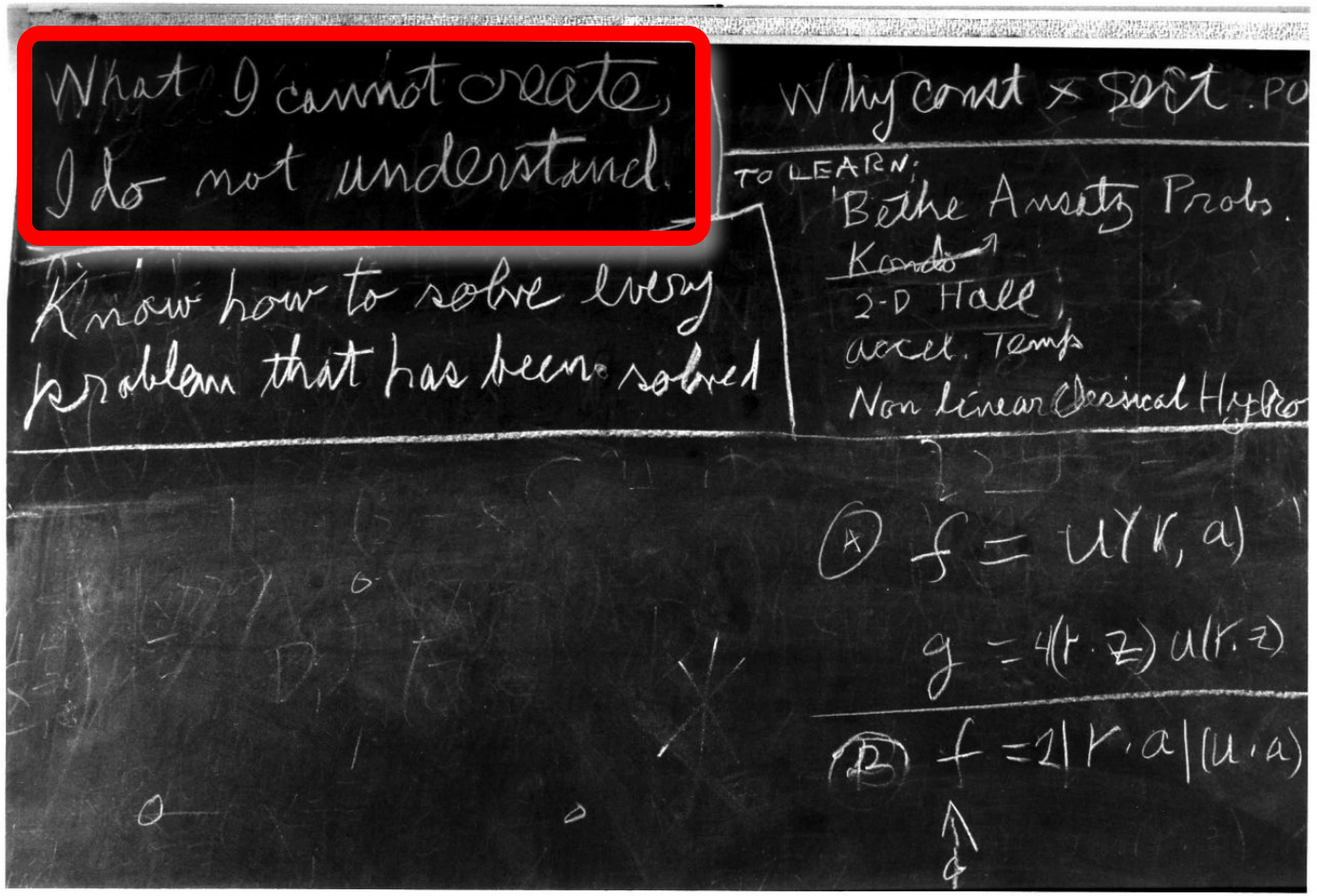


n, Deep
ourse

Generative Models

Richard Feynman's blackboard at the time of his death.

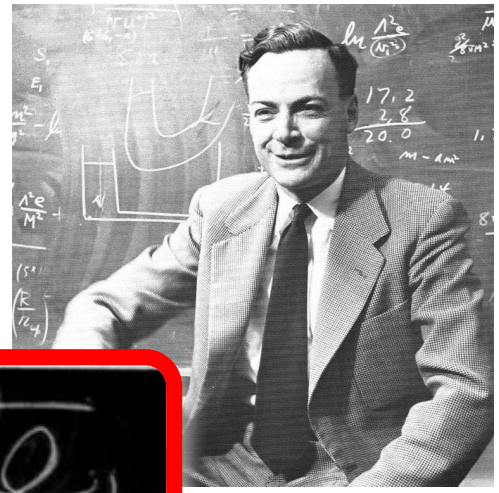


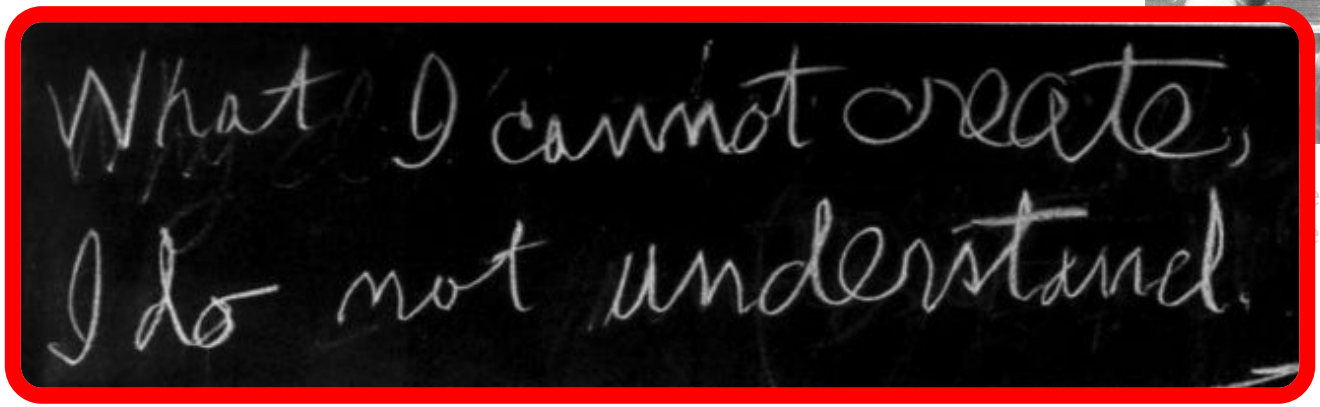


n, Deep
ourse

Generative Models

Richard Feynman's blackboard at the time of his death.





fano Ermon, Deep
Models Course

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

⇒ *Meaning:* to understand a thing, is to be able to create that thing.

Generative Models

And this is very much the mantra of **Generative Modeling in AI**:

⇒ Producing intelligence (*understanding*) via the ability to generate (*create*) things.

Generative Models

And this is very much the mantra of **Generative Modeling in AI**:

⇒ Producing intelligence (*understanding*) via the ability to generate (*create*) things.

Intuitively, producing intelligence via:

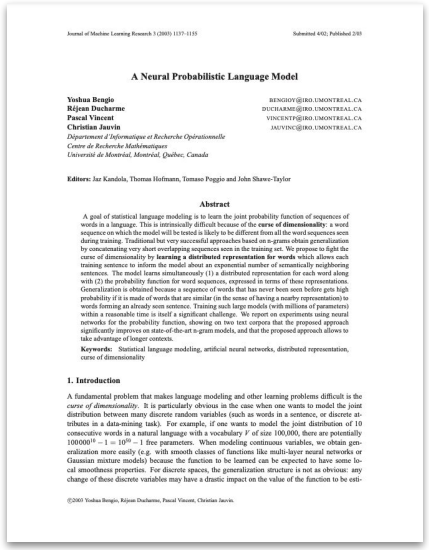
- Learning representations that provide the ability to generate realistic data from the world around us!
- Self-supervised learning.
- Next-token generation.
- Reconstruction of data.

Generative Models

Generative models are taking the world by storm!

Generative Models → Probabilistic Models

These models arise from a long history of probabilistic models in machine learning.



A Neural Probabilistic Language Model

Yoshua Bengio BENGIOY@IRO.UMONTREAL.CA
Réjean Ducharme DUCHARME@IRO.UMONTREAL.CA
Pascal Vincent VINCENTP@IRO.UMONTREAL.CA
Christian Jauvin JAUVINC@IRO.UMONTREAL.CA
Département d'Informatique et Recherche Opérationnelle
Centre de Recherche Mathématiques
Université de Montréal, Montréal, Québec, Canada

Editors: Jaz Kandola, Thomas Hofmann, Tomaso Poggio and John Shawe-Taylor

Abstract

A goal of statistical language modeling is to learn the joint probability function of sequences of words in a language. This is intrinsically difficult because of the **curse of dimensionality**: a word sequence on which the model will be tested is likely to be different from all the word sequences seen during training. Traditional but very successful approaches based on n-grams obtain generalization by concatenating very short overlapping sequences seen in the training set. We propose to fight the curse of dimensionality by **learning a distributed representation for words** which allows each training sentence to inform the model about an exponential number of semantically neighboring sentences. The model learns simultaneously (1) a distributed representation for each word along with (2) the probability function for word sequences, expressed in terms of these representations. Generalization is obtained because a sequence of words that has never been seen before gets high probability if it is made of words that are similar (in the sense of having a nearby representation) to words forming an already seen sentence. Training such large models (with millions of parameters) within a reasonable time is itself a significant challenge. We report on experiments using neural networks for the probability function, showing on two text corpora that the proposed approach significantly improves on state-of-the-art n-gram models, and that the proposed approach allows to take advantage of longer contexts.

Keywords: Statistical language modeling, artificial neural networks, distributed representation, curse of dimensionality

1. Introduction

A fundamental problem that makes language modeling and other learning problems difficult is the *curse of dimensionality*. It is particularly obvious in the case when one wants to model the joint distribution between many discrete random variables (such as words in a sentence, or discrete attributes in a data-mining task). For example, if one wants to model the joint distribution of 10 consecutive words in a natural language with a vocabulary V of size 100,000, there are potentially $100000^{10} - 1 = 10^{50} - 1$ free parameters. When modeling continuous variables, we obtain generalization more easily (e.g. with smooth classes of functions like multi-layer neural networks or Gaussian mixture models) because the function to be learned can be expected to have some local smoothness properties. For discrete spaces, the generalization structure is not as obvious: any change of these discrete variables may have a drastic impact on the value of the function to be esti-

Autoregressive LLMs, 2003

Estimation of Non-Normalized Statistical Models by Score Matching

Aapo Hyvärinen AAPO.HYVARINEN@HELSINKI.FI
Helsinki Institute for Information Technology (HIIT)
Department of Computer Science
FIN-00014 University of Helsinki, Finland

Editor: Peter Dayan

Abstract

One often wants to estimate statistical models where the probability density function is known only up to a multiplicative normalization constant. Typically, one then has to resort to Markov Chain Monte Carlo methods, or approximations of the normalization constant. Here, we propose that such models can be estimated by minimizing the expected squared distance between the gradient of the log-density given by the model and the gradient of the log-density of the observed data. While the estimation of the gradient of log-density function is, in principle, a very difficult non-parametric problem, we prove a surprising result that gives a simple formula for this objective function. The density function of the observed data does not appear in this formula, which simplifies to a sample average of a sum of some derivatives of the log-density given by the model. The validity of the method is demonstrated on multivariate Gaussian and independent component analysis models, and by estimating an overcomplete filter set for natural image data.

Keywords: statistical estimation, non-normalized densities, pseudo-likelihood, Markov chain Monte Carlo, contrastive divergence

1. Introduction

In many cases, probabilistic models in machine learning, statistics, or signal processing are given in the form of non-normalized probability densities. That is, the model contains an unknown normalization constant whose computation is too difficult for practical purposes.

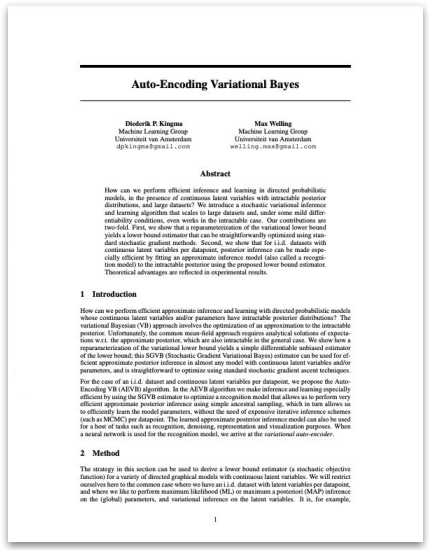
Assume we observe a random vector $\mathbf{x} \in \mathbb{R}^n$ which has a probability density function (pdf) denoted by $p_{\mathbf{x}}(.)$. We have a parametrized density model $p(.;\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is an m-dimensional vector of parameters. We want to estimate the parameter $\boldsymbol{\theta}$ from $\mathbf{x}$, i.e. we want to approximate $p_{\mathbf{x}}(.)$ by $p(.;\boldsymbol{\theta})$ for the estimated parameter value $\hat{\boldsymbol{\theta}}$. (We shall here consider the case of continuous-valued variables only.)

The problem we consider here is that we only are able to compute the pdf given by the model up to a multiplicative constant $Z(\boldsymbol{\theta})$:

$$p(\boldsymbol{\xi};\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} q(\boldsymbol{\xi};\boldsymbol{\theta}).$$

That is, we do know the functional form of q as an analytical expression (or any form that can be easily computed), but we do *not* know how to easily compute Z which is given by

Score Matching, 2005



Auto-Encoding Variational Bayes

Diederik P. Kingma
Machine Learning Group
Universiteit van Amsterdam
dpkingma@gmail.com

Max Welling
Machine Learning Group
Universiteit van Amsterdam
welling.max@gmail.com

Abstract

How can we perform efficient inference and learning in directed probabilistic models, in the presence of continuous latent variables with intractable posterior distributions, and large datasets? We introduce a stochastic variational inference and learning algorithm that scales to large datasets and, under some mild differentiability conditions, even works in the intractable case. Our contributions are two-fold. First, we show that a reparameterization of the variational lower bound yields a lower bound estimator that can be straightforwardly optimized using standard stochastic gradient methods. Second, we show that for i.i.d. datasets with continuous latent variables per datapoint, posterior inference can be made especially efficient by fitting an approximate inference model (also called a recognition model) to the intractable posterior using the proposed lower bound estimator. Theoretical advantages are reflected in experimental results.

1 Introduction

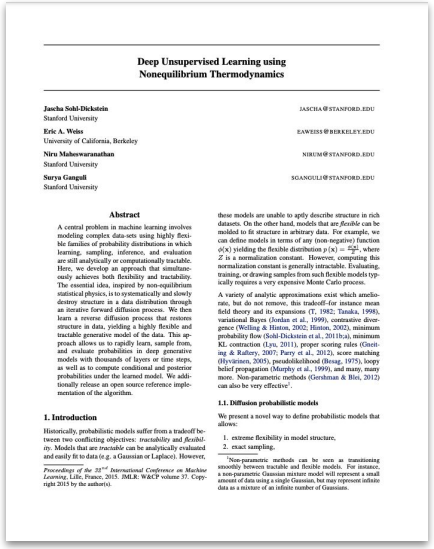
How can we perform efficient approximate inference and learning with directed probabilistic models whose continuous latent variables and/or parameters have intractable posterior distributions? The variational Bayesian (VB) approach involves the optimization of an approximation to the intractable posterior. Unfortunately, the common mean-field approach requires analytical solutions of expectations w.r.t. the approximate posterior, which are also intractable in the general case. We show how a reparameterization of the variational lower bound yields a simple differentiable unbiased estimator of the lower bound; this SGVB (Stochastic Gradient Variational Bayes) estimator can be used for efficient approximate posterior inference in almost any model with continuous latent variables and/or parameters, and is straightforward to optimize using standard stochastic gradient ascent techniques.

For the case of an i.i.d. dataset and continuous latent variables per datapoint, we propose the Auto-Encoding VB (AEVB) algorithm. In the AEVB algorithm we make inference and learning especially efficient by using the SGVB estimator to optimize a recognition model that allows us to perform very efficient approximate posterior inference using simple ancestral sampling, which in turn allows us to efficiently learn the model parameters, without the need of expensive iterative inference schemes (such as MCMC) per datapoint. The learned approximate posterior inference model can also be used for a host of tasks such as recognition, denoising, representation and visualization purposes. When a neural network is used for the recognition model, we arrive at the *variational auto-encoder*.

2 Method

The strategy in this section can be used to derive a lower bound estimator (a stochastic objective function) for a variety of directed graphical models with continuous latent variables. We will restrict ourselves here to the common case where we have an i.i.d. dataset with latent variables per datapoint, and where we like to perform maximum likelihood (ML) or maximum a posteriori (MAP) inference on the (global) parameters, and variational inference on the latent variables. It is, for example,

Variational Autoencoder, 2013



Deep Unsupervised Learning using Nonequilibrium Thermodynamics

Jascha Sohl-Dickstein JASCHA@STANFORD.EDU
Stanford University

Eric A. Weiss EAWEISS@BERKELEY.EDU
University of California, Berkeley

Niru Maheswaranathan NIRUM@STANFORD.EDU
Stanford University

Surya Ganguli SGANGULI@STANFORD.EDU
Stanford University

Abstract

A central problem in machine learning involves modeling complex data-sets using highly flexible families of probability distributions in which learning, sampling, inference, and evaluation are still analytically or computationally tractable. Here, we develop an approach that simultaneously achieves both flexibility and tractability. The essential idea, inspired by non-equilibrium statistical physics, is to systematically and slowly destroy structure in a data distribution through an iterative forward diffusion process. We then learn a reverse diffusion process that restores structure in data, yielding a highly flexible and tractable generative model of the data. This approach allows us to rapidly learn, sample from, and evaluate probabilities in deep generative models with thousands of layers or time steps, as well as to compute conditional and posterior probabilities under the learned model. We additionally release an open source reference implementation of the algorithm.

1. Introduction

Historically, probabilistic models suffer from a tradeoff between two conflicting objectives: *tractability* and *flexibility*. Models that are *tractable* can be analytically evaluated and easily fit to data (e.g. a Gaussian or Laplace). However, these models are unable to aptly describe structure in rich datasets. On the other hand, models that are *flexible* can be molded to fit structure in arbitrary data. For example, we can define models in terms of any (non-negative) function $\phi(\mathbf{x})$ yielding the flexible distribution $p(\mathbf{x}) = \frac{\phi(\mathbf{x})}{Z}$, where Z is a normalization constant. However, computing this normalization constant is generally intractable. Evaluating, training, or drawing samples from such flexible models typically requires a very expensive Monte Carlo process.

A variety of analytic approximations exist which ameliorate, but do not remove, this tradeoff–for instance mean field theory and its expansions (T, 1982; Tanaka, 1998), variational Bayes (Jordan et al., 1999), contrastive divergence (Welling & Hinton, 2002; Hinton, 2002), minimum probability flow (Sohl-Dickstein et al., 2011b;a), minimum KL contraction (Lyu, 2011), proper scoring rules (Gneiting & Raftery, 2007; Parry et al., 2012), score matching (Hyvärinen, 2005), pseudolikelihood (Besag, 1975), loopy belief propagation (Murphy et al., 1999), and many, many more. Non-parametric methods (Gershman & Blei, 2012) can also be very effective[1].

1.1. Diffusion probabilistic models

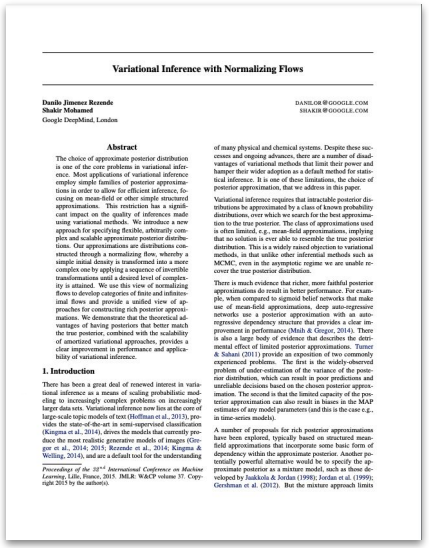
We present a novel way to define probabilistic models that allows:

1. extreme flexibility in model structure,
2. exact sampling,

[1] Non-parametric methods can be seen as transitioning smoothly between tractable and flexible models. For instance, a non-parametric Gaussian mixture model will represent a small amount of data using a single Gaussian, but may represent infinite data as a mixture of an infinite number of Gaussians.

Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

Diffusion Models, 2015



Variational Inference with Normalizing Flows

Danilo Jimenez Rezende DANILOR@GOOGLE.COM
Shakir Mohamed SHAKIR@GOOGLE.COM
Google DeepMind, London

Abstract

The choice of approximate posterior distribution is one of the core problems in variational inference. Most applications of variational inference employ simple families of posterior approximations in order to allow for efficient inference, focusing on mean-field or other simple structured approximations. This restriction has a significant impact on the quality of inferences made using variational methods. We introduce a new approach for specifying flexible, arbitrarily complex and scalable approximate posterior distributions. Our approximations are distributions constructed through a normalizing flow, whereby a simple initial density is transformed into a more complex one by applying a sequence of invertible transformations until a desired level of complexity is attained. We use this view of normalizing flows to develop categories of finite and infinitesimal flows and provide a unified view of approaches for constructing rich posterior approximations. We demonstrate that the theoretical advantages of having posteriors that better match the true posterior, combined with the scalability of amortized variational approaches, provides a clear improvement in performance and applicability of variational inference.

1. Introduction

There has been a great deal of renewed interest in variational inference as a means of scaling probabilistic modeling to increasingly complex problems on increasingly larger data sets. Variational inference now lies at the core of large-scale topic models of text (Hoffman et al., 2013), provides the state-of-the-art in semi-supervised classification (Kingma et al., 2014), drives the models that currently produce the most realistic generative models of images (Gregor et al., 2014; 2015; Rezende et al., 2014; Kingma & Welling, 2014), and are a default tool for the understanding of many physical and chemical systems. Despite these successes and ongoing advances, there are a number of disadvantages of variational methods that limit their power and hamper their wider adoption as a default method for statistical inference. It is one of these limitations, the choice of posterior approximation, that we address in this paper.

Variational inference requires that intractable posterior distributions be approximated by a class of known probability distributions, over which we search for the best approximation to the true posterior. The class of approximations used is often limited, e.g., mean-field approximations, implying that no solution is ever able to resemble the true posterior distribution. This is a widely raised objection to variational methods, in that unlike other inferential methods such as MCMC, even in the asymptotic regime we are unable recover the true posterior distribution.

There is much evidence that richer, more faithful posterior approximations do result in better performance. For example, when compared to sigmoid belief networks that make use of mean-field approximations, deep auto-regressive networks use a posterior approximation with an autoregressive dependency structure that provides a clear improvement in performance (Mnih & Gregor, 2014). There is also a large body of evidence that describes the detrimental effect of limited posterior approximations. Turner & Sahani (2011) provide an exposition of two commonly experienced problems. The first is the widely-observed problem of under-estimation of the variance of the posterior distribution, which can result in poor predictions and unreliable decisions based on the chosen posterior approximation. The second is that the limited capacity of the posterior approximation can also result in biases in the MAP estimates of any model parameters (and this is the case e.g., in time-series models).

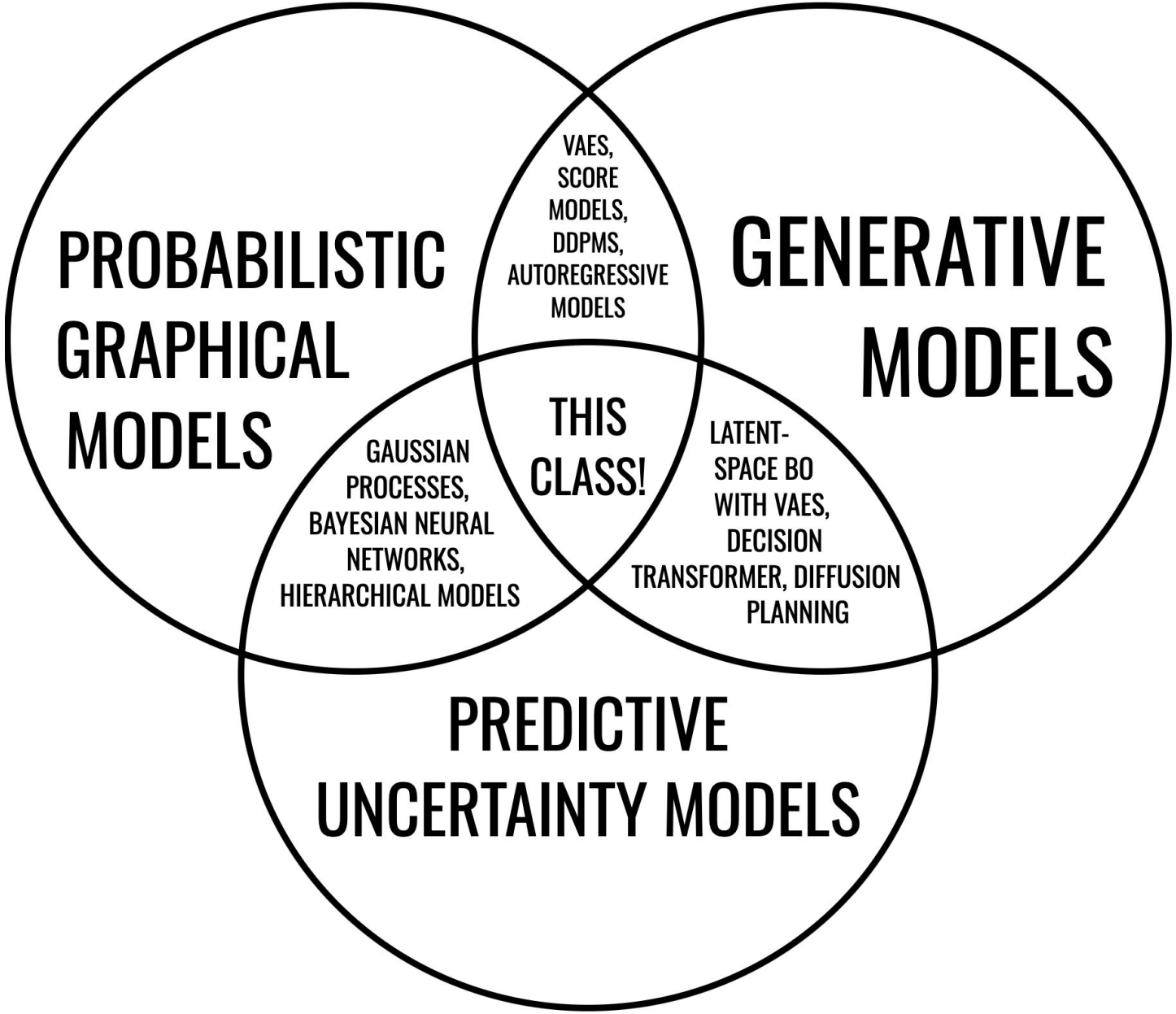
A number of proposals for rich posterior approximations have been explored, typically based on structured mean-field approximations that incorporate some basic form of dependency within the approximate posterior. Another potentially powerful alternative would be to specify the approximate posterior as a mixture model, such as those developed by Jaakkola & Jordan (1998); Jordan et al. (1999); Gershman et al. (2012). But the mixture approach limits

Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

Normalizing Flows, 2015

This Class

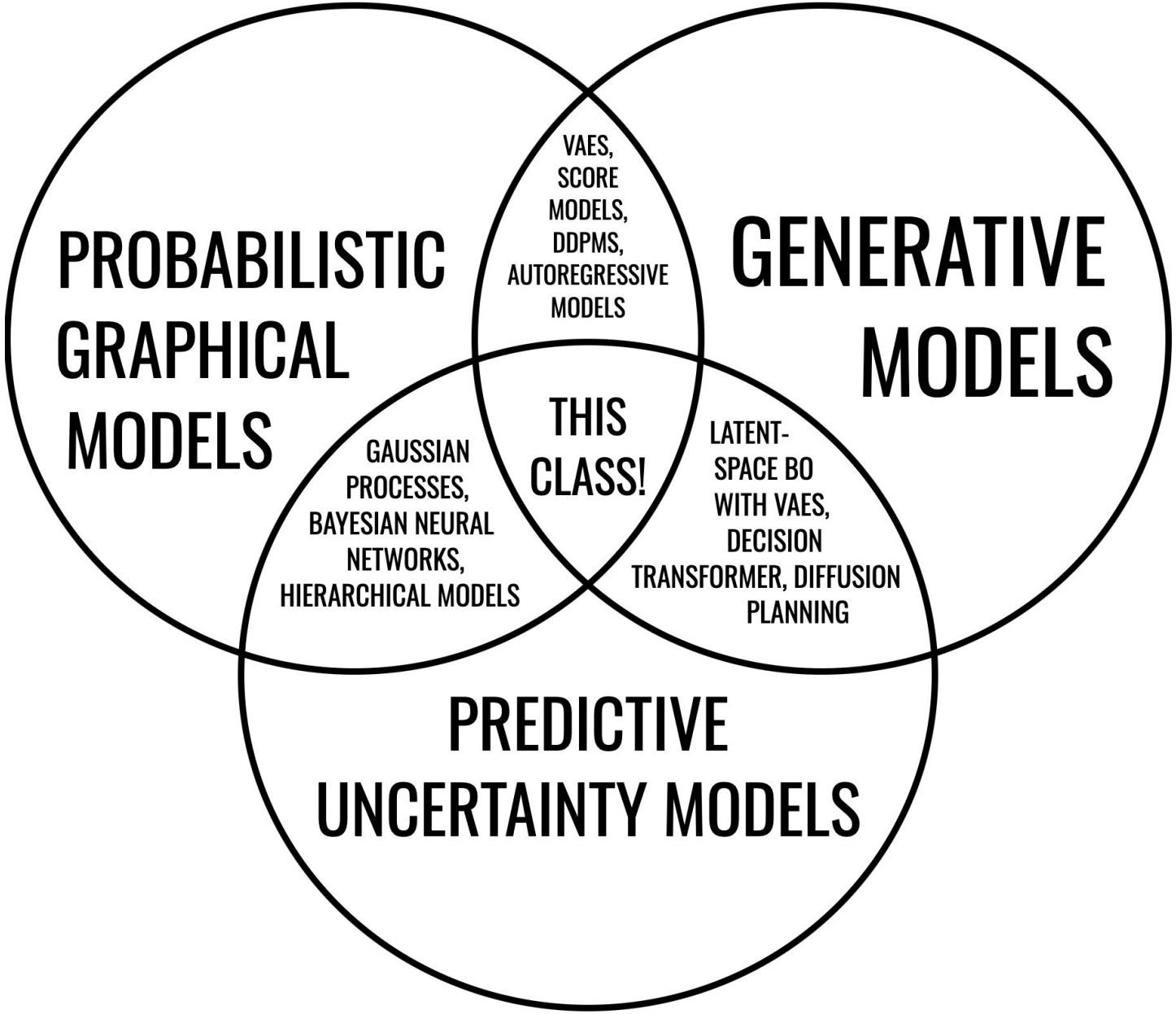
This course focuses on probabilistic models and their central role within modern machine learning and generative modeling.



This Class

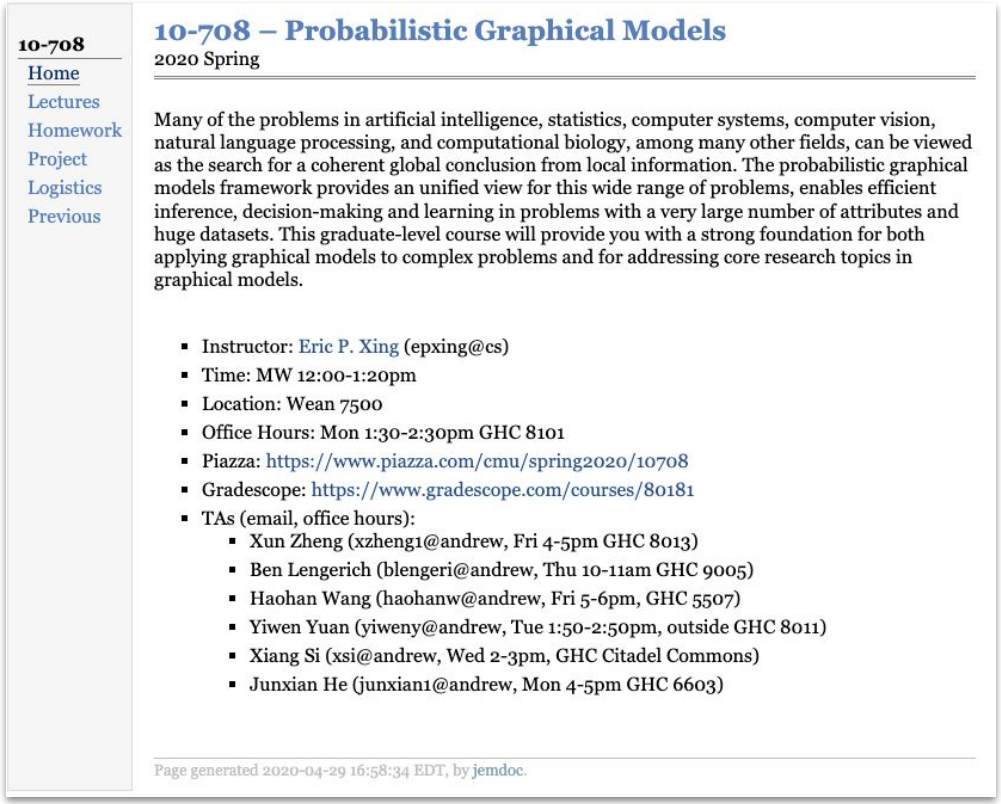
This course focuses on probabilistic models and their central role within modern machine learning and generative modeling.

This course is inspired by some previous classes…



This Class — Inspiration

Probabilistic Graphical Models (PGM) class → *e.g.*, at CMU & Stanford



10-708

Home
Lectures
Homework
Project
Logistics
Previous

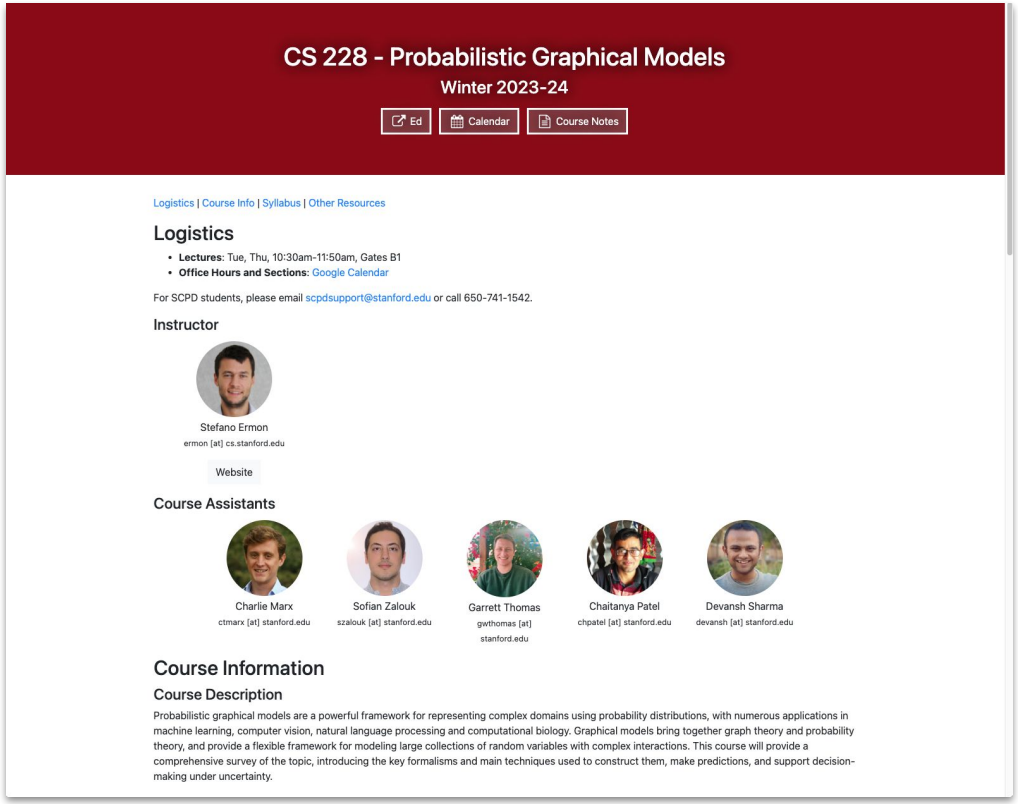
10-708 – Probabilistic Graphical Models

2020 Spring

Many of the problems in artificial intelligence, statistics, computer systems, computer vision, natural language processing, and computational biology, among many other fields, can be viewed as the search for a coherent global conclusion from local information. The probabilistic graphical models framework provides an unified view for this wide range of problems, enables efficient inference, decision-making and learning in problems with a very large number of attributes and huge datasets. This graduate-level course will provide you with a strong foundation for both applying graphical models to complex problems and for addressing core research topics in graphical models.

- Instructor: Eric P. Xing (epxing@cs)
- Time: MW 12:00-1:20pm
- Location: Wean 7500
- Office Hours: Mon 1:30-2:30pm GHC 8101
- Piazza: https://www.piazza.com/cmu/spring2020/10708
- Gradescope: https://www.gradescope.com/courses/80181
- TAs (email, office hours):
  - Xun Zheng (xzheng1@andrew, Fri 4-5pm GHC 8013)
  - Ben Lengerich (blengeri@andrew, Thu 10-11am GHC 9005)
  - Haohan Wang (haohanw@andrew, Fri 5-6pm, GHC 5507)
  - Yiwen Yuan (yiweny@andrew, Tue 1:50-2:50pm, outside GHC 8011)
  - Xiang Si (xsi@andrew, Wed 2-3pm, GHC Citadel Commons)
  - Junxian He (junxian1@andrew, Mon 4-5pm GHC 6603)

Page generated 2020-04-29 16:58:34 EDT, by jemdoc.



CS 228 - Probabilistic Graphical Models

Winter 2023-24

Ed | Calendar | Course Notes

Logistics | Course Info | Syllabus | Other Resources

Logistics

- **Lectures**: Tue, Thu, 10:30am-11:50am, Gates B1
- **Office Hours and Sections**: Google Calendar

For SCPD students, please email scpdsupport@stanford.edu or call 650-741-1542.

Instructor

Stefano Ermon
ermon [at] cs.stanford.edu

Website

Course Assistants

Charlie Marx
ctmarx [at] stanford.edu

Sofian Zalouk
szalouk [at] stanford.edu

Garrett Thomas
gwthomas [at] stanford.edu

Chaitanya Patel
chpatel [at] stanford.edu

Devansh Sharma
devansh [at] stanford.edu

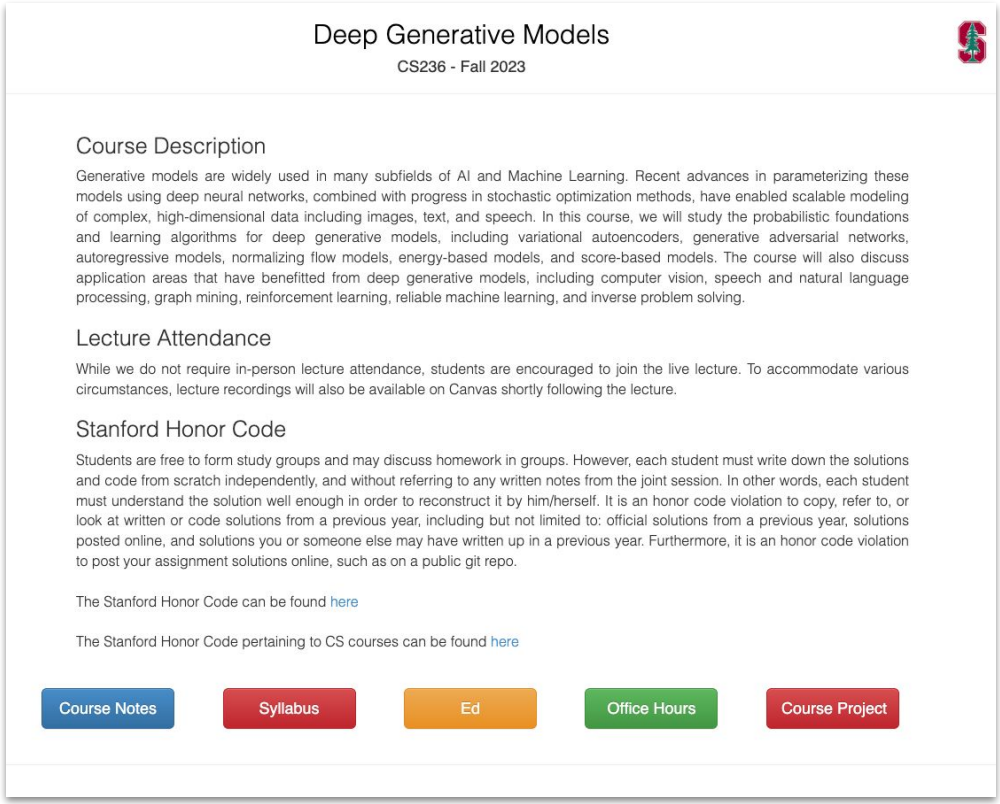
Course Information

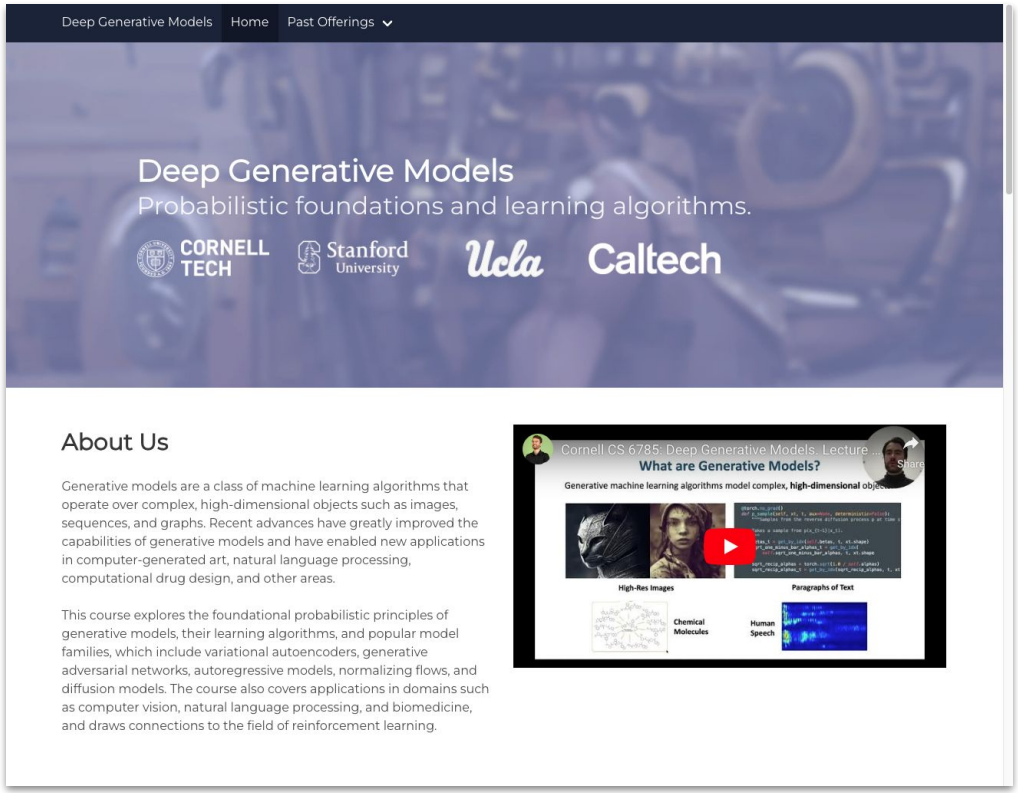
Course Description

Probabilistic graphical models are a powerful framework for representing complex domains using probability distributions, with numerous applications in machine learning, computer vision, natural language processing and computational biology. Graphical models bring together graph theory and probability theory, and provide a flexible framework for modeling large collections of random variables with complex interactions. This course will provide a comprehensive survey of the topic, introducing the key formalisms and main techniques used to construct them, make predictions, and support decision-making under uncertainty.

This Class — Inspiration

Deep Generative Models (DGM) class → *e.g.*, at Stanford, Cornell, UCLA, etc.





Probabilistic Modeling — What is a probabilistic model?

Probabilistic Modeling — What is a probabilistic model?

It’s common to define a mathematical model of the world in the form of an equation.

Probabilistic Modeling — What is a probabilistic model?

It's common to define a mathematical model of the world in the form of an equation.

Example: a simple model is a linear equation of the form:

$$y = \beta^\top x$$

Where y is an outcome variable we want to predict, and x are known input variables that affect the outcome.

Probabilistic Modeling — What is a probabilistic model?

It's common to define a mathematical model of the world in the form of an equation.

Example: a simple model is a linear equation of the form:

$$y = \beta^\top x$$

Where y is an outcome variable we want to predict, and x are known input variables that affect the outcome.

- y — the price of a house.
- x — features that affect the price (*location*, *# bedrooms*, *age of house*, etc.)

Assume that y is a linear function of these inputs, parameterized by β.

Probabilistic Modeling — Why be probabilistic?

Probabilistic Modeling — Why be probabilistic?

Often real-world has a significant amount of uncertainty

(*e.g.*, the price y has a certain chance of going up if a new restaurant opens nearby that is popular…etc).

Probabilistic Modeling — Why be probabilistic?

Often real-world has a significant amount of uncertainty

(*e.g.*, the price y has a certain chance of going up if a new restaurant opens nearby that is popular…etc).

It is natural to deal with this uncertainty by modeling the world in the form of a probability distribution: $p(x, y)$

Probabilistic Modeling — Why be probabilistic?

Often real-world has a significant amount of uncertainty

(*e.g.*, the price y has a certain chance of going up if a new restaurant opens nearby that is popular…etc).

It is natural to deal with this uncertainty by modeling the world in the form of a probability distribution: $p(x, y)$

Given such a model, we could answer questions like:

Probabilistic Modeling — Why be probabilistic?

Often real-world has a significant amount of uncertainty

(*e.g.*, the price y has a certain chance of going up if a new restaurant opens nearby that is popular…etc).

It is natural to deal with this uncertainty by modeling the world in the form of a probability distribution: $p(x, y)$

Given such a model, we could answer questions like:

- "What is the probability that housing price will be >$50,000?"

Marginal Probability

Probabilistic Modeling — Why be probabilistic?

Often real-world has a significant amount of uncertainty

(*e.g.*, the price y has a certain chance of going up if a new restaurant opens nearby that is popular…etc).

It is natural to deal with this uncertainty by modeling the world in the form of a probability distribution: $p(x, y)$

Given such a model, we could answer questions like:

- "What is the probability that housing price will be >$50,000?"
- "Given that the house costs $100,000, what is the probability that it has three bedrooms?"

Conditional Probability

Probabilistic Modeling — Why be probabilistic?

Often real-world has a significant amount of uncertainty

(*e.g.*, the price y has a certain chance of going up if a new restaurant opens nearby that is popular…etc).

It is natural to deal with this uncertainty by modeling the world in the form of a probability distribution: $p(x, y)$

Given such a model, we could answer questions like:

- "What is the probability that housing price will be >$50,000?"
- "Given that the house costs $100,000, what is the probability that it has three bedrooms?"
- "Given that the house costs $1,000,000, what is an example of a plausible house?"

Sampling

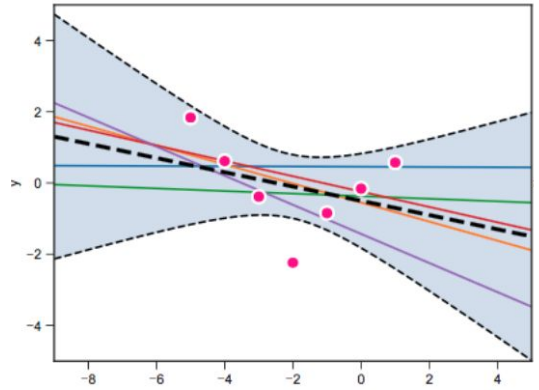
Probabilistic Modeling — Why be probabilistic?

Probability is important here! For a few key reasons:

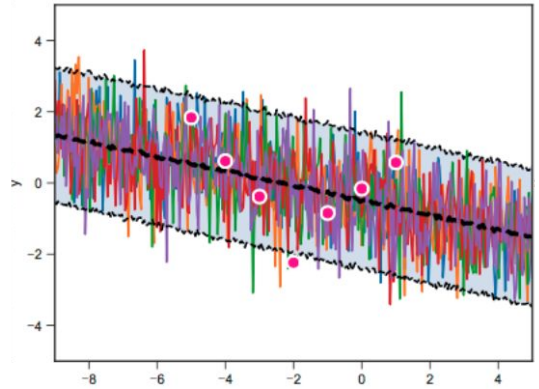
Probabilistic Modeling — Why be probabilistic?

Probability is important here! For a few key reasons:

1. Need to handle uncertainty about the environment. We cannot make perfect predictions, and there is uncertainty due to:
   - lack of knowledge about the world (*epistemic uncertainty*).
   - inherent stochasticity on outcomes (*aleatoric uncertainty*).



Epistemic uncertainty over linear model parameters



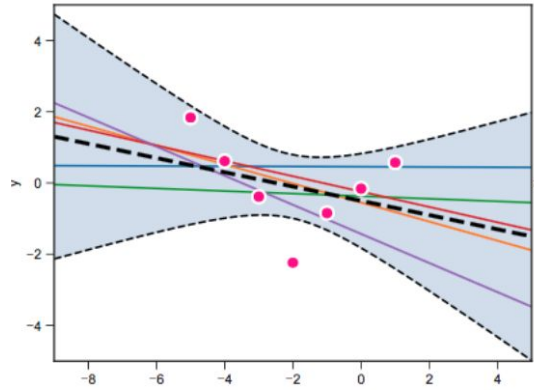
Aleatoric uncertainty shown for a single parameter

Probabilistic Modeling — Why be probabilistic?

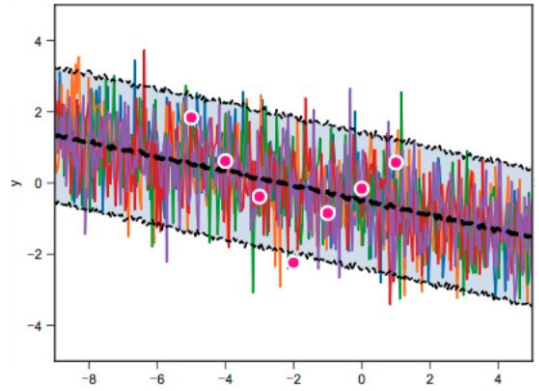
Probability is important here! For a few key reasons:

1. Need to handle uncertainty about the environment.
   We cannot make perfect predictions, and there is uncertainty due to:
   - lack of knowledge about the world (*epistemic uncertainty*).
   - inherent stochasticity on outcomes (*aleatoric uncertainty*).

2. Want to use uncertainty over our predictions.
   Predicting a single value is often not enough.
   We want a system that outputs its beliefs about what is going on in the world.



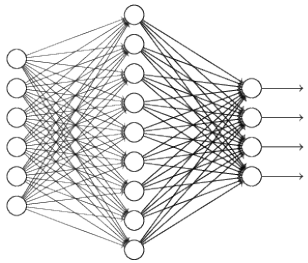
Epistemic uncertainty over linear model parameters



Aleatoric uncertainty shown for a single parameter

Probabilistic Modeling — Uncertainty

Machine learning is based on predictions — typically “point predictions”.

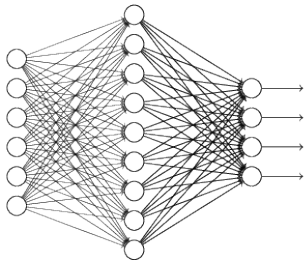


Dog

Probabilistic Modeling — Uncertainty

Machine learning is based on predictions — typically “point predictions”.

Predictions **with reliable uncertainties** (*e.g.*, with a measure of confidence) are useful for enabling trustworthy machine learning.



Dog?

Cat?

Rabid Cat?

Cotton Ball?

Trap?

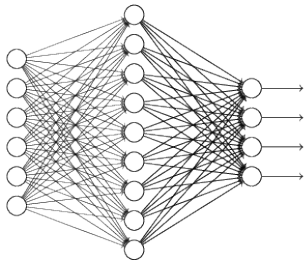
Probabilistic Modeling — Uncertainty

Machine learning is based on predictions — typically “point predictions”.

Predictions **with reliable uncertainties** (*e.g.*, with a measure of confidence) are useful for enabling trustworthy machine learning.

This predictive uncertainty quantification is useful for *decision-making under uncertainty*.

(Applies to both classical probabilistic models and deep learning models.)



Dog?

Cat?

Rabid Cat?

Cotton Ball?

Trap?

“Should I pet it?”

Deep Uncertainty Models

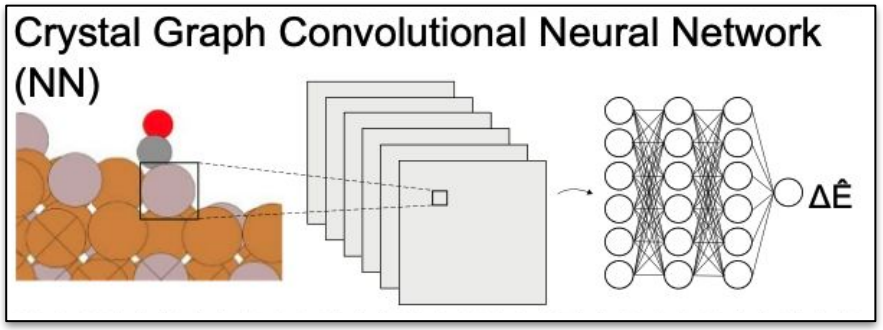
For example, we'll study methods for incorporating predictive uncertainty into neural networks:

Deep Uncertainty Models —Example: Computational Catalyst Design

For example, we'll study methods for incorporating predictive uncertainty into neural networks:

Deep Uncertainty Models —Example: Computational Catalyst Design

For example, we'll study methods for incorporating predictive uncertainty into neural networks:



*Xie, Tian, and Jeffrey C. Grossman. "**Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties.**" Physical review letters, 2018.*

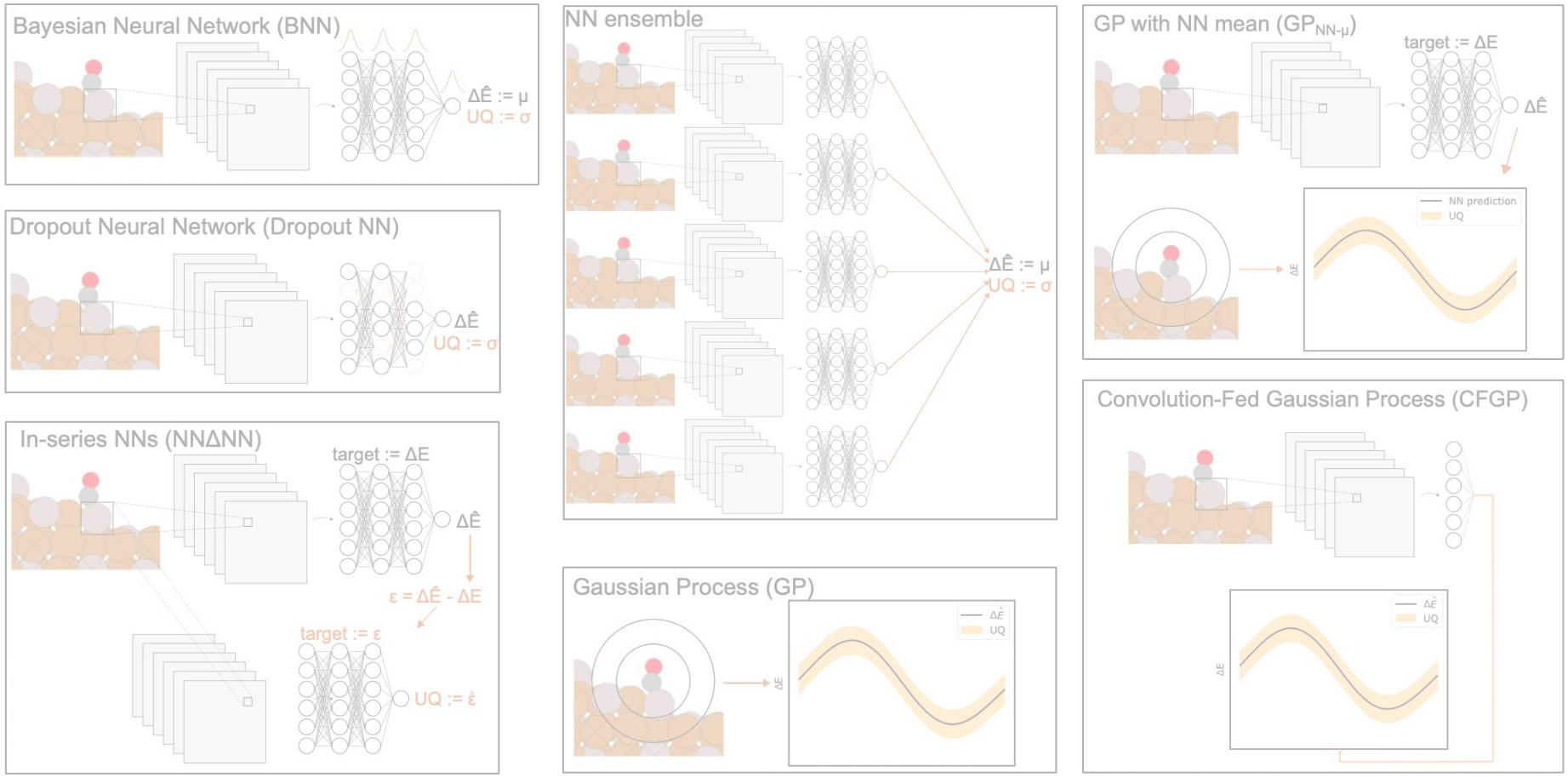
CGCNN Model (a graph convolutional neural network model)

- *Inputs*: three-dimensional atomic structure (a graph).
- *Outputs*: DFT-calculated site adsorption energies ΔE (*regression*).

"Methods for comparing uncertainty quantifications for material property predictions", *Tran, *Neiswanger, et al., MLST. 2020

"Computational catalyst discovery: Active classification through myopic multiscale sampling", *Tran, *Neiswanger, et al., J. Chem. Phys. 2021

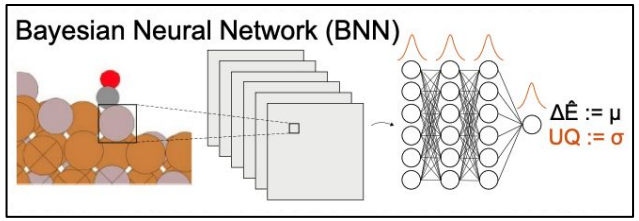
Deep Uncertainty Models —Example: Computational Catalyst Design

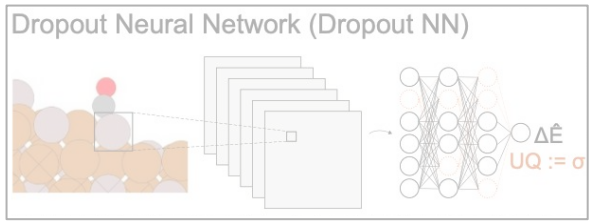


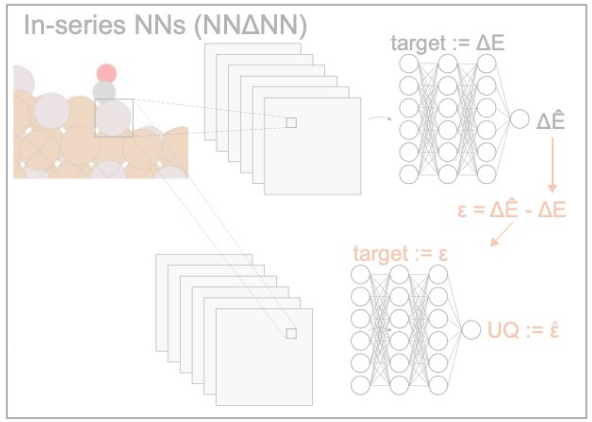
"Methods for comparing uncertainty quantifications for material property predictions", *Tran, *Neiswanger, et al., MLST. 2020

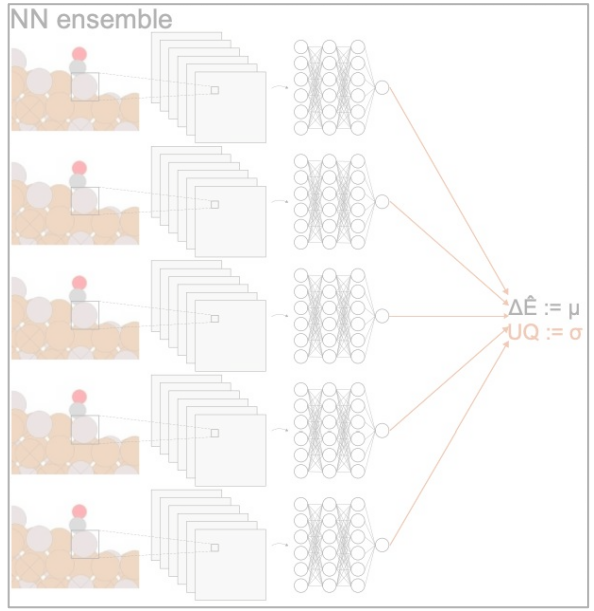
"Computational catalyst discovery: Active classification through myopic multiscale sampling", *Tran, *Neiswanger, et al., J. Chem. Phys. 2021

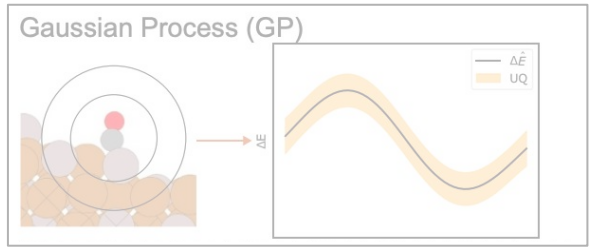
Deep Uncertainty Models —Example: Computational Catalyst Design

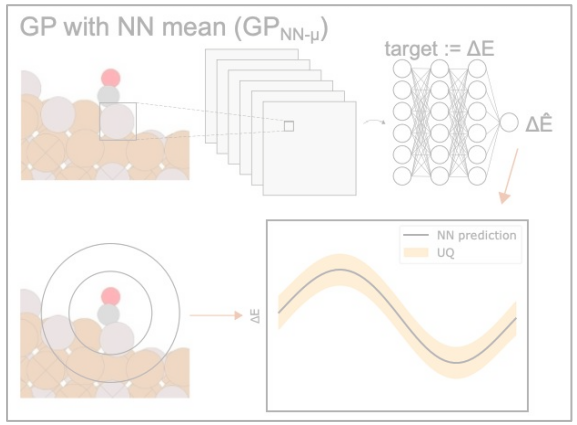


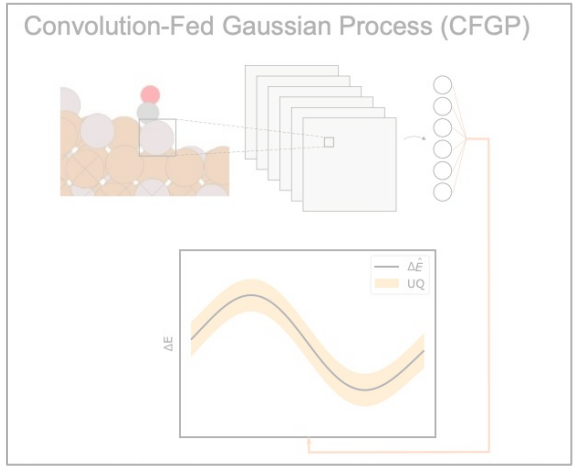








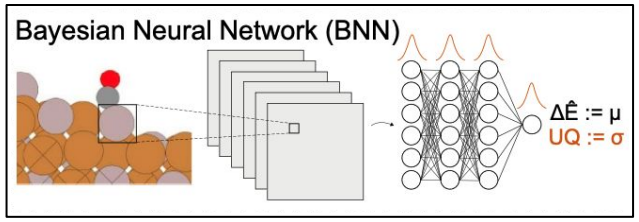


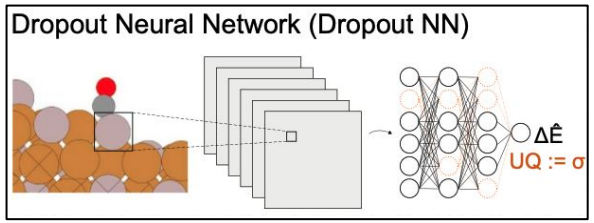


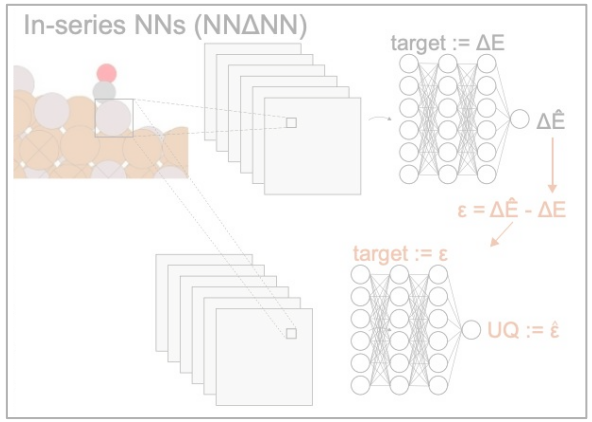
"Methods for comparing uncertainty quantifications for material property predictions", *Tran, *Neiswanger, et al., MLST. 2020

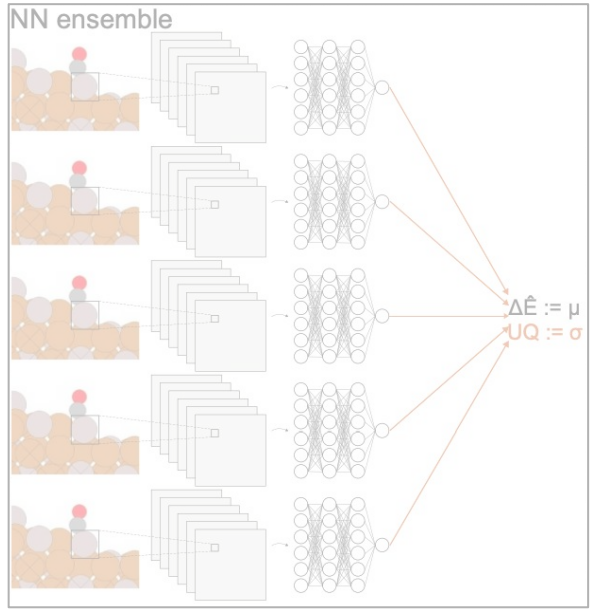
"Computational catalyst discovery: Active classification through myopic multiscale sampling", *Tran, *Neiswanger, et al., J. Chem. Phys. 2021

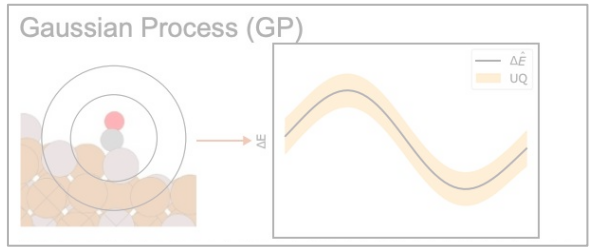
Deep Uncertainty Models —Example: Computational Catalyst Design

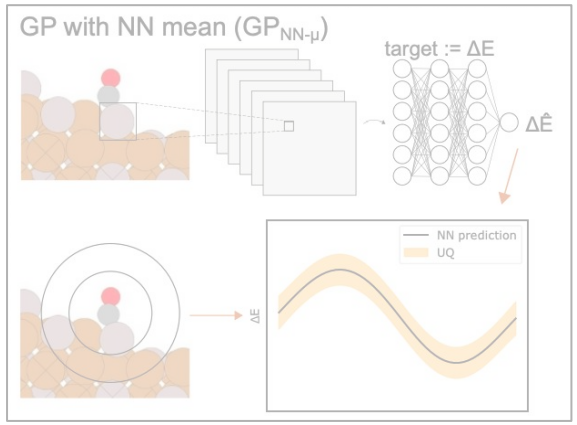


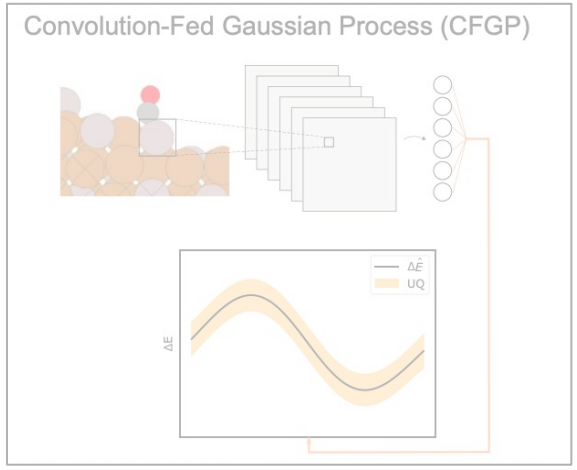








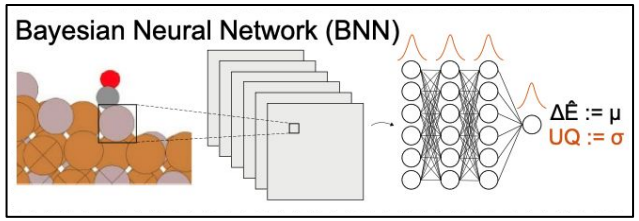


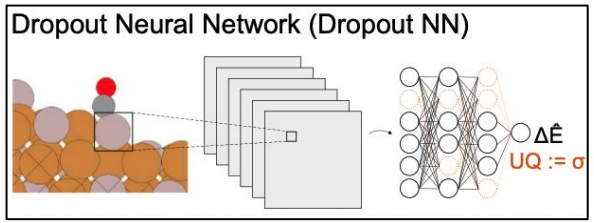


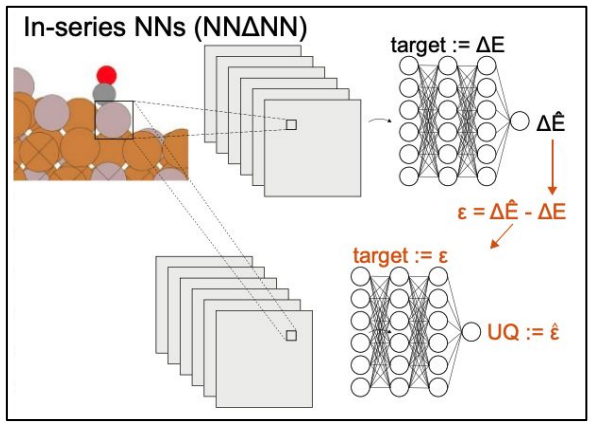
"Methods for comparing uncertainty quantifications for material property predictions", *Tran, *Neiswanger, et al., MLST. 2020

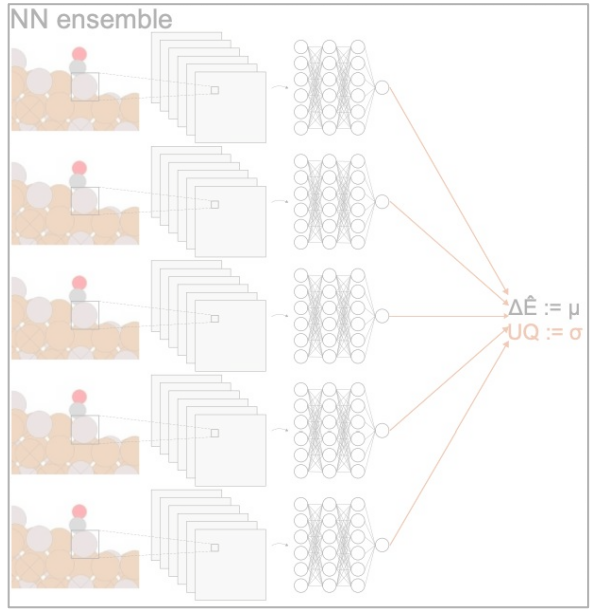
"Computational catalyst discovery: Active classification through myopic multiscale sampling", *Tran, *Neiswanger, et al., J. Chem. Phys. 2021

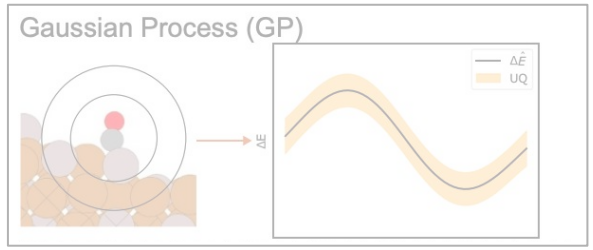
Deep Uncertainty Models —Example: Computational Catalyst Design

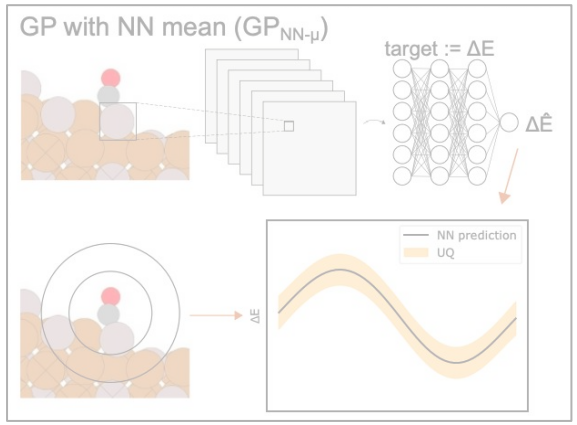


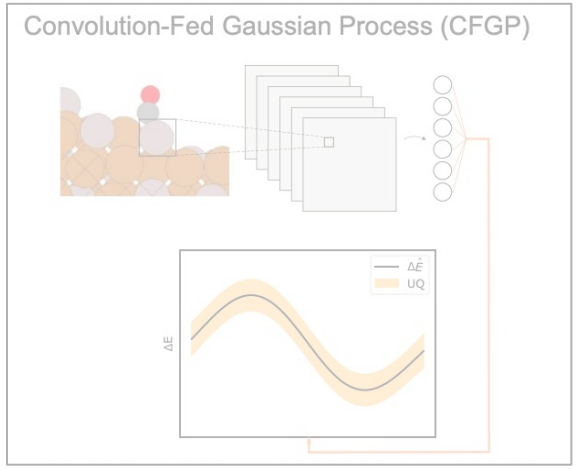








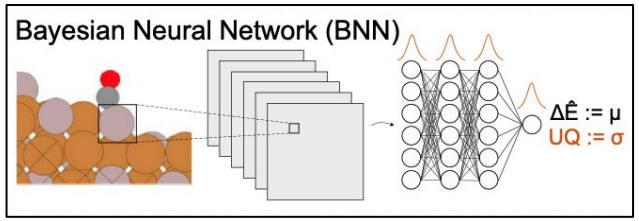


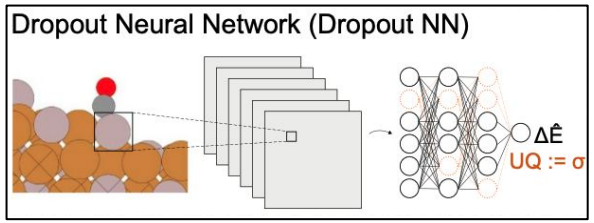


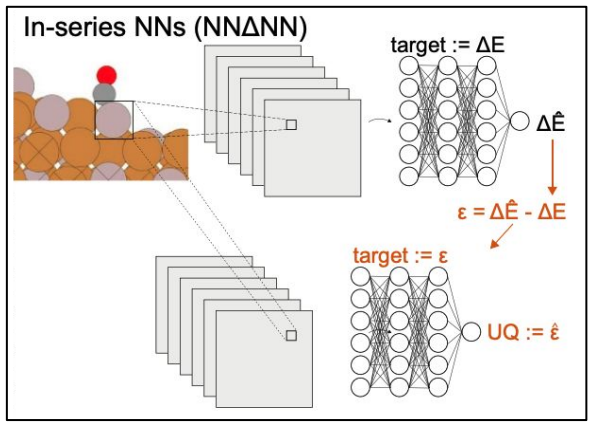
"Methods for comparing uncertainty quantifications for material property predictions", *Tran, *Neiswanger, et al., MLST. 2020

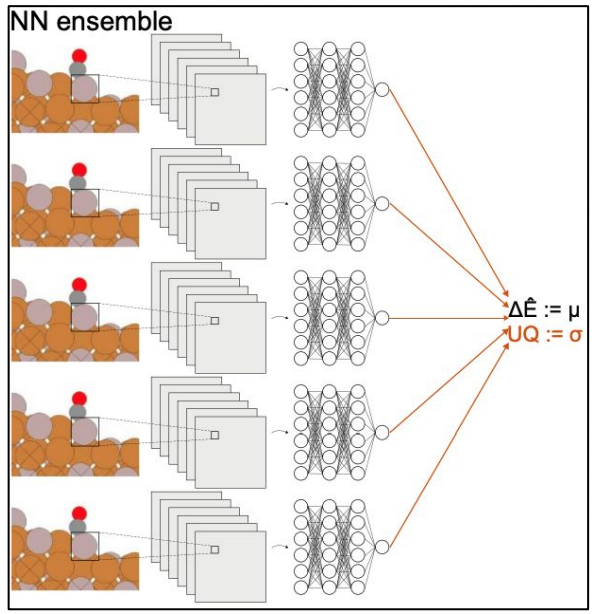
"Computational catalyst discovery: Active classification through myopic multiscale sampling", *Tran, *Neiswanger, et al., J. Chem. Phys. 2021

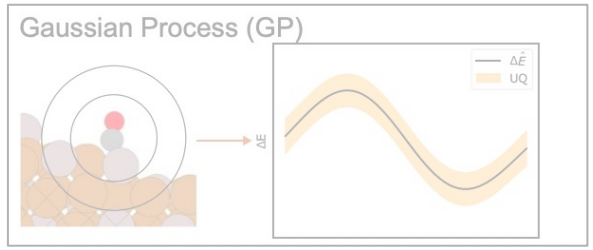
Deep Uncertainty Models —Example: Computational Catalyst Design

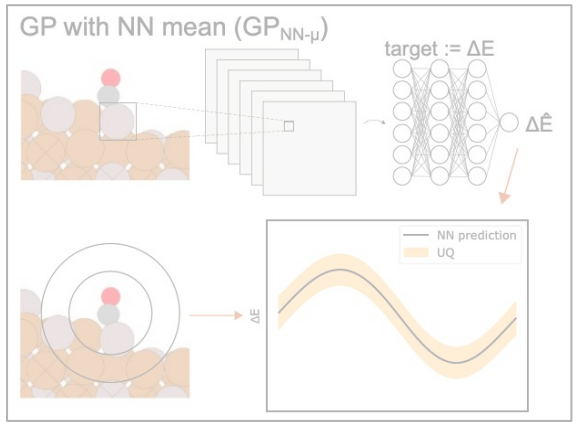


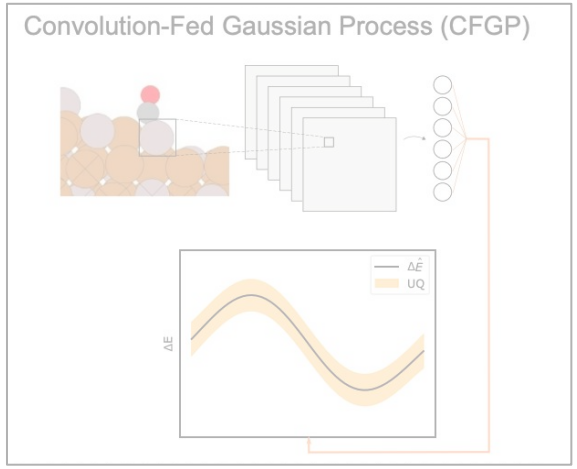








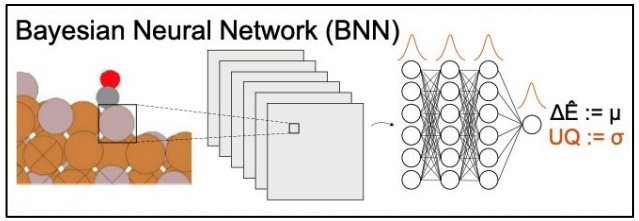


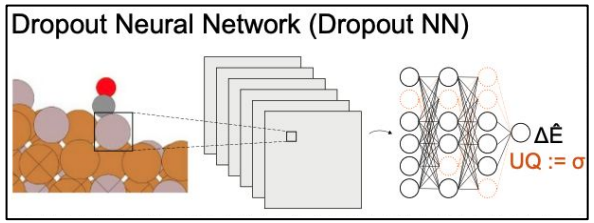


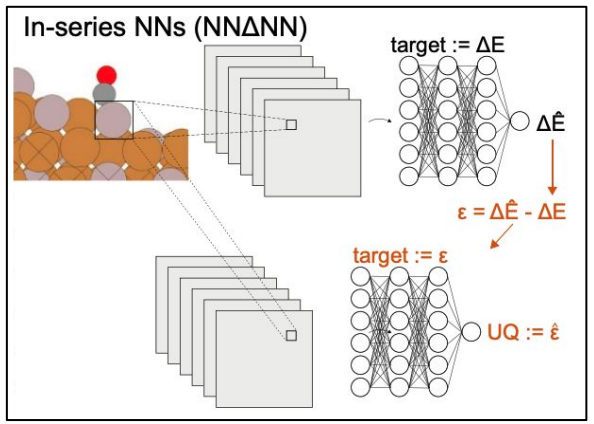
"Methods for comparing uncertainty quantifications for material property predictions", *Tran, *Neiswanger, et al., MLST. 2020

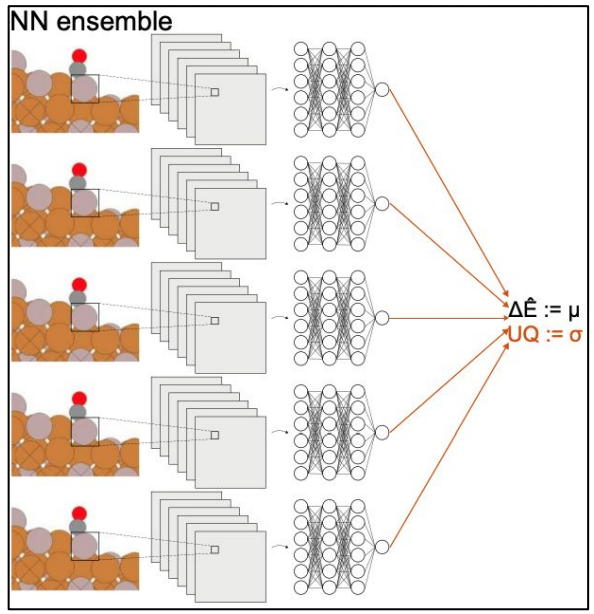
"Computational catalyst discovery: Active classification through myopic multiscale sampling", *Tran, *Neiswanger, et al., J. Chem. Phys. 2021

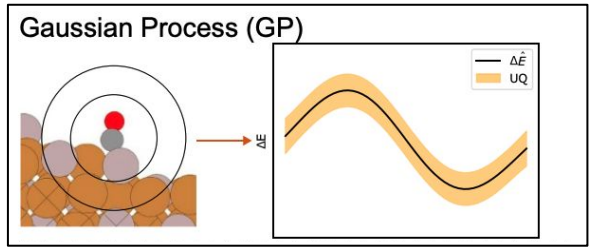
Deep Uncertainty Models —Example: Computational Catalyst Design

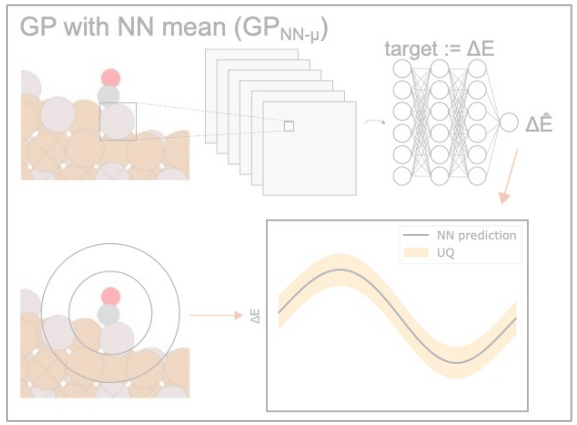


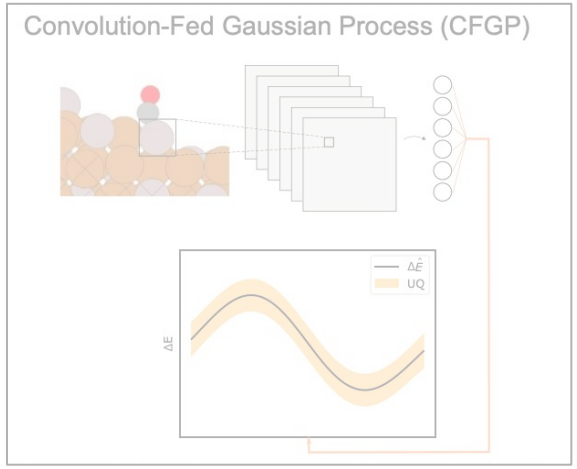








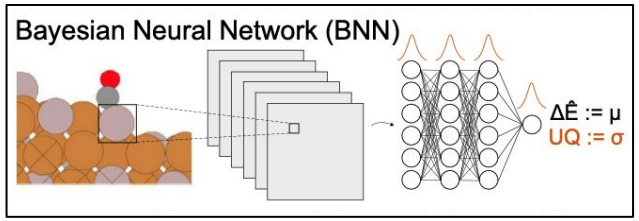


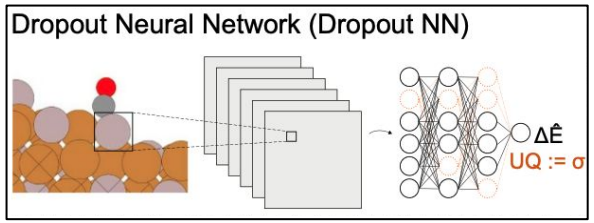


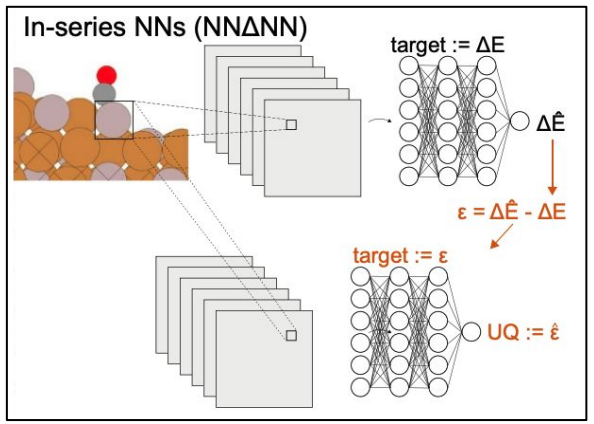
"Methods for comparing uncertainty quantifications for material property predictions", *Tran, *Neiswanger, et al., MLST. 2020

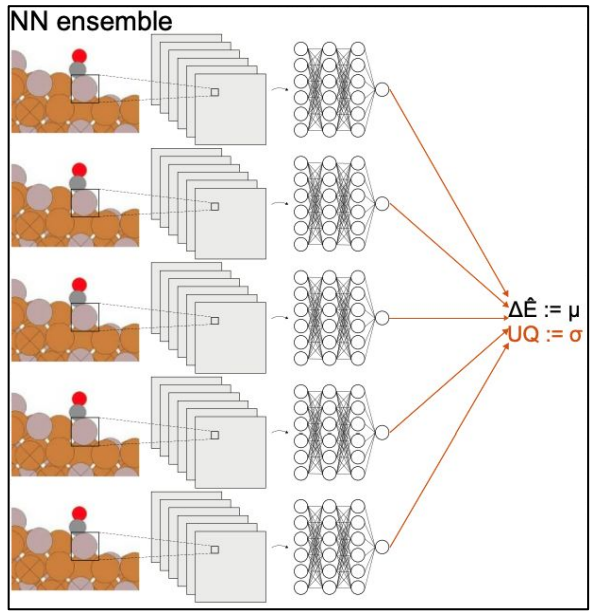
"Computational catalyst discovery: Active classification through myopic multiscale sampling", *Tran, *Neiswanger, et al., J. Chem. Phys. 2021

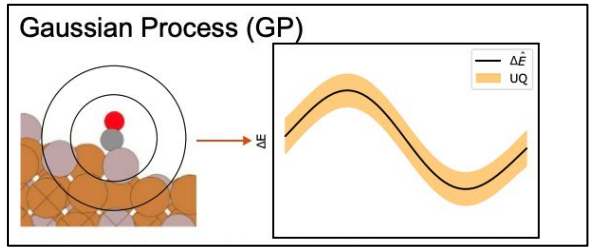
Deep Uncertainty Models —Example: Computational Catalyst Design

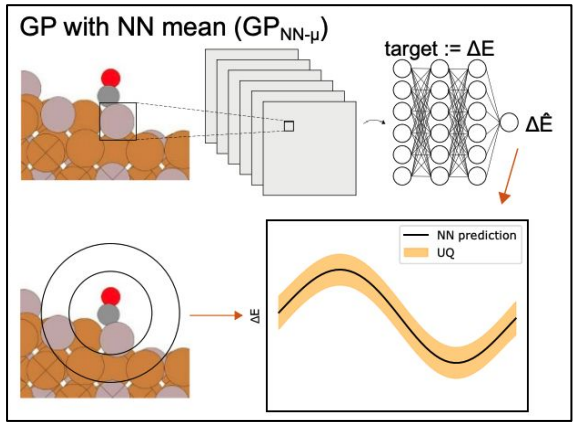


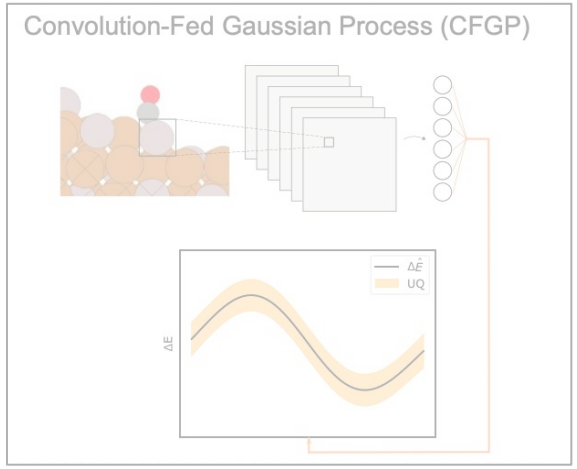








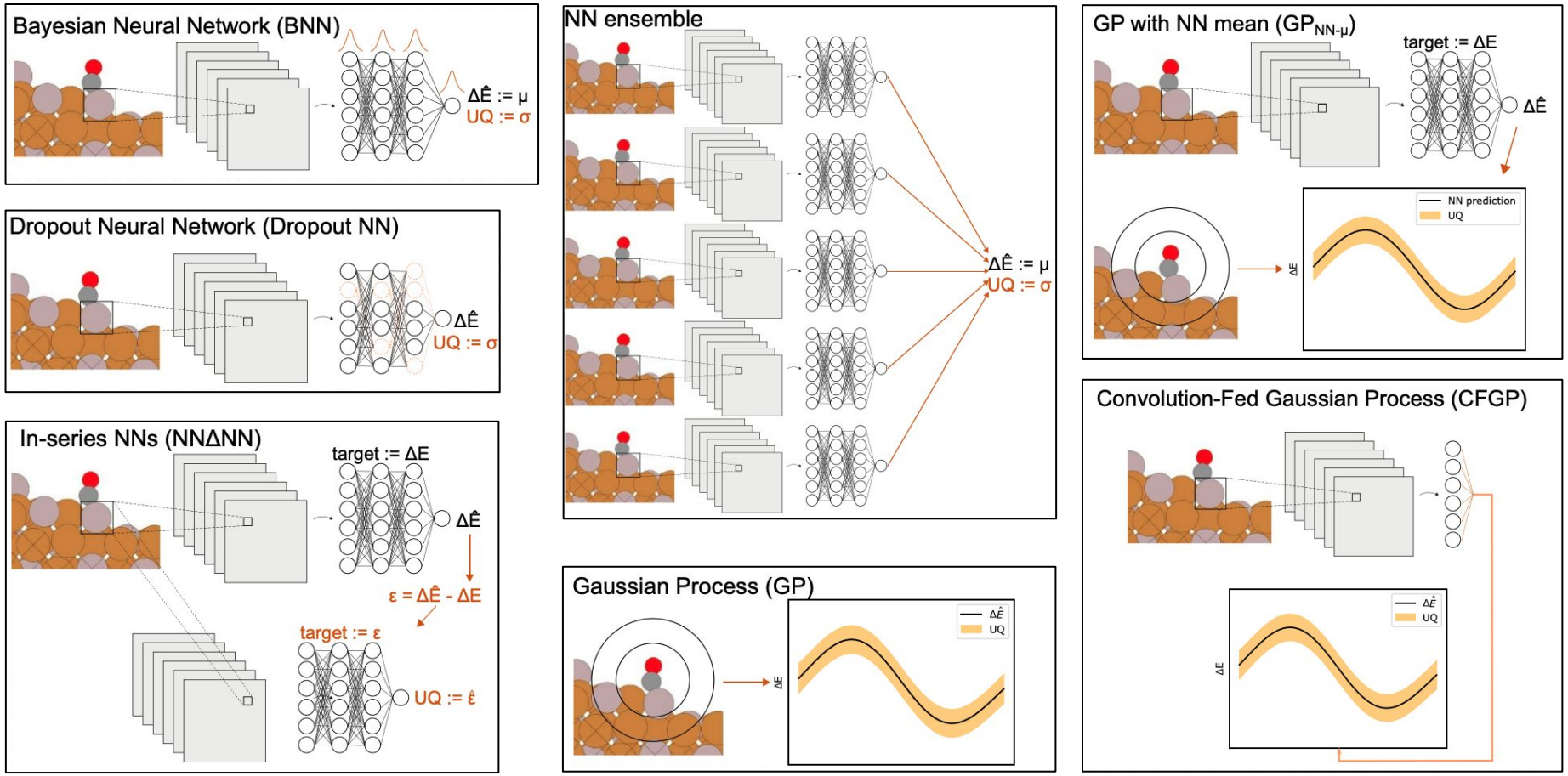




"Methods for comparing uncertainty quantifications for material property predictions", *Tran, *Neiswanger, et al., MLST. 2020

"Computational catalyst discovery: Active classification through myopic multiscale sampling", *Tran, *Neiswanger, et al., J. Chem. Phys. 2021

Deep Uncertainty Models —Example: Computational Catalyst Design



"Methods for comparing uncertainty quantifications for material property predictions", *Tran, *Neiswanger, et al., MLST. 2020

"Computational catalyst discovery: Active classification through myopic multiscale sampling", *Tran, *Neiswanger, et al., J. Chem. Phys. 2021

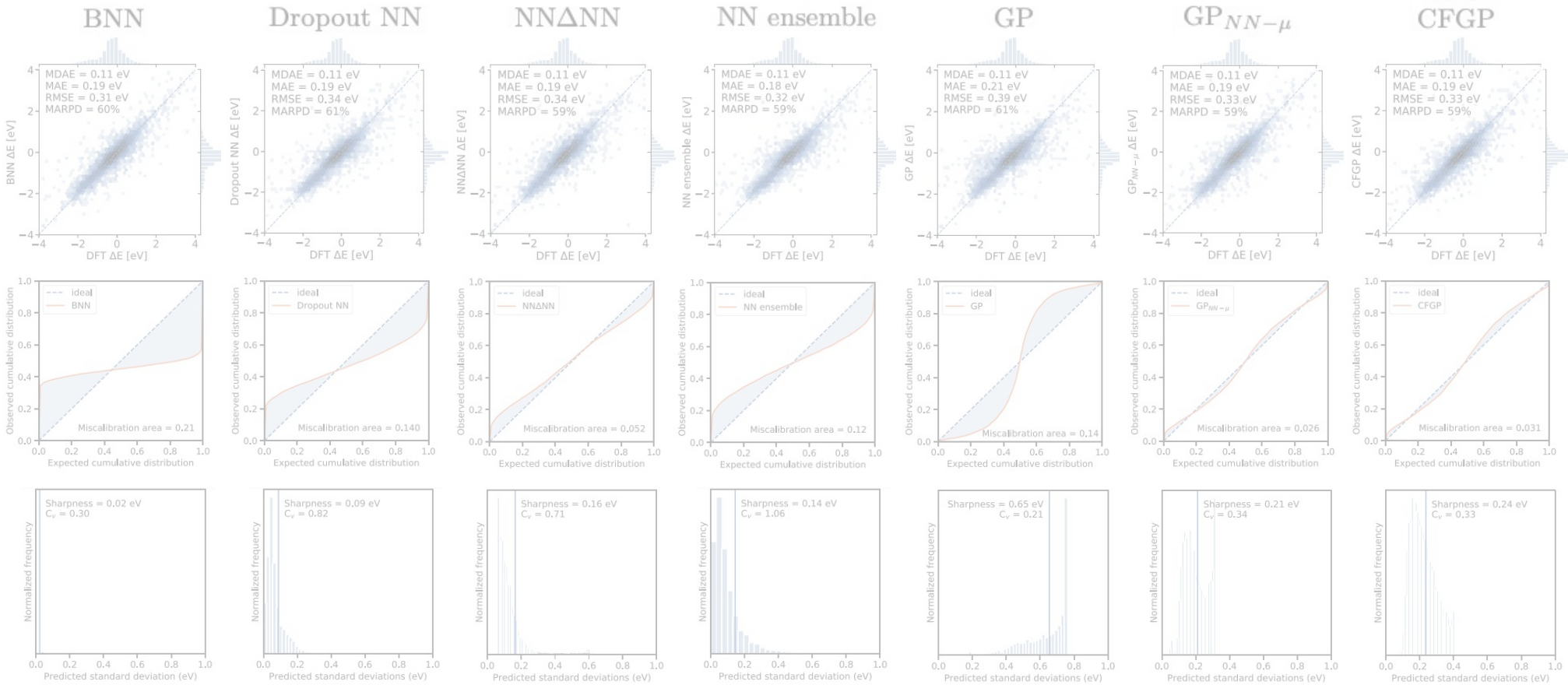
Deep Uncertainty Models —Example: Computational Catalyst Design

Metrics for deep uncertainty models.

“Methods for comparing uncertainty quantifications for material property predictions”, *Tran, *Neiswanger, et al., MLST. 2020

“Computational catalyst discovery: Active classification through myopic multiscale sampling”, *Tran, *Neiswanger, et al., J. Chem. Phys. 2021

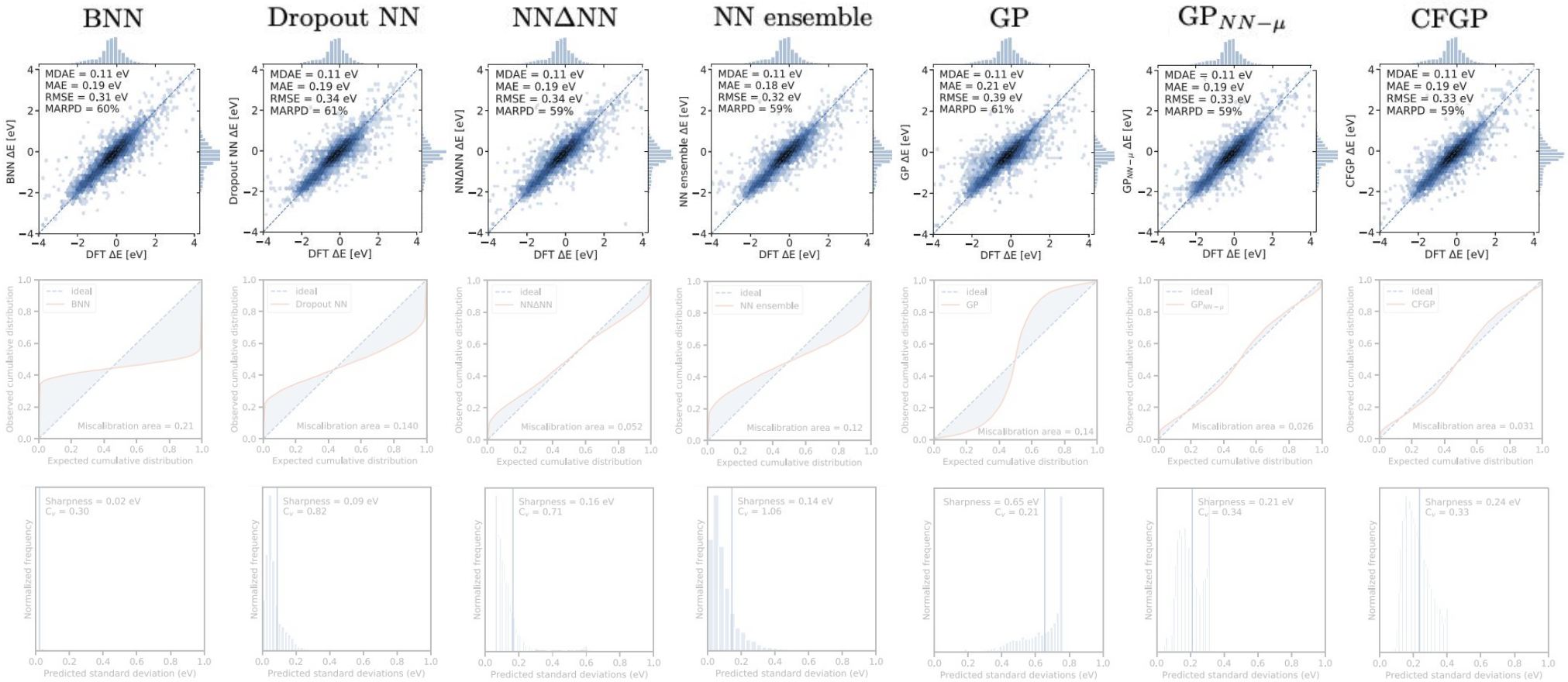
Deep Uncertainty Models —Example: Computational Catalyst Design



"Methods for comparing uncertainty quantifications for material property predictions", *Tran, *Neiswanger, et al., MLST. 2020

"Computational catalyst discovery: Active classification through myopic multiscale sampling", *Tran, *Neiswanger, et al., J. Chem. Phys. 2021

Deep Uncertainty Models —Example: Computational Catalyst Design



"Methods for comparing uncertainty quantifications for material property predictions", *Tran, *Neiswanger, et al., MLST. 2020

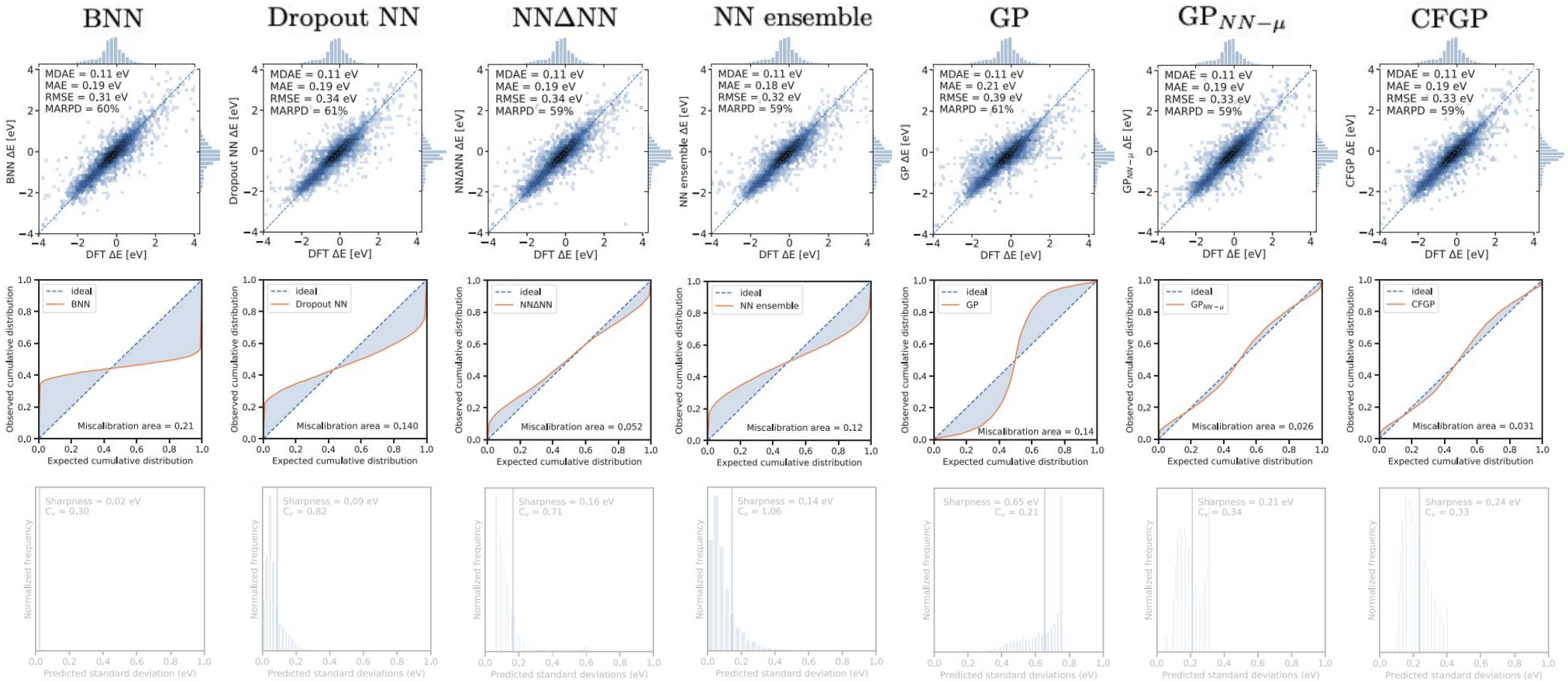
"Computational catalyst discovery: Active classification through myopic multiscale sampling", *Tran, *Neiswanger, et al., J. Chem. Phys. 2021

Deep Uncertainty Models —Example: Computational Catalyst Design

Accuracy

Calibration

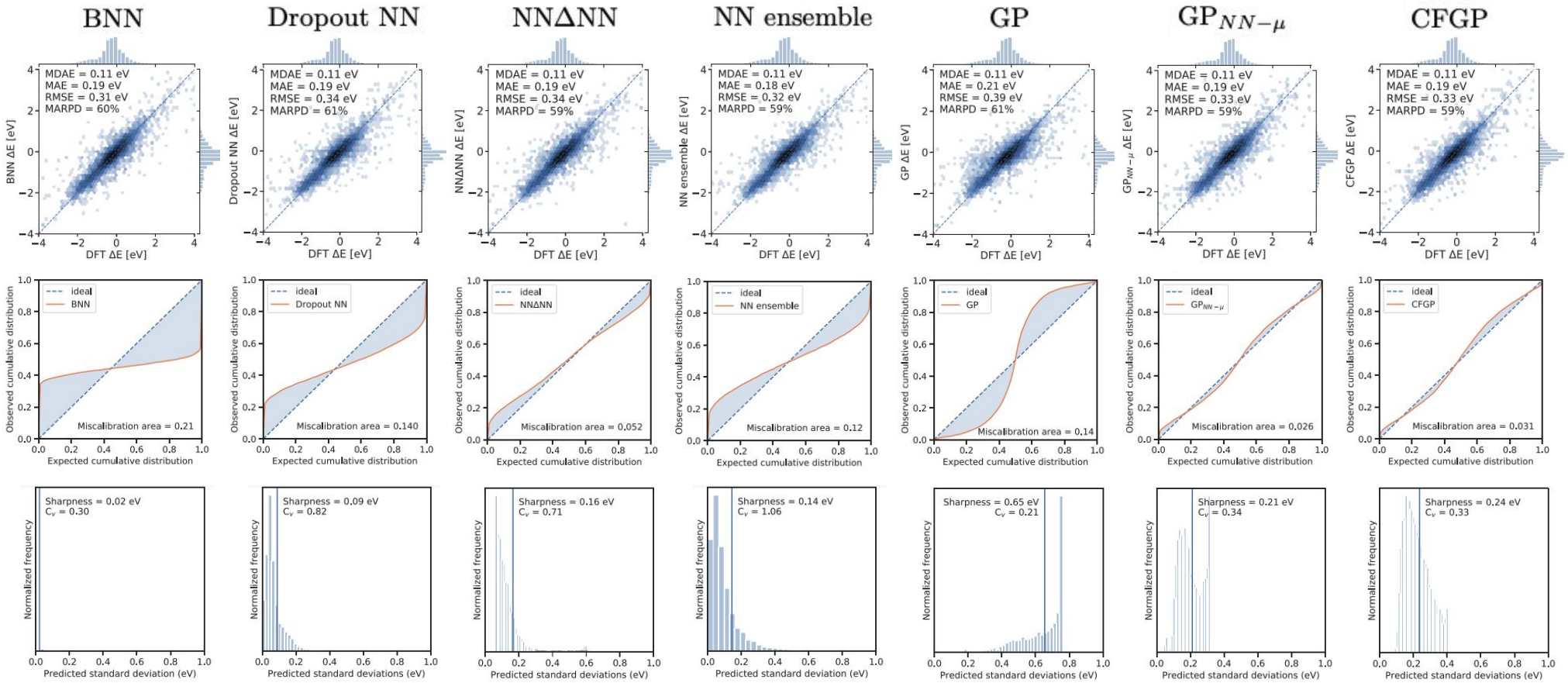
Sharpness



"Methods for comparing uncertainty quantifications for material property predictions", *Tran, *Neiswanger, et al., MLST. 2020

"Computational catalyst discovery: Active classification through myopic multiscale sampling", *Tran, *Neiswanger, et al., J. Chem. Phys. 2021

Deep Uncertainty Models —Example: Computational Catalyst Design



"Methods for comparing uncertainty quantifications for material property predictions", *Tran, *Neiswanger, et al., MLST. 2020

"Computational catalyst discovery: Active classification through myopic multiscale sampling", *Tran, *Neiswanger, et al., J. Chem. Phys. 2021

Deep Uncertainty Models —Decision Making Under Uncertainty

Many applications of these models in decision making under uncertainty.

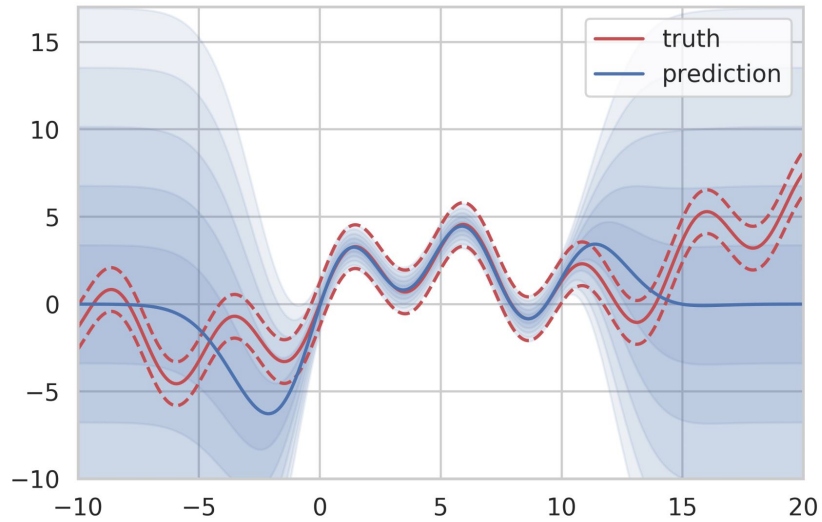
(Especially in the sciences and engineering).

Deep Uncertainty Models —Decision Making Under Uncertainty

Many applications of these models in decision making under uncertainty.

(Especially in the sciences and engineering).

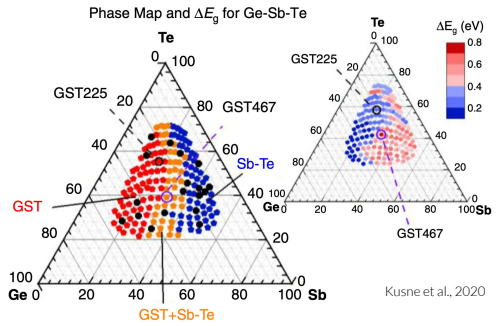
E.g., using probabilistic models for optimization, experimental design, and active learning.



Deep Uncertainty Models —Decision Making Under Uncertainty

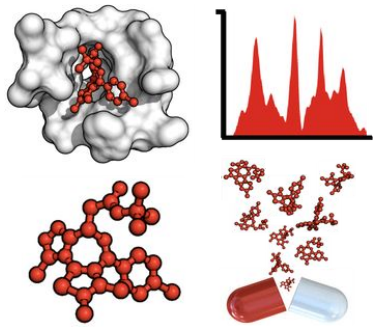
SCIENCE

Materials design



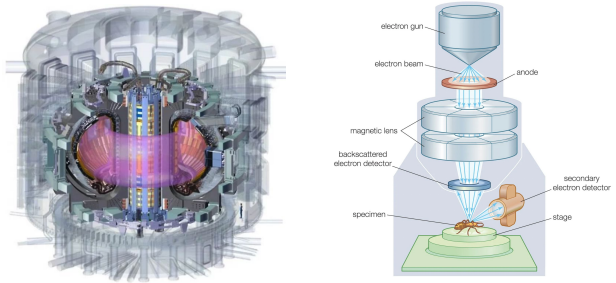
Materials design over large spaces of compounds.

Drug discovery



Target identification, lead optimization.

Large Science Machines



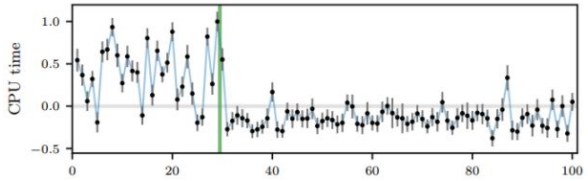
Optimize settings & controls for science use-cases.

[Kusne et al., 2020], [Eisenstein, 2020], [Fenstermacher et al., 2022]

Deep Uncertainty Models —Decision Making Under Uncertainty

ENGINEERING

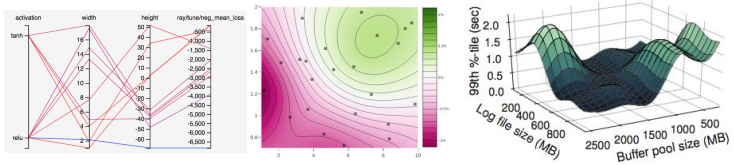
Computer Systems



OTTER TUNE

Monitor and tune configurations for performance (latency, throughput).

Machine Learning



Find and assess new methods, models, architectures, hyperparameters.

[Van Aken et al, 2017], [Balandat et al., 2020]

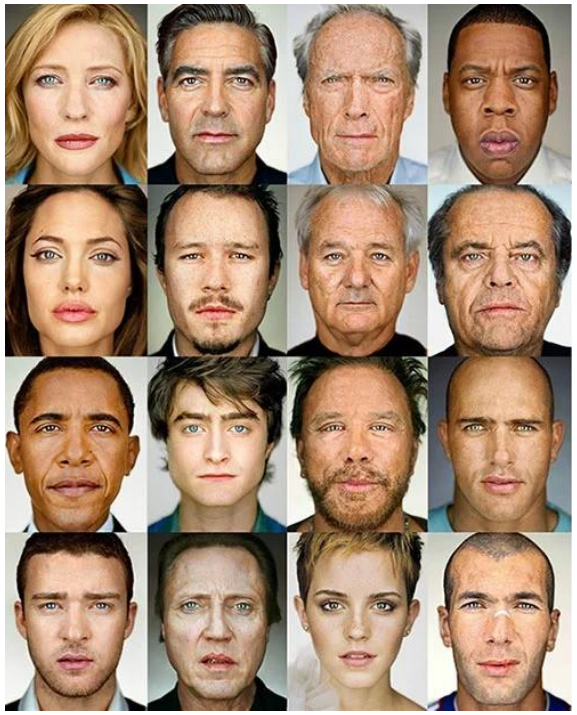
Probabilistic Modeling — High Dimensions

Often we want to model distributions of **high-dimensional data**.

Probabilistic Modeling — High Dimensions

Often we want to model distributions of **high-dimensional data**.

E.g., joint distributions over **image or video data**.



Martin Schoeller, Celebrity Portraits

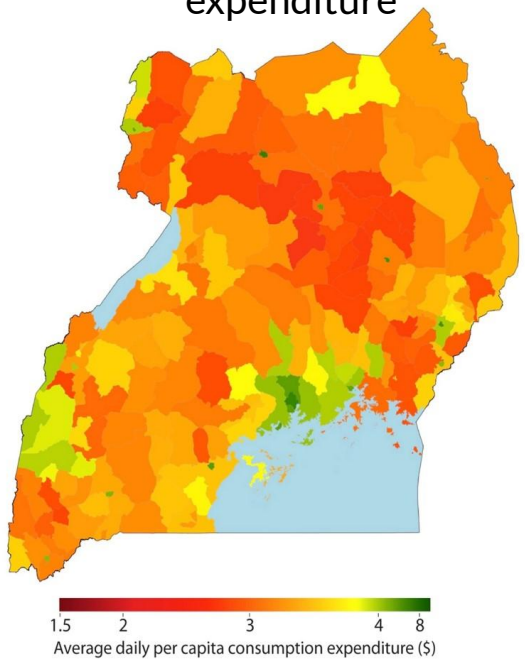
Kevin Beaty, "We made gif portraits of everyone we met on Colfax for 5 hours"

Probabilistic Modeling — High Dimensions

Often we want to model distributions of **high-dimensional data**.

E.g., joint distributions over **spatiotemporal data**.

Uganda, est. daily per-capita expenditure



Source: Stefano Ermon, Stanford University

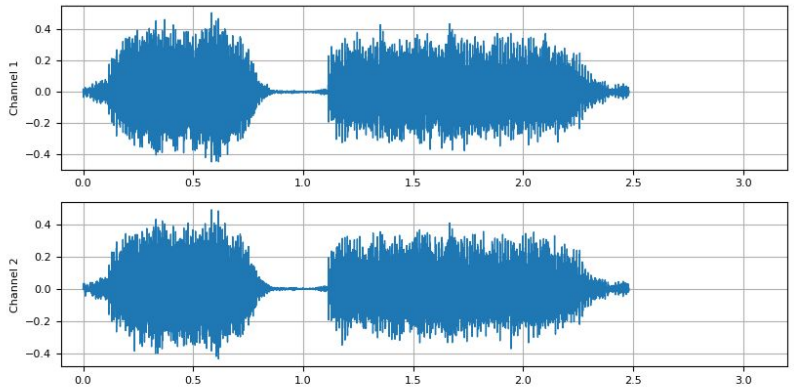
Seasonal Bird Migrations Trends (from eBird)

Source: Cornell Lab of Ornithology

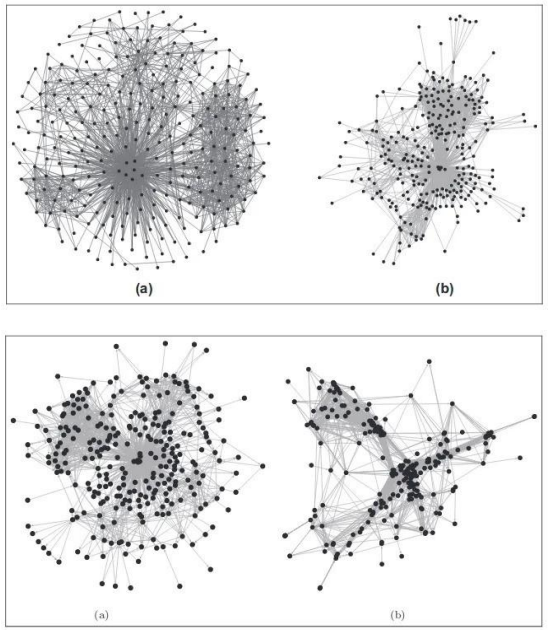
Probabilistic Modeling — High Dimensions

Often we want to model distributions of **high-dimensional data**.

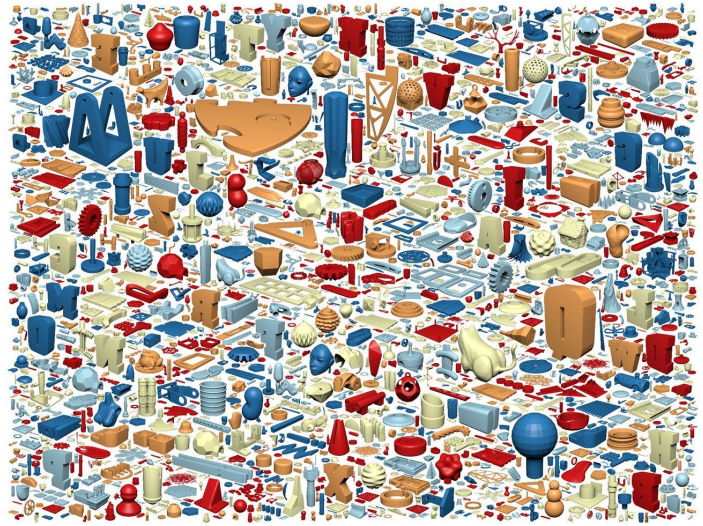
E.g., joint distributions over **audio, graph, 3d structure data**, and more.



Source: Torchaudio package



Source: Sviatoslav Kovalev, “Large Graph Visualization Tools and Approaches”



Source: Thingi10K dataset.

Probabilistic Modeling — High Dimensions

A few tasks for high-dimensional probabilistic modeling include:

Probabilistic Modeling — High Dimensions

A few tasks for high-dimensional probabilistic modeling include:

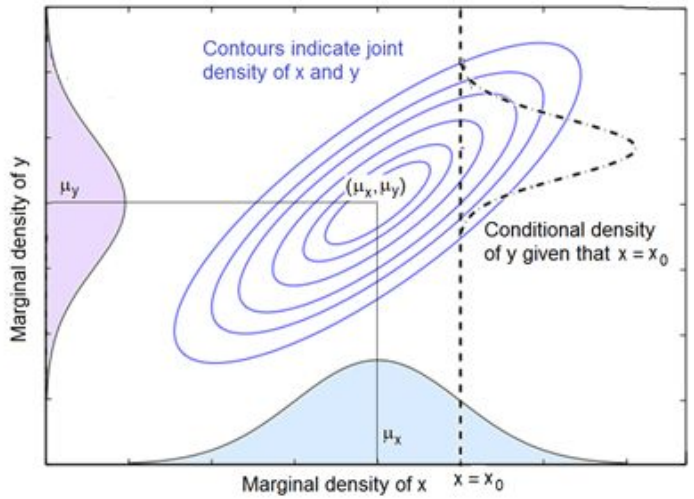
1. Representation: What is the best way to represent or approximate a distribution? Exact specification? Simplified approximation? Samples from distribution?

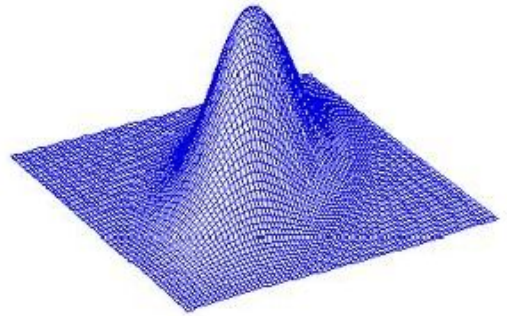
Probabilistic Modeling — High Dimensions

A few tasks for high-dimensional probabilistic modeling include:

1. Representation: What is the best way to represent or approximate a distribution? Exact specification? Simplified approximation? Samples from distribution?

2. Inference: How do we infer marginal or conditional distributions? Or statistics of these distributions?





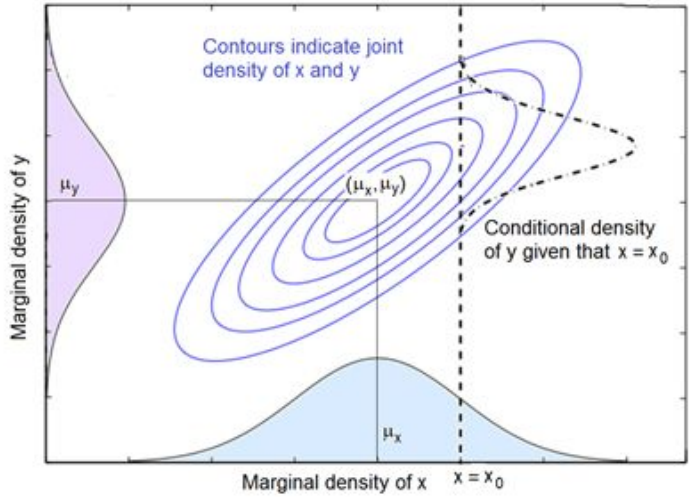
Probabilistic Modeling — High Dimensions

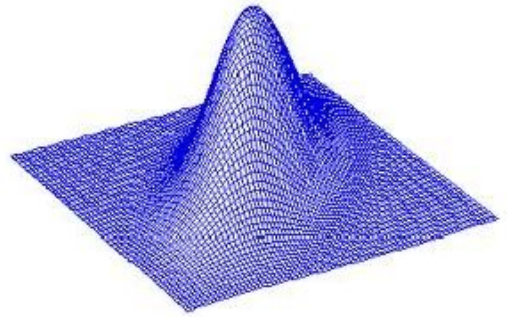
A few tasks for high-dimensional probabilistic modeling include:

1. Representation: What is the best way to represent or approximate a distribution? Exact specification? Simplified approximation? Samples from distribution?

2. Inference: How do we infer marginal or conditional distributions? Or statistics of these distributions?

3. Learning: How can we fit a model to a high-dimensional dataset?

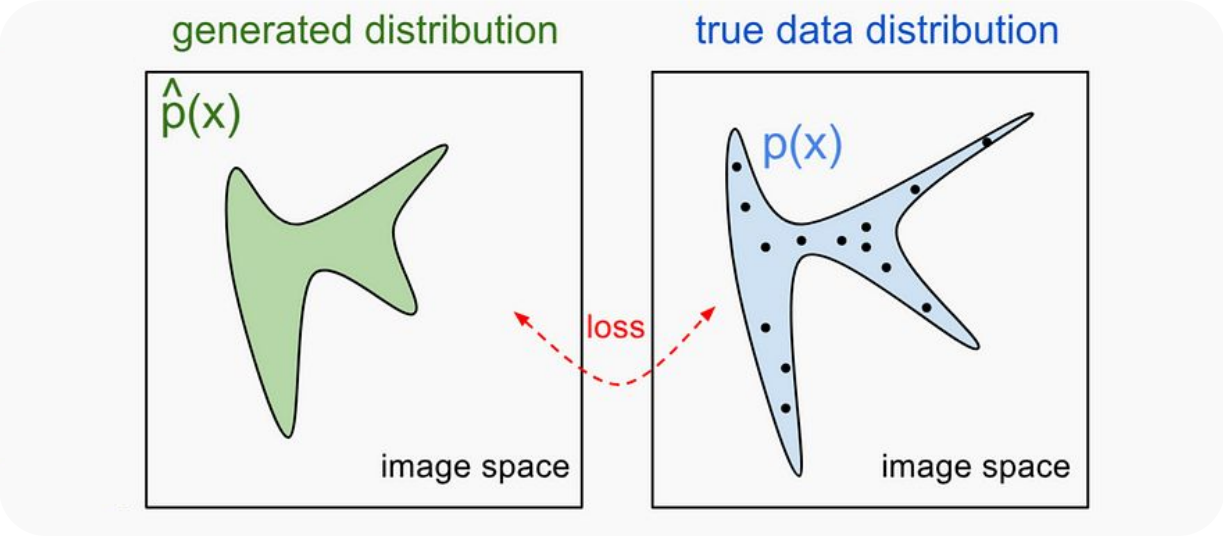




Probabilistic Models → Generative Models

4. Generation: After learning a model (distribution) from data, how can generate new samples from this distribution.

This leads us to the task of generative modeling.

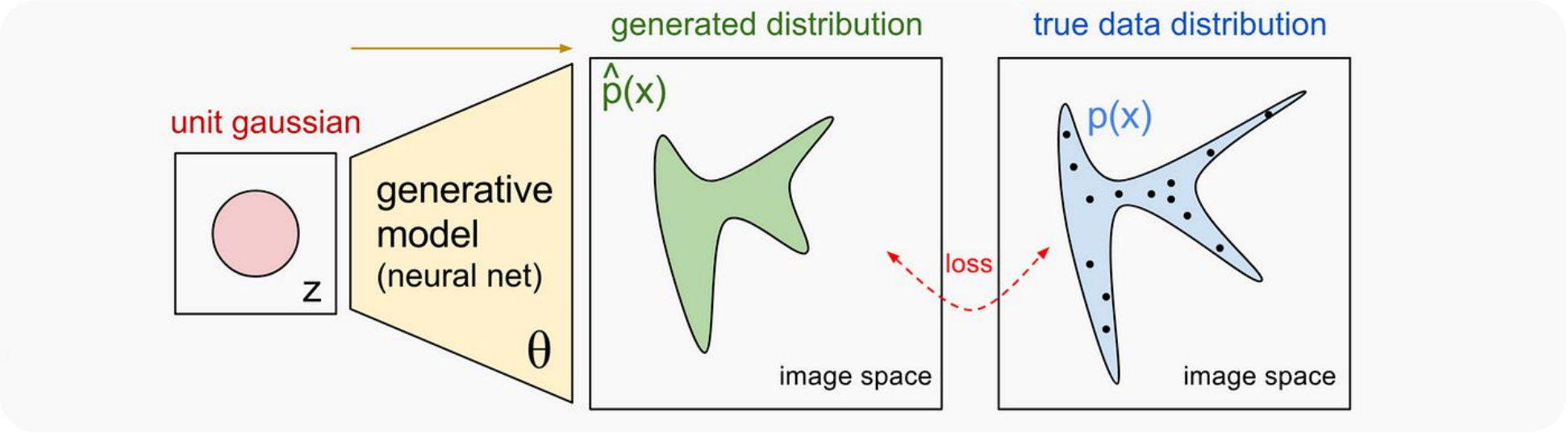


Source: OpenAI

Probabilistic Models → Generative Models

4. Generation: After learning a model (distribution) from data, how can generate new samples from this distribution.

This leads us to the task of ~~generative modeling~~ deep generative modeling.

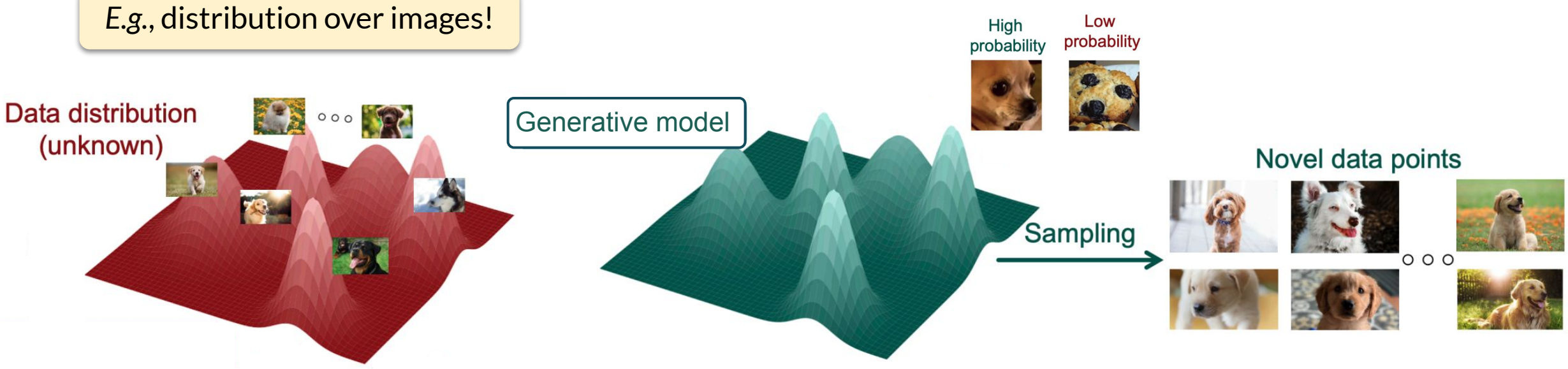


Source: OpenAI

Probabilistic Models → Generative Models

4. Generation: After learning a model (distribution) from data, how can generate new samples from this distribution.

E.g., distribution over images!



Source: Yang Song

Probabilistic Models → Generative Models

4. Generation: After learning a model (distribution) from data, how can generate new samples from this distribution.

Sometimes we aim to model/sample from:

- A marginal distribution, $p(x)$ — *e.g.*, a distribution over images.
- A condition distribution, $p(x|y)$ — *e.g.*, a distribution given a label y.

All probabilistic models!

Generative Models — Example Types

Generative Models Include:

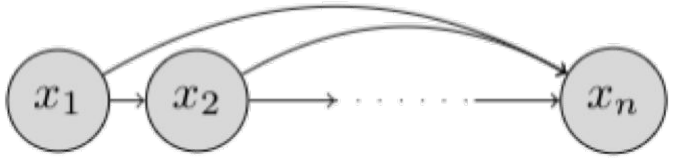
Generative Models — Example Types

Generative Models Include:

$$
\begin{aligned}
&p(x_1) \\
&p(x_2 \mid x_1) \\
&p(x_3 \mid x_2, x_1) \\
&p(x_4 \mid x_3, x_2, x_1) \\
&\ldots \\
&p(x_n \mid x_{n-1}, \ldots, x_2, x_1)
\end{aligned}
$$

Autoregressive Models
(e.g., LLMs)

Once upon a ... [EOS]



Generative Models — Example Types

Generative Models Include:

Diffusion and
Score-based Models

Flow-based Models
(e.g. NF, CNF, FM)

$$\mathcal{N}(x_1, x_2, x_3, \ldots, x_n \mid \mathbf{0}, \mathbf{I})$$
$$p_1(x_1, x_2, x_3, \ldots, x_n)$$
$$p_2(x_1, x_2, x_3, \ldots, x_n)$$
$$\ldots$$
$$p_n(x_1, x_2, x_3, \ldots, x_n)$$
$$p_{\text{data}}(x_1, x_2, x_3, \ldots, x_n)$$

Generative Modeling — Intuition for the Difficulty

It’s interesting to consider the difficulty of the problem at hand.

Generative Modeling — Intuition for the Difficulty

It’s interesting to consider the difficulty of the problem at hand.

Typical phone photo: 700 x 1400 pixels, each pixel in [0, 255], and (R, G, B) channels.

Generative Modeling — Intuition for the Difficulty

It's interesting to consider the difficulty of the problem at hand.

Typical phone photo: 700 x 1400 pixels, each pixel in [0, 255], and (R, G, B) channels.

⇒ Number of possible images: $256^{(700 \times 1400 \times 3)} \approx 10^{(7{,}000{,}000)}$ **WOW**

Generative Modeling — Intuition for the Difficulty

It's interesting to consider the difficulty of the problem at hand.

Typical phone photo: 700 x 1400 pixels, each pixel in [0, 255], and (R, G, B) channels.

⇒ Number of possible images: $256^{(700 \times 1400 \times 3)} \approx 10^{(7{,}000{,}000)}$ **WOW**

Number of atoms in universe is just 10^{80} :-).

Generative Modeling — Intuition for the Difficulty

It's interesting to consider the difficulty of the problem at hand.

Typical phone photo: 700 x 1400 pixels, each pixel in [0, 255], and (R, G, B) channels.

⇒ Number of possible images: $256^{(700 \times 1400 \times 3)} \approx 10^{(7{,}000{,}000)}$ **WOW**

Number of atoms in universe is just 10^{80} :-).

More importantly, number of images in ImageNet is 10^7, and LAION is 10^9.

Generative Modeling — Intuition for the Difficulty

It's interesting to consider the difficulty of the problem at hand.

Typical phone photo: 700 x 1400 pixels, each pixel in [0, 255], and (R, G, B) channels.

⇒ Number of possible images: 256^(700 x 1400 x 3) ≈ 10^(7,000,000) **WOW**

Number of atoms in universe is just 10^80 :-).

More importantly, number of images in ImageNet is 10^7, and LAION is 10^9.

⇒ Learning a generative model is a **highly underdetermined problem**!

Generative Models — Many Applications

And yet we have methods that work!

Generative Models — Many Applications

Applications include:

Generative Models — Many Applications

Applications include: **Image Generation**

Generative Models — Many Applications

Applications include: **Image Generation**

Source: Ian Goodfellow, 2019, Twitter
https://x.com/goodfellow_ian/status/1084973596236144640

Generative Models — Many Applications

Applications include: **Image Generation** … and in 2024:

Sources: Flux (Black Forest Labs) and Midjourney

Generative Models — Many Applications

Applications include: **Image Generation** … and in 2024:

Sources: Flux (Black Forest Labs) and Midjourney

Generative Models — Many Applications

Applications include: **Video Generation** (more recently)

Generative Models — Many Applications

Applications include: **Video Generation** (more recently)

Sources: Sora (OpenAI) and Pika Labs.

$$p(\text{next image} \mid \text{previous images})$$

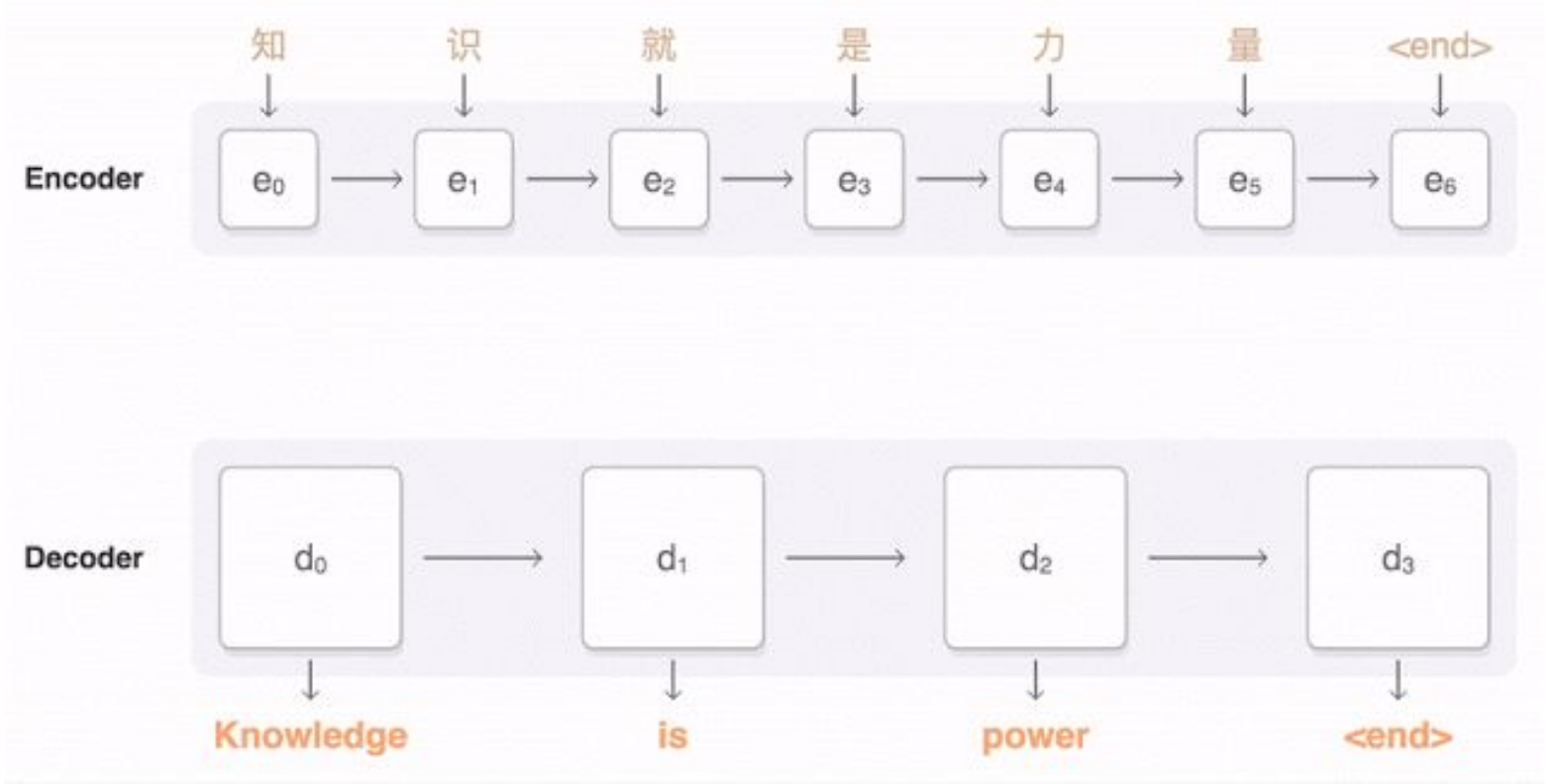
Generative Models — Many Applications

Applications include: **Text Generation**

Generative Models — Many Applications

Applications include: **Text Generation** $p(\text{next word} \mid \text{previous words})$

2017: Machine Translation with Seq2seq via RNN and attention mechanism

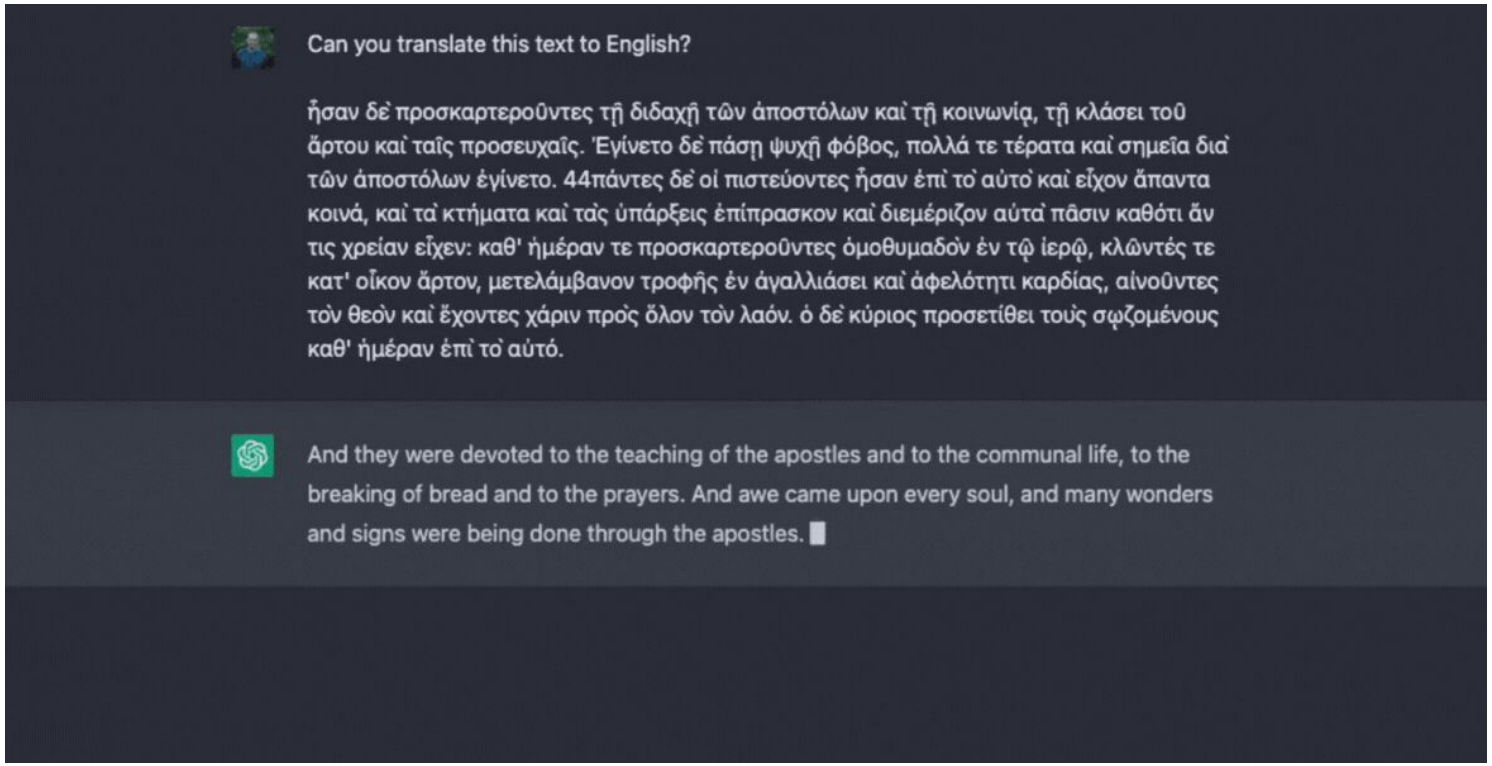


Source: Google

Generative Models — Many Applications

Applications include: **Text Generation** $p(\text{next word} \mid \text{previous words})$

2024: ChatGPT from OpenAI



Source: Bureauworks

Generative Models — Many Applications

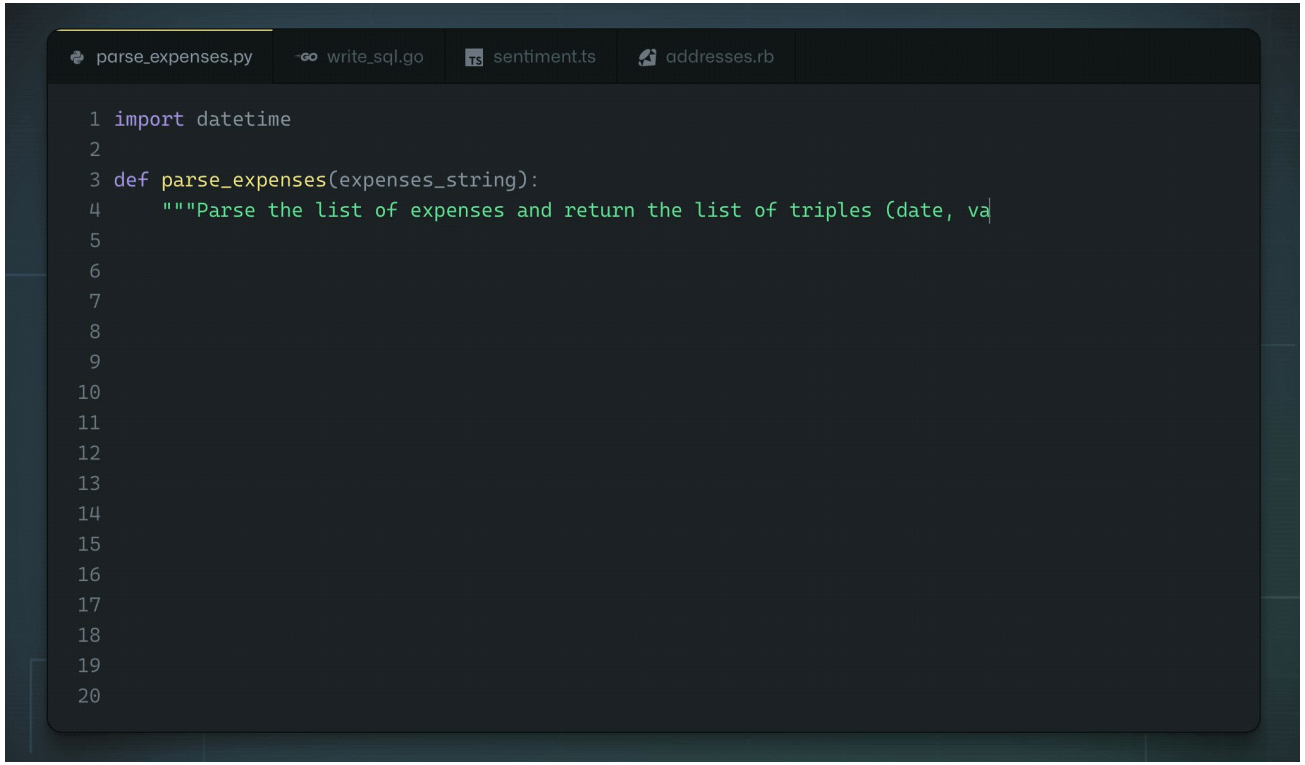
Applications include: **Code Generation**

Generative Models — Many Applications

Applications include: **Code Generation**

$$p(\text{next word} \mid \text{previous words})$$

2021: OpenAI Codex → GitHub Copilot



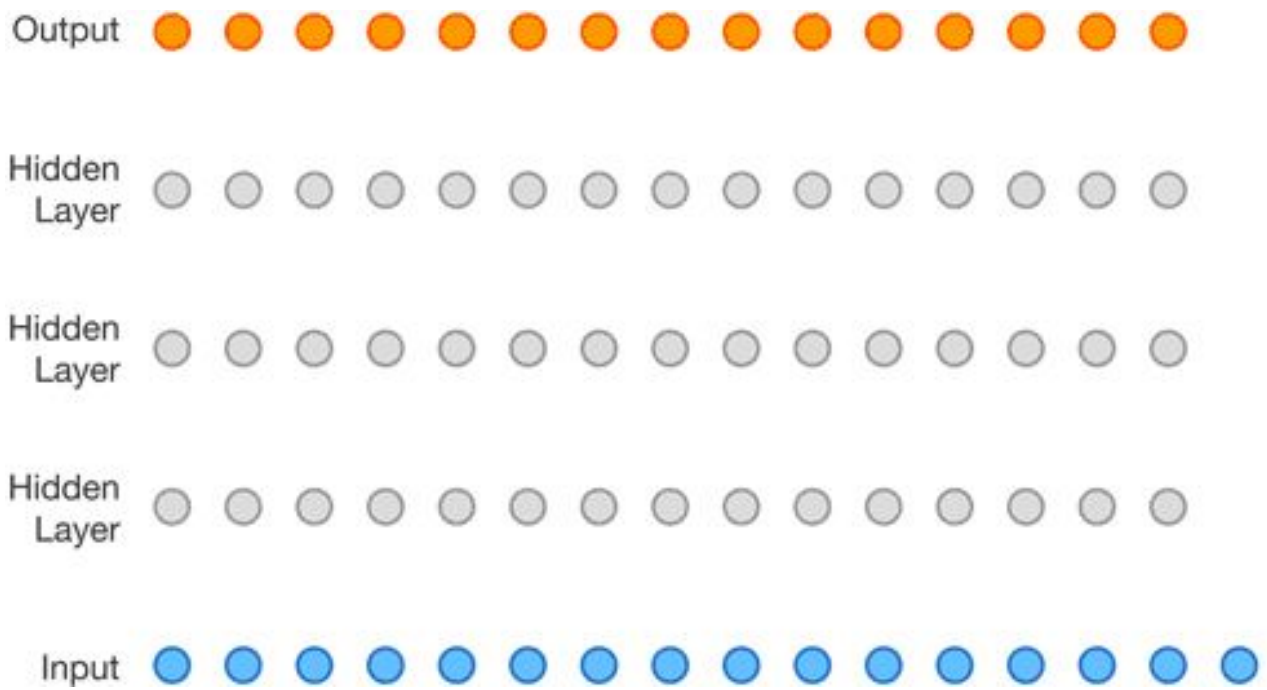
Source: OpenAI/Github Codex

Generative Models — Many Applications

Applications include: **Speech Generation**

Generative Models — Many Applications

Applications include: **Speech Generation** (Text to Speech) $p(\text{speech} \mid \text{text})$



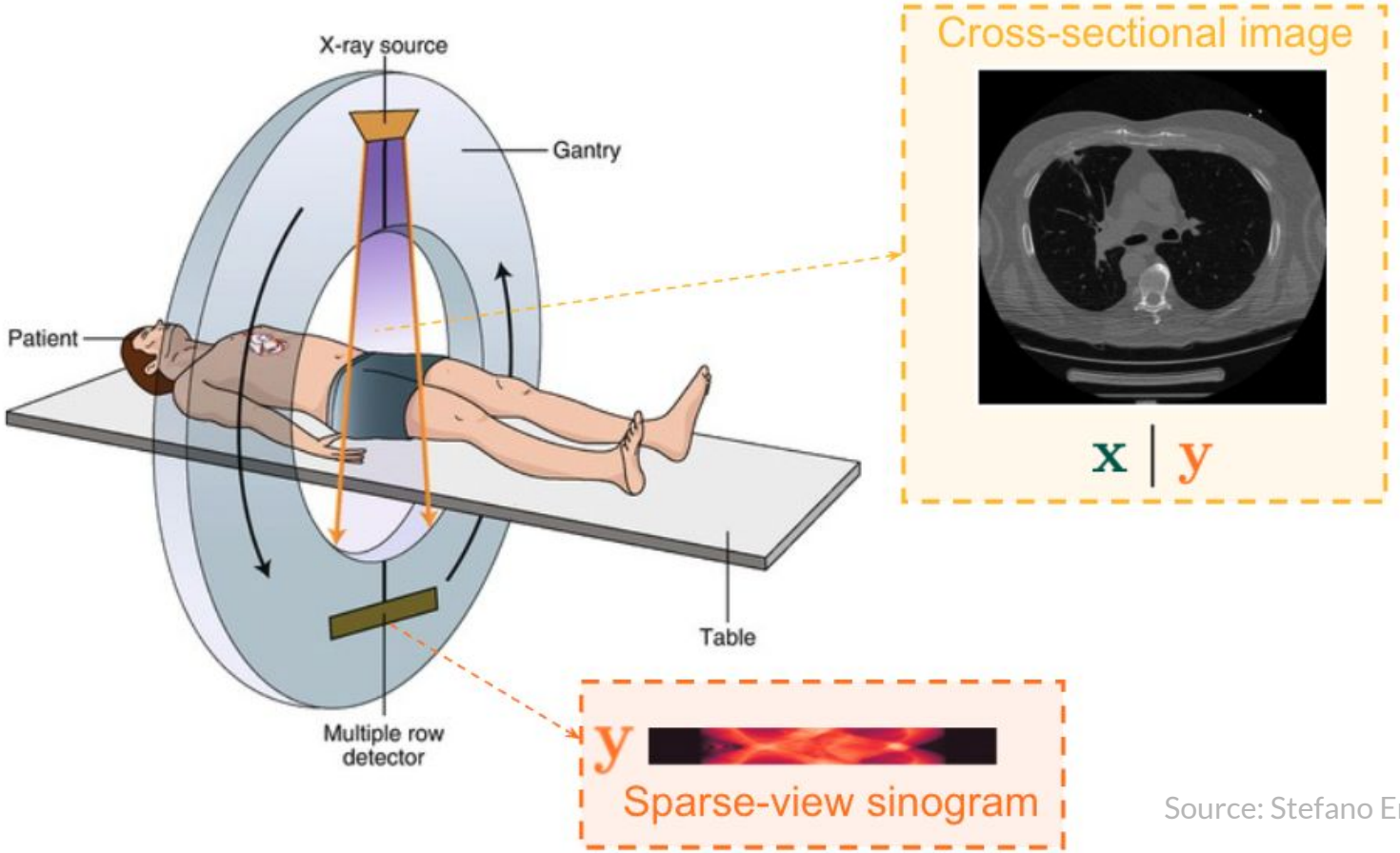
WaveNet, 2016 (Google DeepMind)

Generative Models — Many Applications

Applications include: **Medical Image Reconstruction**

Generative Models — Many Applications

Applications include: **Medical Image Reconstruction** $p(\text{image} \mid \text{X-ray signal})$



Source: Stefano Ermon, Deep Generative Models

Generative Models — Many Applications

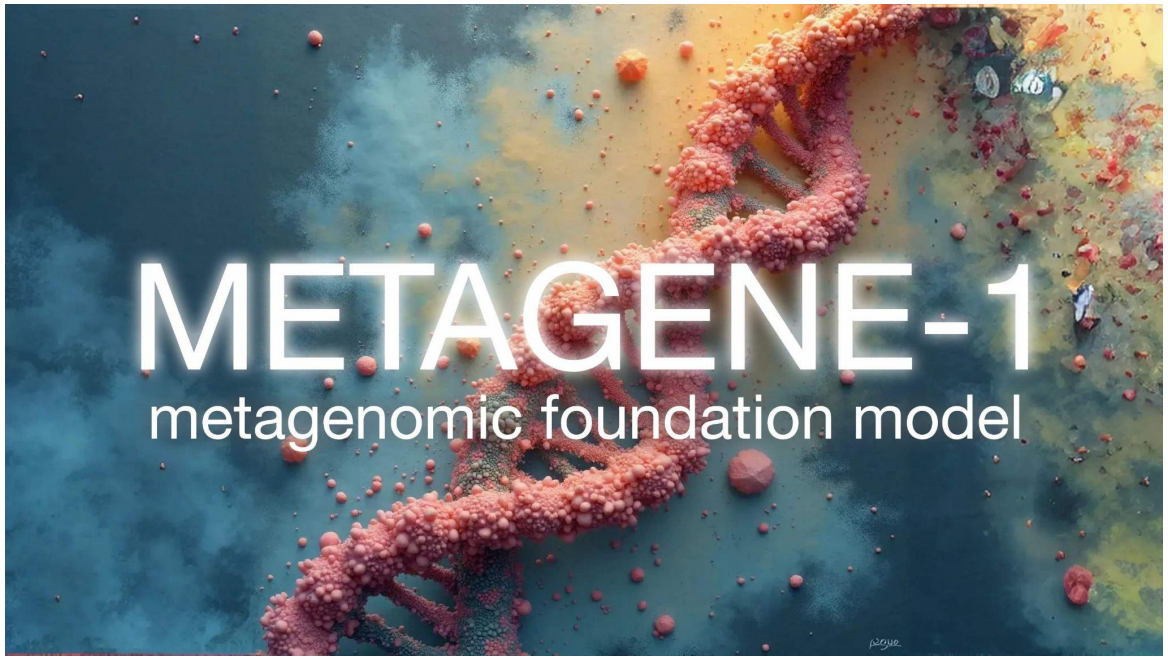
Applications include: **Science**

Generative Models — Many Applications

Applications include: **Science** — DNA, RNA, etc. $p(\text{next DNA} \mid \text{previous DNA})$

Generative Models — Many Applications

Applications include: **Science** — DNA, RNA, etc. $p(\text{next DNA} \mid \text{previous DNA})$



Trained on DNA/RNA.

Autoregressive model - like LLMs.

7B parameters (~small LLMs).

Trained on sequenced wastewater.

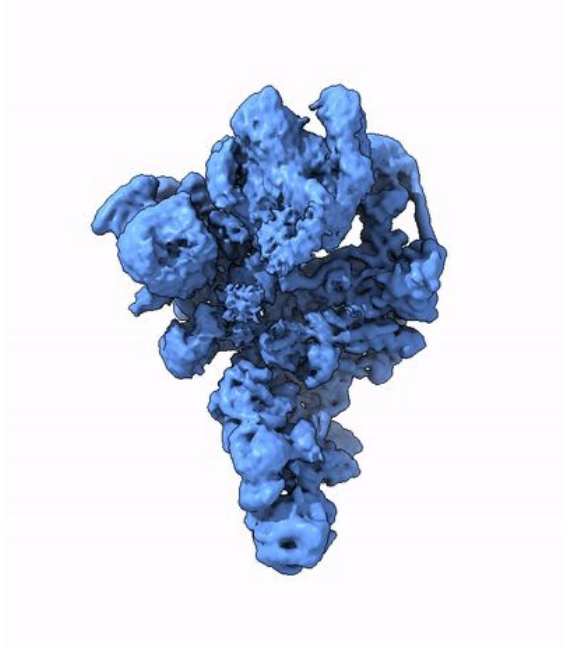
Dataset of 1.5 trillion base pairs.

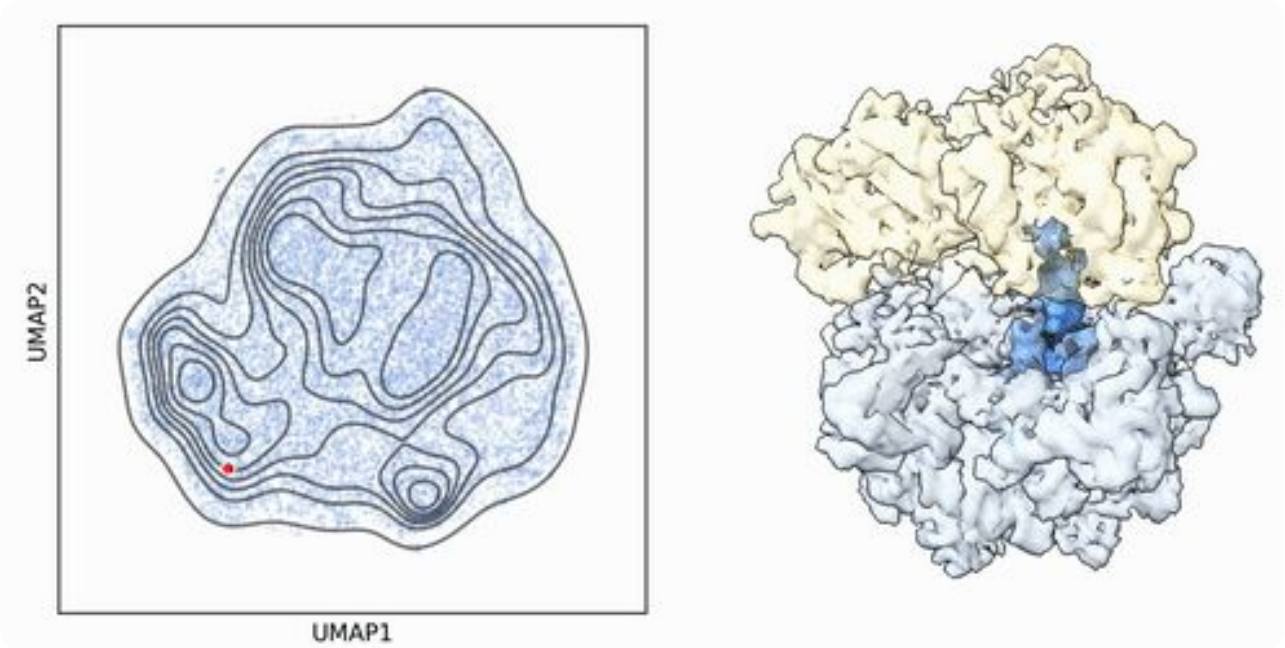
https://metagene.ai

Source: METAGENE-1: Metagenomic Foundation Model for Pandemic Monitoring

Generative Models — Many Applications

Applications include: **Science** — Cryo-EM for 3D Protein Structure Modeling





$$p(\text{3D structure} \mid \text{cryo-EM image})$$

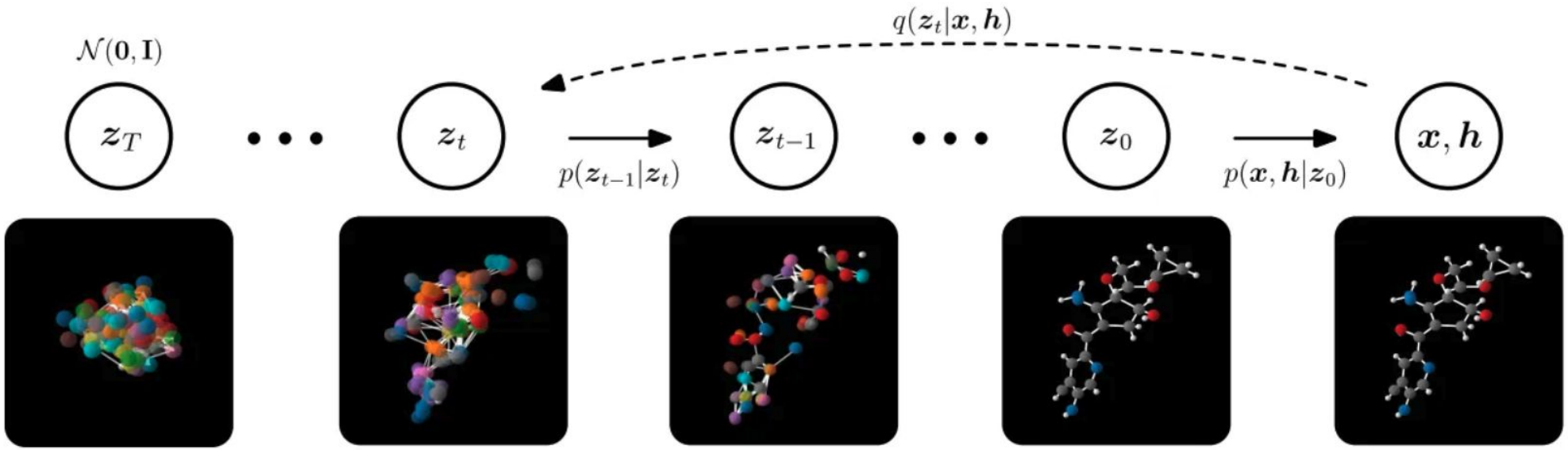
Sources:

CryoDRGN: reconstruction of heterogeneous cryo-EM structures using neural networks.

CryoDRGN-ET: deep reconstructing generative networks for visualizing dynamic biomolecules inside cells.

Generative Models — Many Applications

Applications include: **Science** — Molecular generation in chemistry



$p(\text{molecules})$

Sources: Equivariant Diffusion for Molecule Generation in 3D

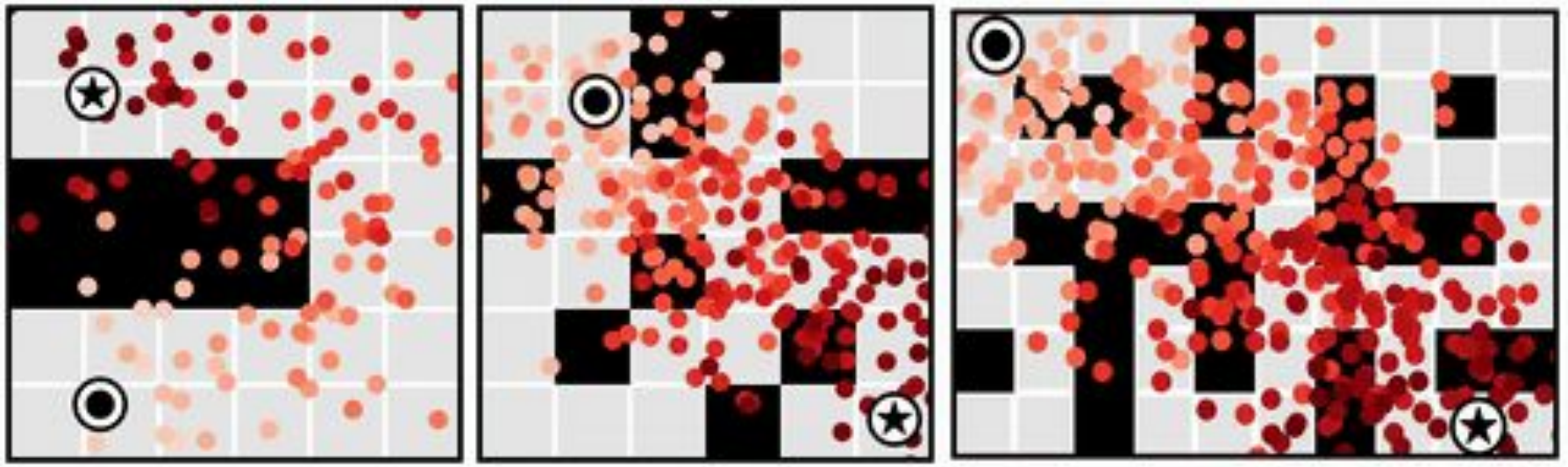
Generative Models — Many Applications

Applications include: **Actions & Planning**

Generative Models — Many Applications

Applications include: **Actions & Planning**

Diffusion for planning $p(\text{action sequence} \mid \text{environment map})$



Source: Planning with Diffusion for Flexible Behavior Synthesis

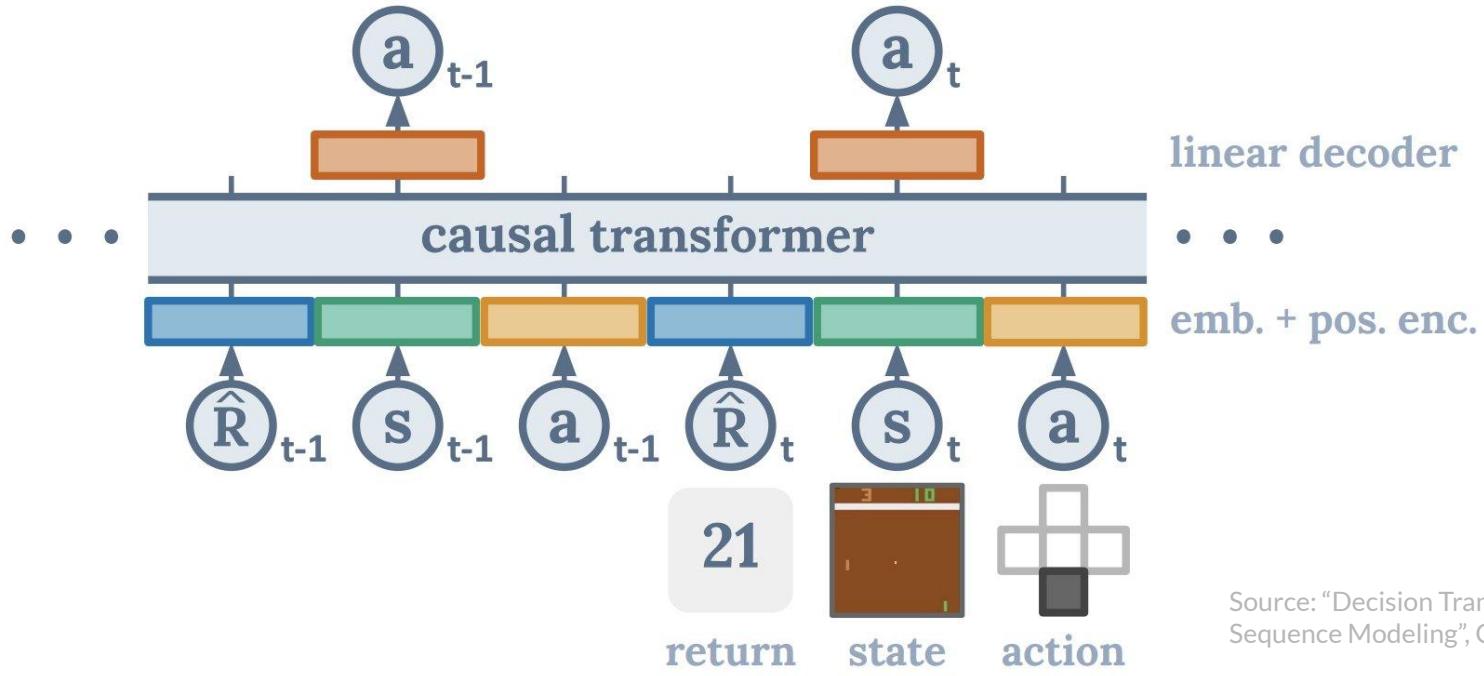
Generative Models — Many Applications

Applications include: **Decision Making**

Generative Models — Many Applications

Applications include: **Decision Making**

Decision Transformer — autoregressive model: $p(\text{states}, \text{actions}, \text{returns})$



Source: "Decision Transformer: Reinforcement Learning via Sequence Modeling", Chen at al., 2021

Generative Models — Many Applications

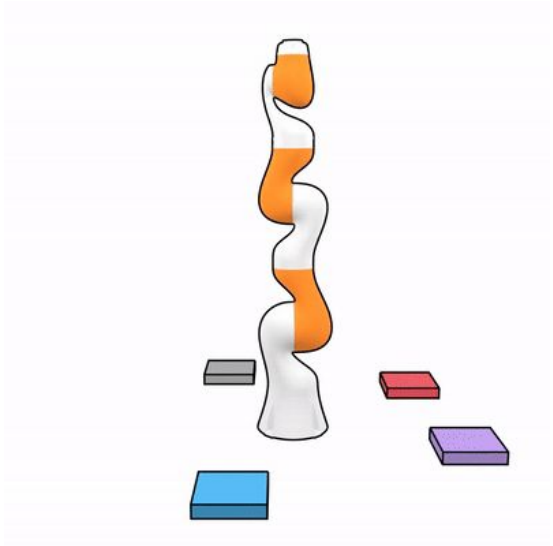
Applications include: **Robotics**

Generative Models — Many Applications

Applications include: **Robotics**

$$p\left(\text{actions} \mid \text{images}, \text{text}\right)$$

Diffusion planning for robotics



Source: Planning with Diffusion for Flexible Behavior Synthesis.

Vision-language-action model

Source: Physical Intelligence π0.

Generative Models — Many Applications

Applications include: **Conditional Generative Models** — for other tasks…

Generative Models — Many Applications

Applications include: **Conditional Generative Models** — for other tasks…

$$p\,(\text{zebra images} \mid \text{horse images})$$

Source: Andrej Risteski

Generative Models — Many Applications

Applications include: **Style Transfer** — *e.g.*, on images

Generative Models — Many Applications

Applications include: **Style Transfer** — *e.g.*, on images

$p(\text{Monet painting} \mid \text{input image})$ $p(\text{Van Gogh painting} \mid \text{input image})$

Input Image

Monet

Van Gogh

Source: Andrej Risteski

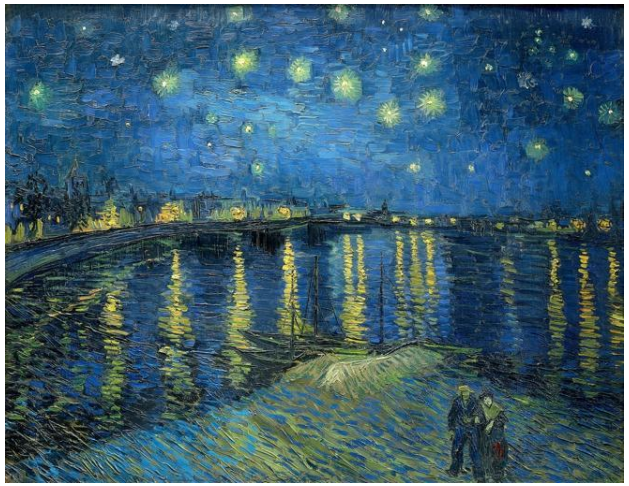
Generative Models — Many Applications

Applications include: **Actions for Art Generation**

Generative Models — Many Applications

Applications include: **Actions for Art Generation**

Inverse painting: $p(\text{paint strokes} \mid \text{painting})$



Source: Inverse painting project

SketchODE: $p(\text{pen strokes} \mid \text{text})$

Source: SketchODE project.

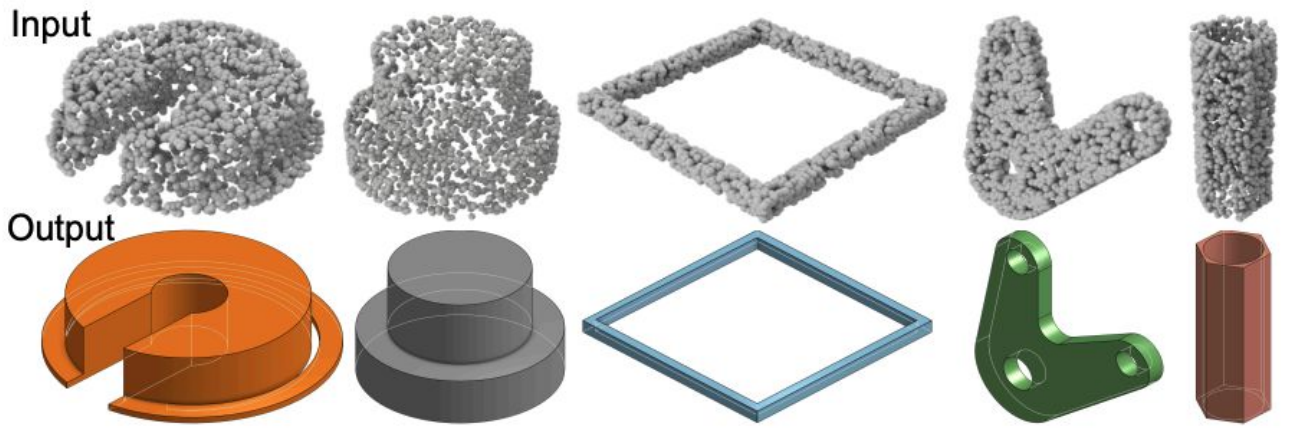
Generative Models — Many Applications

Applications include: **Generation for CAD (computer-aided design)**

Generative Models — Many Applications

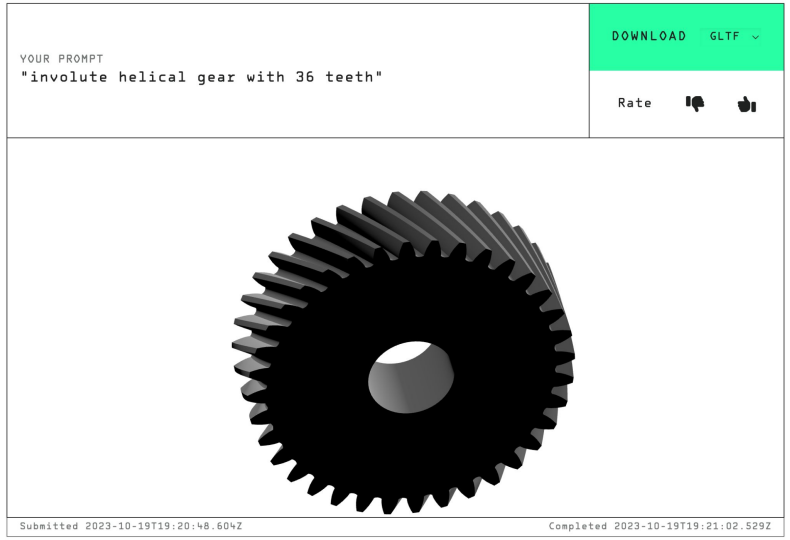
Applications include: **Generation for CAD (computer-aided design)**

Point-cloud to CAD



Source: DeepCAD: A Deep Generative Network for Computer-Aided Design Models

Text to CAD



Source: zoo.dev

Generative Models — Downstream Tasks

After learning a generative model, often want to carry out tasks such as:

Generative Models — Downstream Tasks

After learning a generative model, often want to carry out tasks such as:

1. **Sampling:** How can we generate novel data from the model distribution, *i.e.*, $x_{\text{new}} \sim p_\theta(x)$?

2. **Density estimation:** Given a datapoint x, what is the probability density assigned by the model, $p_\theta(x)$.

3. **Unsupervised representation learning:** How can we learn meaningful feature representations for a datapoint x? (And then use them!)

We will cover these topics in this class.

Instructor and Teaching Assistant

Willie Neiswanger
Instructor

Oliver Liu
Teaching Assistant

Instructor and Teaching Assistant

Oliver Liu
Teaching Assistant

Oliver Liu

(My PhD student).

Works on:

- Large multimodal models,
- scientific discovery,
- reasoning and decision-making.

About me — some background context

Willie Neiswanger

Instructor

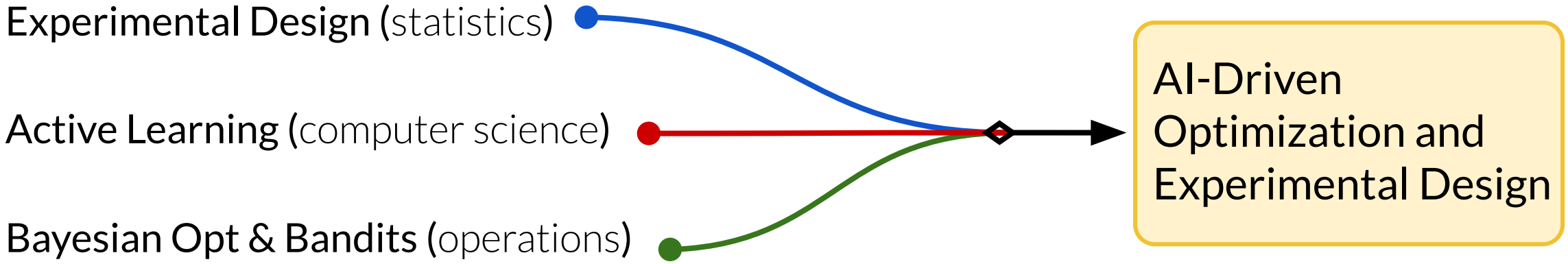
About me — some background context

In PhD and postdoc: *Probabilistic modeling with application to science and engineering.*

About me — some background context

In PhD and postdoc: *Probabilistic modeling with application to science and engineering.*

Use techniques from:

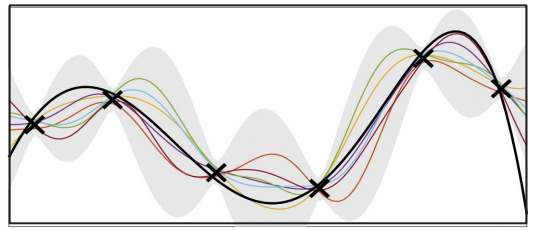


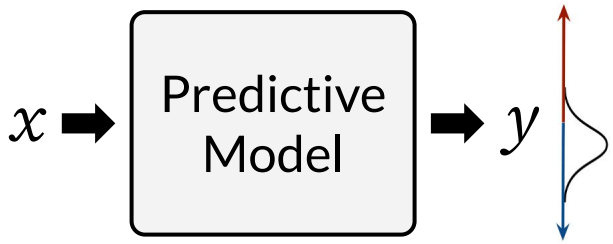
Focus on combining probabilistic machine learning with decision making.

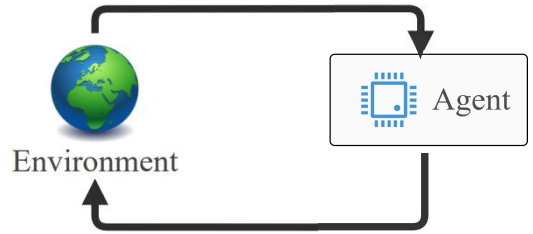
About me — some background context

Probabilistic machine learning

- Classic probabilistic models
- Generative models







About me — some background context

Probabilistic machine learning

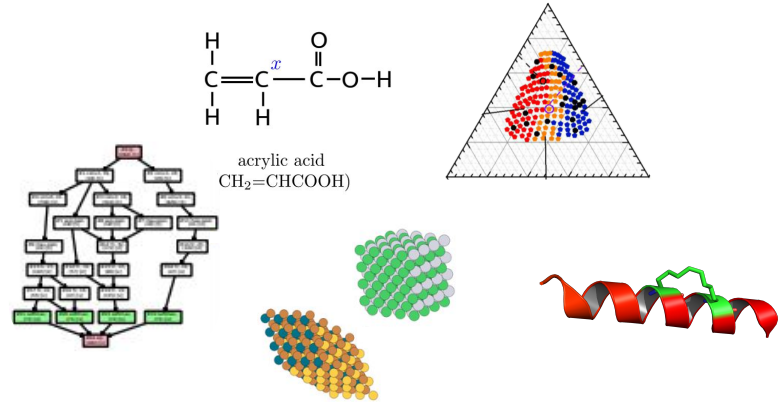
- Classic probabilistic models
- Generative models

Applications to science and engineering:

About me — some background context

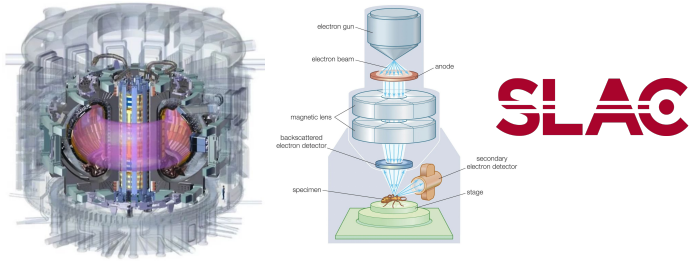
Probabilistic machine learning

- Classic probabilistic models
- Generative models



Applications to science and engineering:

- Black-box optimization & experimental design:
  - Materials science (high-throughput screening).
  - Scientific machines (particle accelerators, tokamaks).
  - Computer/ML systems (config. tuning, hyperparameter opt).



About me — some background context

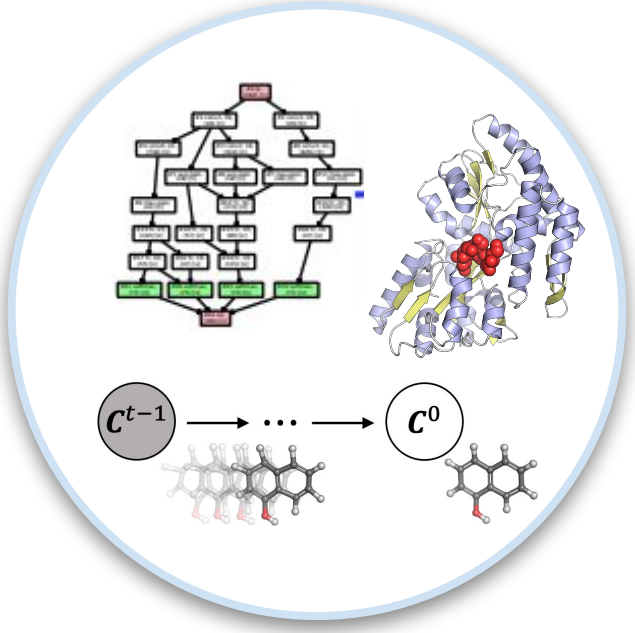
Probabilistic machine learning

- Classic probabilistic models
- Generative models

Applications to science and engineering:

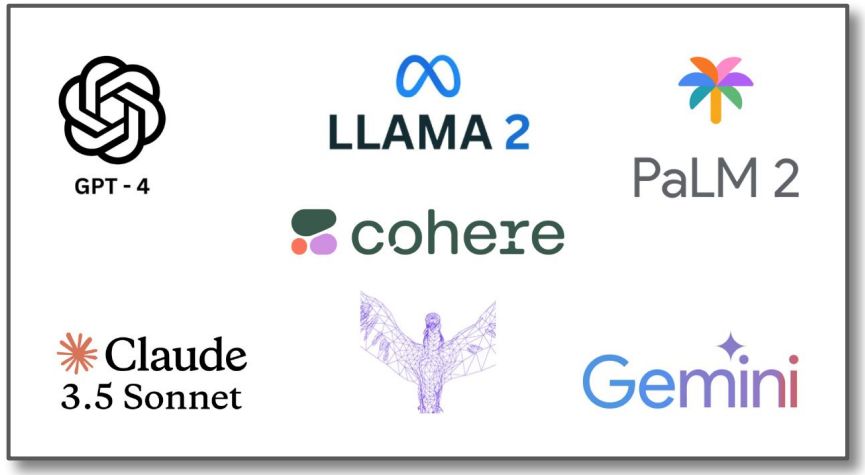
- Black-box optimization & experimental design:
  - Materials science (high-throughput screening).
  - Scientific machines (particle accelerators, tokamaks).
  - Computer/ML systems (config. tuning, hyperparameter opt).

- Generative + probabilistic modeling:
  - Chemical / molecular design
  - Neural architecture design

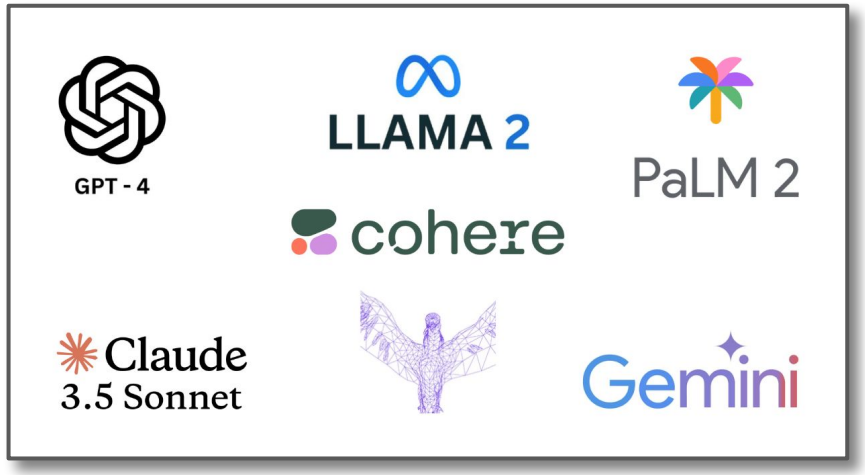


At the end of my postdoc (~mid 2023)

Some collaborators began a project on open source reproductions of LLMs …



At the end of my postdoc (~mid 2023)



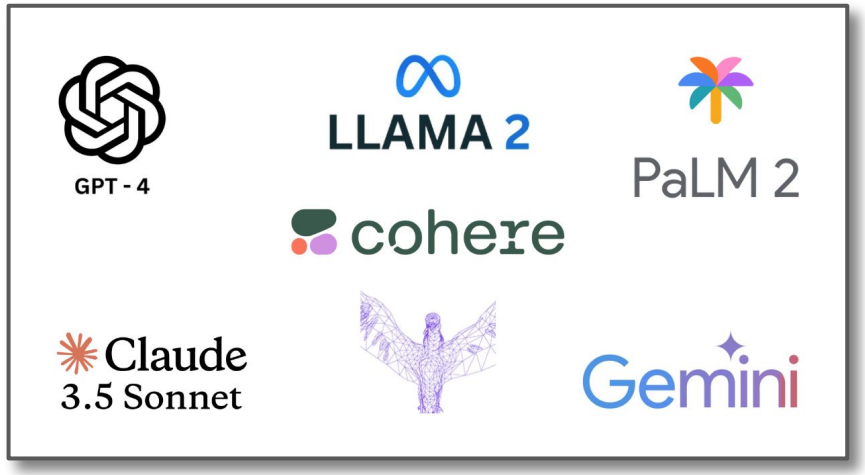
Some collaborators began a project on open source reproductions of LLMs …

Through support from some universities/companies they got resources to do LLM pretraining.

⇒ I thought it would be a good opportunity to learn

⇒ Started working on large generative models.

At the end of my postdoc (~mid 2023)



Some collaborators began a project on open source reproductions of LLMs …

Through support from some universities/companies they got resources to do LLM pretraining.

⇒ I thought it would be a good opportunity to learn

⇒ Started working on large generative models.

Increasingly working on LLM & large generative modeling projects for past ~2 years.

- Including LLM/foundation model pretraining, more recently diffusion/flow models.

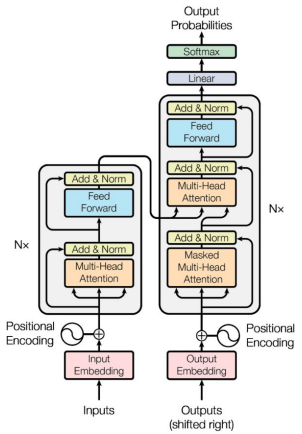
My current research

⇒ It influenced my research!

I still do probabilistic modeling and decision making for science, but…

I also combine this with large generative models, e.g.

- Generative/foundation models for scientific data (*e.g.*, in biology, materials).
- LLMs for (sequential) decision making, and it's variants.



Syllabus: Goals, Assignments, Grading, and More

Course Description & Learning Objectives

This course focuses on probabilistic models and their central role within modern machine learning and generative modeling.

We aim to cover topics including:

Course Description & Learning Objectives

This course focuses on probabilistic models and their central role within modern machine learning and generative modeling.

We aim to cover topics including:

- **Probabilistic graphical models** (*e.g.*, Bayesian networks) and **approximate inference** algorithms (*e.g.*, MCMC and variational inference).

Course Description & Learning Objectives

This course focuses on probabilistic models and their central role within modern machine learning and generative modeling.

We aim to cover topics including:

- **Probabilistic graphical models** (*e.g.*, Bayesian networks) and **approximate inference** algorithms (*e.g.*, MCMC and variational inference).
- **Deep generative models** (*e.g.*, autoregressive, score-matching, diffusion, and flow-based approaches).

Course Description & Learning Objectives

This course focuses on probabilistic models and their central role within modern machine learning and generative modeling.

We aim to cover topics including:

- **Probabilistic graphical models** (*e.g.*, Bayesian networks) and **approximate inference** algorithms (*e.g.*, MCMC and variational inference).
- **Deep generative models** (*e.g.*, autoregressive, score-matching, diffusion, and flow-based approaches).
- **Uncertainty quantification** in supervised/deep learning (Gaussian processes, Bayesian neural networks, ensembles, Monte Carlo dropout, UQ in LLMs)

Course Description & Learning Objectives

This course focuses on probabilistic models and their central role within modern machine learning and generative modeling.

We aim to cover topics including:

- **Probabilistic graphical models** (*e.g.*, Bayesian networks) and **approximate inference** algorithms (*e.g.*, MCMC and variational inference).
- **Deep generative models** (*e.g.*, autoregressive, score-matching, diffusion, and flow-based approaches).
- **Uncertainty quantification** in supervised/deep learning (Gaussian processes, Bayesian neural networks, ensembles, Monte Carlo dropout, UQ in LLMs)
- Methods for probabilistic **model-based sequential decision-making** (*e.g.*, Bayesian optimization, and information-based experimental design)

Preparation and Prerequisites

This course is designed for students currently pursuing research, or who wish to pursue research, in probabilistic machine learning or deep generative models!

- A major component will be carrying out a project relevant to your research interests.
- Also: reading and presenting research papers on probabilistic/generative models.

Preparation and Prerequisites

This course is designed for students currently pursuing research, or who wish to pursue research, in probabilistic machine learning or deep generative models!

- A major component will be carrying out a project relevant to your research interests.
- Also: reading and presenting research papers on probabilistic/generative models.

Students are expected to be comfortable with reading and presenting machine learning conference papers (*e.g.*, NeurIPS, ICLR, ICML).

Beneficial to have familiarity with machine learning, algorithms, and probability.

However, there are no formal (required) class prerequisites.

Typical Class Structure

First half (1.5 - 2 hours) — Lecture from me on course materials.

- Core lesson on the topic of the day.
- [Periodically] “Lab session”: implementation or programming lesson related to topic.
- Also aim to review some classic papers on the topic (+ give some history of the field).

Typical Class Structure

Second half — Activities related to paper presentations, course project, and more.

- Student paper presentations on probabilistic/generative ML papers.
- Small-group meetings for feedback on projects.
- Project pitches, and project final presentations.

Laptops in Class — Recommended

You are welcome to use a laptop in this class (though not required), *e.g.*, for

- Taking notes during lectures.
- (If interested) following along with implementation/lab session.
- Sometimes will be useful for roles during paper presentations (e.g., *scribe).*
- Sometimes will be useful for other tasks during course projects (e.g., sign-up for presentation schedule).

Class Website

Course website URL:
willieneis.github.io/probgen-spring2025

Please check it out for:

- Class news and updates.
- Lecture schedule & topics.
- Reminders of assignments, grading policies.

CSCI 699: Probabilistic and Generative Models, Spring 2025

Spring 2025, Fridays at 2:00-5:20pm in KAP 144
Instructor: Willie Neiswanger

This course focuses on probabilistic models and their central role within modern machine learning and generative modeling. With probabilistic methods increasingly driving advancements in AI, this course will explore its applications across a range of topics, including , approximate inference algorithms (MCMC, variational inference), deep generative models (autoregressive, score-matching, diffusion, and flow-based models), and model-based sequential decision making.

Course Staff

Willie Neiswanger
Instructor
Office hours: Friday 5-6:30pm
Location: TBD

Oliver Liu
Teaching Assistant
Office hours: Day-Time and Day-Time
Location: TBD

Logistics

- **Assignments**: Submit all written assignments, including all project-related write-ups, on Brightspace. Grades and feedback will also be provided on Brightspace.
- **General discussion**: Please use the official course Slack channel for general questions.
- **Other discussion**: Email Willie and Oliver (`neiswang@` and `zliu2898@`) or come to office hours to discuss individual matters, such as project ideas or grading.

Prerequisites

This course is designed for students currently pursuing research, or who wish to pursue research, in probabilistic machine learning or deep generative models. It will be beneficial to have familiarity with machine learning (at the level of CSCI 567), algorithms (at the level of CSCI 570), and probability (at the level of MATH 505a). Students are expected to be comfortable with reading and presenting modern machine learning conference papers.

Schedule

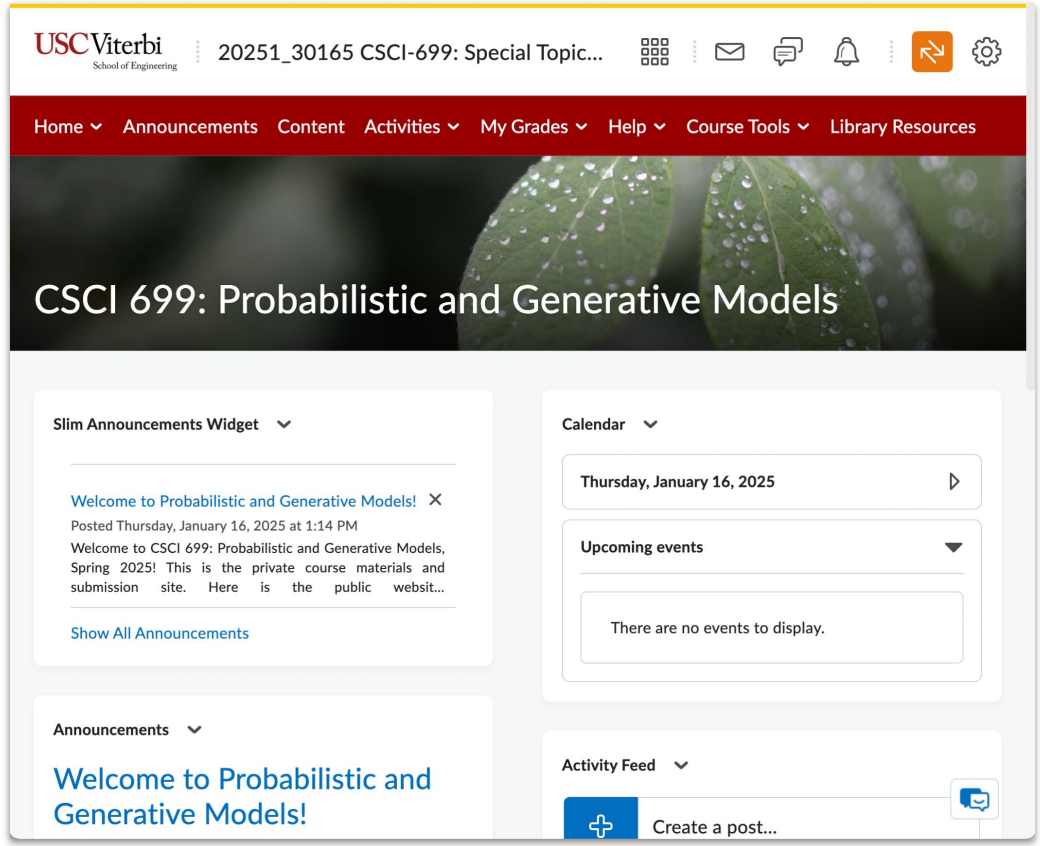
Course Submission Site

We'll be using Brightspace - learning management system.

URL: https://brightspace.usc.edu/

Mainly used for

- Submitting assignments (reports for class project, scribe notes, etc),
- Internal course content, documents, etc.
- In general: anything internal to the class that we do not intend to (immediately) make public.



Communications with Instructor/TA

For individual matters, send us an **email**:
neiswang@usc.edu, **zliu2898@usc.edu**

- (You are also welcome to direct message us on **Slack** via your USC Slack account)

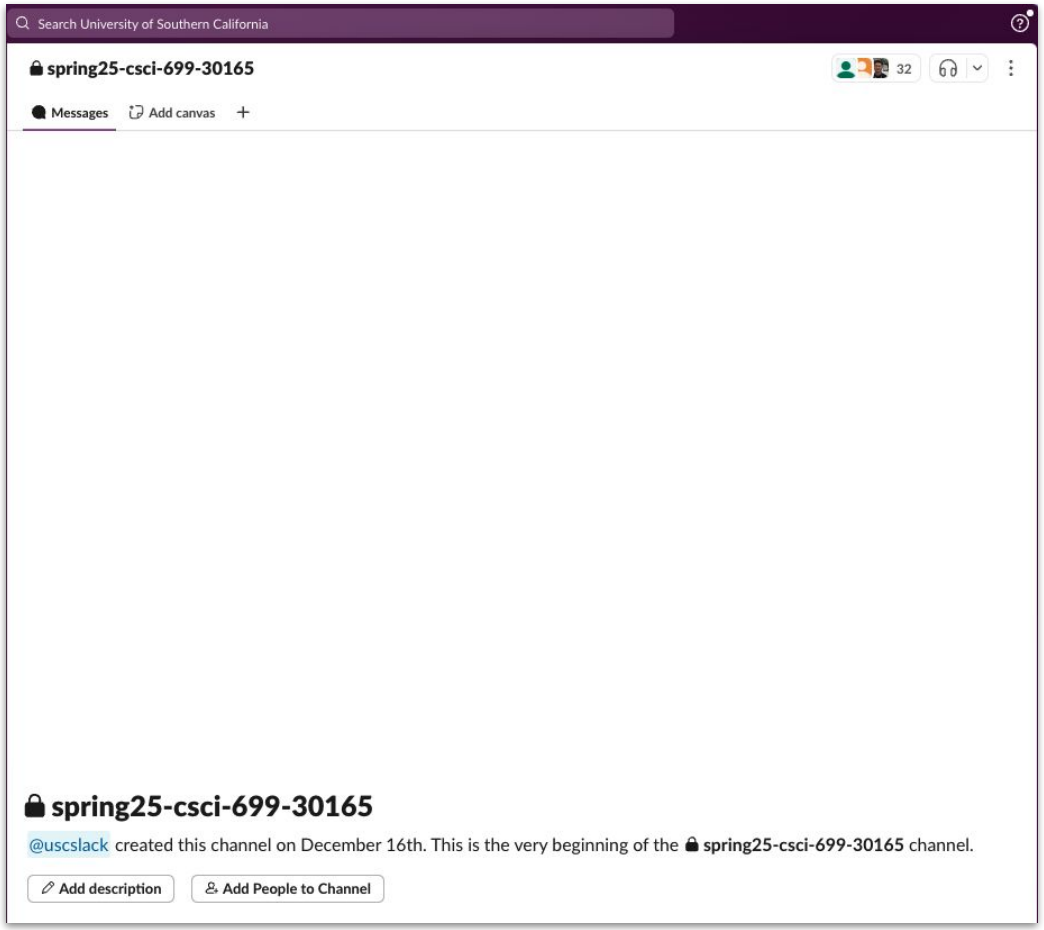
Communications with Instructor/TA

For individual matters, send us an **email**:
neiswang@usc.edu, **zliu2898@usc.edu**

- (You are also welcome to direct message us on **Slack** via your USC Slack account)

For general discussion, or broader questions relevant to the class, there is a Slack channel that can be used:
spring25-csci-699-30165

- You should be already added here — please let me know if you aren't!



Assignments and Grading — Grading Breakdown

Assignments and Grading — Grading Breakdown

Assignment	% of Grade
1. Paper Presentation	20%
2. In-class Participation and Discussion	
2a. Role 1 — Discussion Lead 1	8%
2a. Role 2 — Discussion Lead 2	8%
2a. Role 3 — Scribe	9%
3. Course Project	
3a. Project Pitch	8%
3b. Midway Report	10%
3c. Final Presentation	12%
3d. Final Report	25%

Paper Presentations

Paper Presentations — Goal

- During the semester, each student gives **one 20-minute presentation** on a paper relevant to probabilistic and generative models.
- Ideally from a modern machine learning conference (e.g., NeurIPS, ICML, ICLR, AAAI, etc)
  - Could be a “classic” (older) paper relevant to modern models, as well.
- This will accomplish a few things:
  - Gives the class a chance to see a broad set of interesting probabilistic/generative modeling papers.
  - Gives each student (more) experience with distilling key content from a paper & presenting it.

Paper Presentations — Which papers to present?

There are a few options:

Paper Presentations — Which papers to present?

There are a few options:

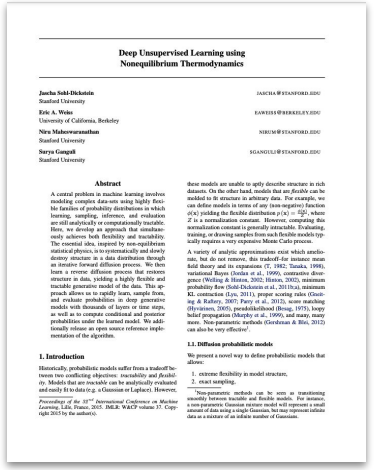
- You are encouraged to choose a relevant paper to present, ideally which ties into your own research interests.
  - We will have a spreadsheet where you will list the paper you have chosen, so instructors can confirm, and also so that we don't have overlapping papers.

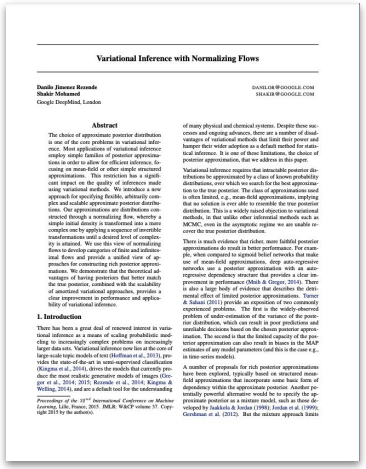
Paper Presentations — Which papers to present?

There are a few options:

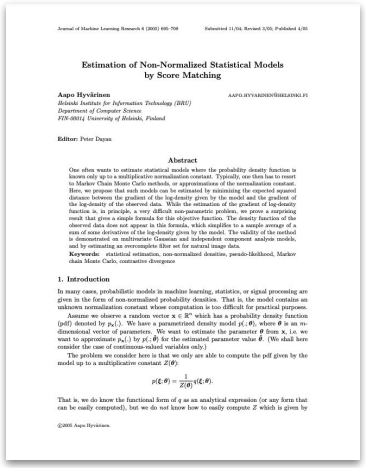
- You are encouraged to choose a relevant paper to present, ideally which ties into your own research interests.
  - We will have a spreadsheet where you will list the paper you have chosen, so instructors can confirm, and also so that we don't have overlapping papers.

- I will share a list of interesting probabilistic/generative modeling papers for inspiration— you are welcome to present on one of these.


Deep Unsupervised Learning using Nonequilibrium Thermodynamics


Variational Inference with Normalizing Flows


Auto-Encoding Variational Bayes


Estimation of Non-Normalized Statistical Models by Score Matching

Paper Presentations — Content of each paper presentation?

Ideally I’d like you to include:

Paper Presentations — Content of each paper presentation?

Ideally I'd like you to include:

1. Motivation (main problem the paper is trying to solve).
2. Brief background on prior work (*i.e.*, related methods, relevant papers).
3. Method and key equations.
    - It'll be important to boil the method down to a concise explanation given ~20 minute presentation.
4. Empirical results — how well does the method work in practice?
5. Your opinion on the paper, its (potential) impact/promise, etc.

Paper Presentations — Schedule

Paper Presentations — Schedule

Starting at the **end of February** we will go through a few (~4) presentations per class (2nd half of class).

Paper Presentations — Schedule

Starting at the **end of February** we will go through a few (~4) presentations per class (2nd half of class).

In the spreadsheet, which I will share, students will sign up for a time slot during the semester.

To make it fair, students who sign up for presentations on the first two dates will get slightly more-lenient grading (a point of extra credit on this assignment).

A	B	C	D	E	F	G	H
Date	Presentation ID	Presenter Name **(sign up!)**	Paper Title	Link/url to paper	Discussion Lead 1 **(sign up!)**	Discussion Lead 2 **(sign up!)**	Scribe **(sign up!)**
February 28							
	1						
	2						
	3						
	4						
March 7							
	5						

In-Class Participation and Discussion

There is a grade for in-class participation/discussion (during student presentations).
Students will sign up for one of three roles (again in the spreadsheet).
Each student has to do each role once during semester.

In-Class Participation and Discussion

There is a grade for in-class participation/discussion (during student presentations).

Students will sign up for one of three roles (again in the spreadsheet).

Each student has to do each role once during semester.

Roles 1 and 2 — Discussion Leads

- Read the paper before the class; Responsible for formulating a short list (~5 questions) for discussion; bring up a couple of these during class; submit all after class.

Role 3 — Scribe

- Read the paper before the class; take notes on paper during presentation; write up a ~1 page summary of the presentation.

Course Project

Course Project — Overview & Goals

Another important part of this class will be a semester-long course project.

Aims to give hands-on experience in probabilistic & generative models.

I'd like you to choose a topic that is relevant to your personal research interests (or a topic you've wanted the chance to learn more about).

- So that you can use the methods from this class in your research.

Course Project — Group Project

This will be a **group project** — groups of 3-4 students.

- Aiming for ~10 groups total (due to timing constraints)
- We will help facilitate this during class.
- *E.g.*, second half of next class everyone will introduce themselves and describe research interests, which we will write/share, to help in matching.
- Need to aim to form teams and select project idea by roughly end of this month.

Course Project — Guidance & Expectations

What does this project entail?

Course Project — Guidance & Expectations

What does this project entail?

I want people to use a probabilistic or generative model in some way!

- Application of prob/gen models from this class on a novel task or dataset.
- Algorithmic improvements in learning, inference, or evaluation of prob/gen models.
- Theoretical analysis of any aspect of existing prob/gen models.

Course Project — Guidance & Expectations

What does this project entail?

I want people to use a probabilistic or generative model in some way!

- Application of prob/gen models from this class on a novel task or dataset.
- Algorithmic improvements in learning, inference, or evaluation of prob/gen models.
- Theoretical analysis of any aspect of existing prob/gen models.

Goal is to complete a small-scale implementation or pilot study during the class.

I'm more focused on interesting conceptual ideas, rather than on performance/results.

Aim to connect it to the research you are focusing on outside of this class!

Course Project — Guidance & Expectations

The project **does not** need to be at the level of submission to a major machine learning conference.

(But it's great if works out to be the initial seed of a paper that gets submitted – you can let me know if you are interested in help on this).

Course Project — Guidance & Expectations

The project **does not** need to be at the level of submission to a major machine learning conference.

(But it's great if works out to be the initial seed of a paper that gets submitted – you can let me know if you are interested in help on this).

Possible topics:

PGMs, MCMC/VI, deep generative models (VAE, diffusion, flow matching), autoregressive models/LLMs, deep uncertainty quantification, probabilistic predictive model, decision making (BayesOpt).

Possible application areas:

NLP/language, computer vision/image/video, tabular data, discrete data, scientific data (biology, chemistry, physics), connect to other areas of CS? Other fields? Art?

Course Project — Guidance & Expectations

This is a chance to be creative with projects!

Source: sketchode project

Try something creative, perhaps try to model a new data type.

Take a risk, see if you can make something work.

If you put in effort/give an honest shot, we will not grade harshly
if the project doesn't work out.

Goal is to just to complete a small-scale implementation or pilot student during the class.

Course Project — Assignments

The project will be worth a substantial portion of the grade, and consist of four main assignments:

Course Project — Assignments

The project will be worth a substantial portion of the grade, and consist of four main assignments:

- **Project pitch:** Each group will come up with a project idea, make a few slides, and share their idea with the class for feedback (**10 minutes long**, on **Feb 7 & Feb 14**)

Course Project — Assignments

The project will be worth a substantial portion of the grade, and consist of four main assignments:

- **Project pitch:** Each group will come up with a project idea, make a few slides, and share their idea with the class for feedback (**10 minutes long**, on **Feb 7 & Feb 14**)

- **Midway report:** Each group will write a short (4 page) midway report for their project, focusing on a literature review, implementation plan, and any initial experiments.
  - Latex template will be provided.

Course Project — Assignments

The project will be worth a substantial portion of the grade, and consist of four main assignments:

- **Project pitch:** Each group will come up with a project idea, make a few slides, and share their idea with the class for feedback (**10 minutes long**, on **Feb 7 & Feb 14**)
- **Midway report:** Each group will write a short (4 page) midway report for their project, focusing on a literature review, implementation plan, and any initial experiments.
  - Latex template will be provided.
- **Final presentation:** Each group will give a presentation to the class on their final project (**30 minutes long**, on **Apr 25 & May 2**)

Course Project — Assignments

The project will be worth a substantial portion of the grade, and consist of four main assignments:

- **Project pitch:** Each group will come up with a project idea, make a few slides, and share their idea with the class for feedback (**10 minutes long**, on **Feb 7 & Feb 14**)
- **Midway report:** Each group will write a short (4 page) midway report for their project, focusing on a literature review, implementation plan, and any initial experiments.
  - Latex template will be provided.
- **Final presentation:** Each group will give a presentation to the class on their final project (**30 minutes long**, on **Apr 25 & May 2**)
- **Final report:** Each group will submit a final report for their project, describing all details, background, prior work, and results (**8-10 pages long**, **due May 9**).
  - Latex template will be provided.

Course Project — Good Course Project Examples

From some previous PGM and DGM courses:

Stanford CS236 Deep Generative Models, course project topics and titles:

- link, more recent

CMU 10-708 Probabilistic Graphical Models, course projects and reports:

- link

Course Project — Compute Resources

Depending on the type of model you choose to focus on, external compute resources (in particular, GPU resources), may be useful

- Keeping in mind we are aiming for a small-scale/pilot implementation for these projects!

Course Project — Compute Resources

Depending on the type of model you choose to focus on, external compute resources (in particular, GPU resources), may be useful

Keeping in mind we are aiming for a small-scale/pilot implementation for these projects!

CARC Allocation:

- I've requested a CARC allocation for this project, will give access to some GPU resources for training models.

Google Colab

- Free resources for students to run machine learning code.
- Implement training in notebook. Examples (e.g., for diffusion models):
  - link (HF diffusers), link (diffusion from scratch), link (simple diffusion)

LightningAI Studio Student Account

- Some GPU resources for students, link

Assignments and Grading — To Summarize…

Assignment	**% of Grade**
1. Paper Presentation *(Individual)*	20%
2. In-class Participation and Discussion *(Individual)*	
2a. Role 1 — Discussion Lead 1	8%
2a. Role 2 — Discussion Lead 2	8%
2a. Role 3 — Scribe	9%
3. Course Project *(Group)*	
3a. Project Pitch	8%
3b. Midway Report	10%
3c. Final Presentation	12%
3d. Final Report	25%

Academic Integrity

This course will follow the expectations for academic integrity as stated in the USC Student Handbook.

This is primarily a paper presentation & (group) project class, so less risk of any academic integrity violations.

Please **ask the instructor/TA** if you are unsure about what constitutes unauthorized assistance, or what information requires citation and/or attribution!

Use of Generative AI

… *e.g.*, in your paper presentations, group project (midway report, final report, etc).

Use of Generative AI

… *e.g.*, in your paper presentations, group project (midway report, final report, etc).

Short answer, **yes** you can use it — this is a generative models class, after all.

And learning to use AI in an effective way is an important emerging skill.

Use of Generative AI

… *e.g.*, in your paper presentations, group project (midway report, final report, etc).

Short answer, **yes** you can use it — this is a generative models class, after all.

And learning to use AI in an effective way is an important emerging skill.

However, keep in mind the following caveats:

- AI tools are permitted to help you brainstorm topics or revise work you have already written.
- If you provide minimum-effort prompts, you will get low-quality results. You will need to refine your prompts to get good outcomes. This will take work.
- Proceed with caution when using AI, don't assume the information provided is accurate.
- You need to **acknowledge that you used AI**. Please include sentences at the end of any assignment that used AI, explaining how and why you used AI.

Optional Readings — and Supplementary Materials

The following resources are useful for many of the topics covered in this class:

1. Kevin Murphy, “Machine Learning: A Probabilistic Perspective”, 2012 (link).
2. Kevin Murphy, “Probabilistic Machine Learning: Advanced Topics”, 2023 (link).
3. Chris Bishop, “Pattern Recognition and Machine Learning”, 2006 (link).
4. Chris Bishop, “Deep Learning - Foundations and Concepts”, 2024 (link).
5. Stefano Ermon, Deep Generative Models, Course Notes (link).

A Tour of the Class

A Tour of the Class

Want to roughly go through the topics we'll learn in this class.

And the schedule* over the coming semester.

*subject to change or adjustment if needed 😎.

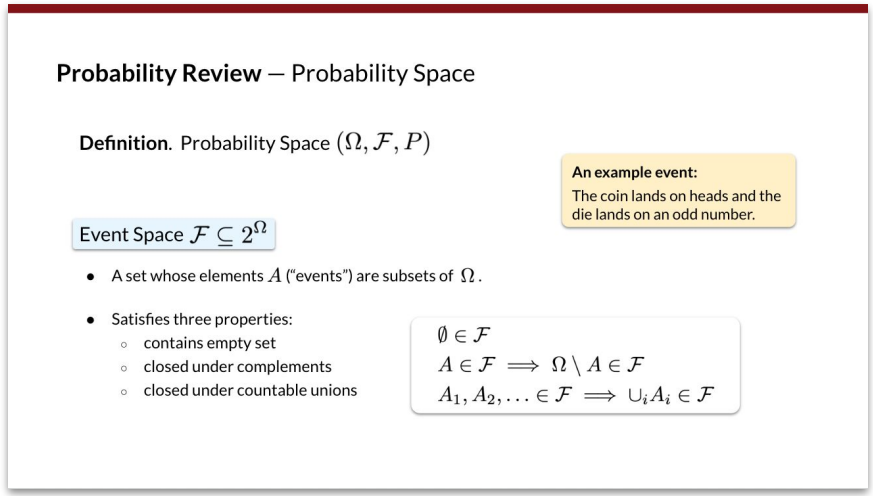
Lecture 1 (Today, January 17) — Class Overview and Probability Review

Lecture 1 (Today, January 17) — Class Overview and Probability Review

We are here 

currently going through an overview of the class.

In a few minutes we'll do a probability review of terminology/notation useful for various models & methods we'll learn in this class.

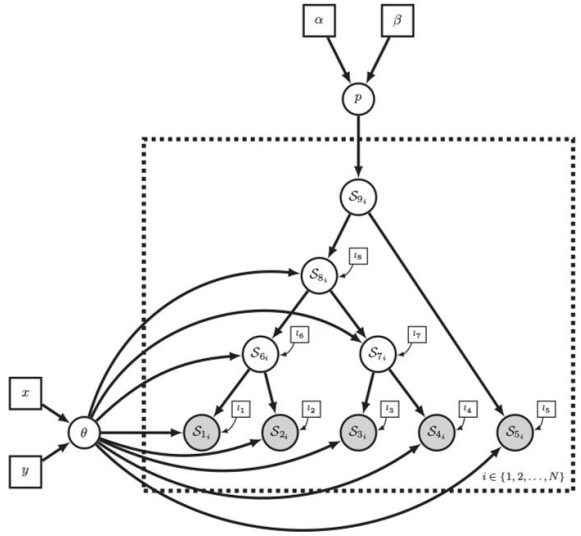


Lecture 2 (January 24) — Probabilistic Graphical Models (PGMs)

Lecture 2 (January 24) — Probabilistic Graphical Models (PGMs)

Lecture: PGMs — using ideas from both probability and graph theory to derive efficient machine learning algorithms

- Directed vs undirected graphical models. Bayes networks vs Markov random fields.
- Plate notation, generative process notation.
- Inference vs learning in graphical models.
- Famous/classic PGMs (HMMs, GMMs, LDA, VAE, etc.)
- Some probabilistic programming.



Source: "Probabilistic Graphical Model Representation in Phylogenetics", Höhna 2014

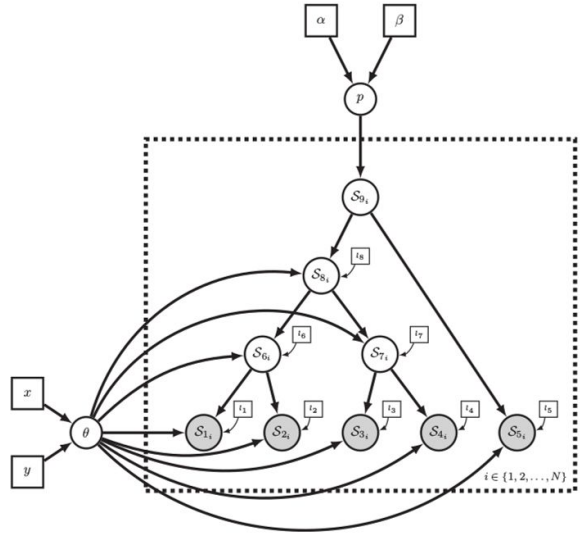
Lecture 2 (January 24) — Probabilistic Graphical Models (PGMs)

Lecture: PGMs — using ideas from both probability and graph theory to derive efficient machine learning algorithms

- Directed vs undirected graphical models. Bayes networks vs Markov random fields.
- Plate notation, generative process notation.
- Inference vs learning in graphical models.
- Famous/classic PGMs (HMMs, GMMs, LDA, VAE, etc.)
- Some probabilistic programming.

After:

- Class introductions and research interests, set up for group project.
- Compute resources.



Source: "Probabilistic Graphical Model Representation in Phylogenetics", Höhna 2014

Lecture 3 (January 31) — PGMs: Classic Algorithms

Lecture 3 (January 31) — PGMs: Classic Algorithms

Lecture: Classic algorithms in probabilistic graphical models (PGMs) for exact and approximate inference & learning.

- Forward-backward algorithm in HMMs.
- Viterbi algorithm in HMMs.
- Exact inference and variable elimination algorithm.
- Belief Propagation (message passing) algorithm.
- Expectation-maximization (EM) algorithm.

Source: USC Viterbi Magazine, “The Viterbi Algorithm at 50”

Lecture 3 (January 31) — PGMs: Classic Algorithms

Lecture: Classic algorithms in probabilistic graphical models (PGMs) for exact and approximate inference & learning.

- Forward-backward algorithm in HMMs.
- Viterbi algorithm in HMMs.
- Exact inference and variable elimination algorithm.
- Belief Propagation (message passing) algorithm.
- Expectation-maximization (EM) algorithm.

Source: USC Viterbi Magazine, "The Viterbi Algorithm at 50"

After:

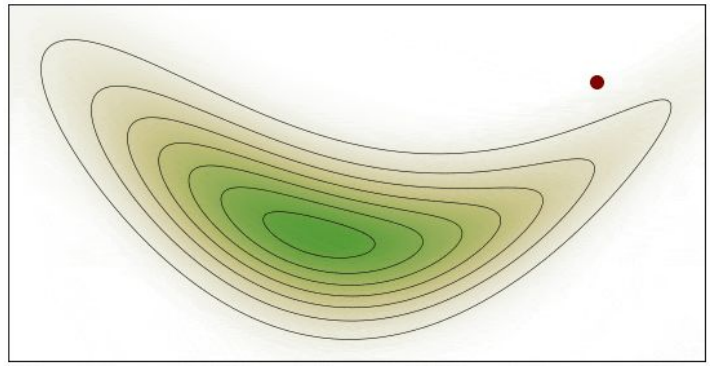
- Lab session on probabilistic programming (Pyro, Stan, NumPyro, GPyTorch, etc).
- Check-in on group formation and project ideation.

Lecture 4 (February 7) — Approximate Bayesian Inference

Lecture 4 (February 7) — Approximate Bayesian Inference

Lecture: Approximate inference algorithms in probabilistic models, including the classics — Markov chain Monte Carlo (MCMC), and variational inference (VI).

- Markov chain Monte Carlo (MCMC) methods, including:
  - Metropolis–Hastings algorithm, Gibbs sampling, slice sampling.
  - Gradient-based methods: Langevin Monte Carlo, Hamiltonian Monte Carlo.
- Variational inference (VI) methods, including
  - Evidence lower bound (ELBO), Stochastic VI, black-box VI.

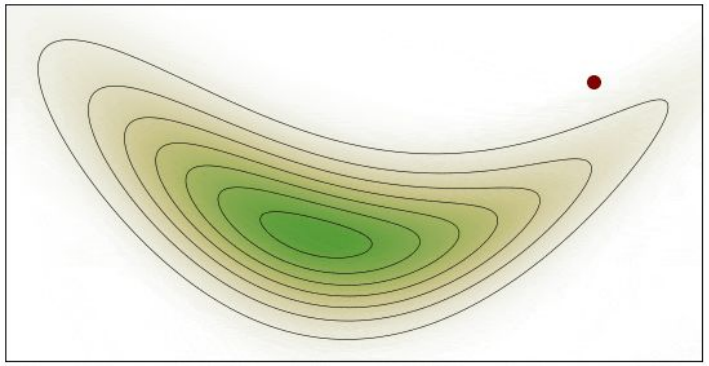


Source: “Creating animations with MCMC”, Krepl 2018, Wikimedia commons,

Lecture 4 (February 7) — Approximate Bayesian Inference

Lecture: Approximate inference algorithms in probabilistic models, including the classics — Markov chain Monte Carlo (MCMC), and variational inference (VI).

- Markov chain Monte Carlo (MCMC) methods, including:
  - Metropolis–Hastings algorithm, Gibbs sampling, slice sampling.
  - Gradient-based methods: Langevin Monte Carlo, Hamiltonian Monte Carlo.
- Variational inference (VI) methods, including
  - Evidence lower bound (ELBO), Stochastic VI, black-box VI.



Source: "Creating animations with MCMC", Krepl 2018, Wikimedia commons,

After:

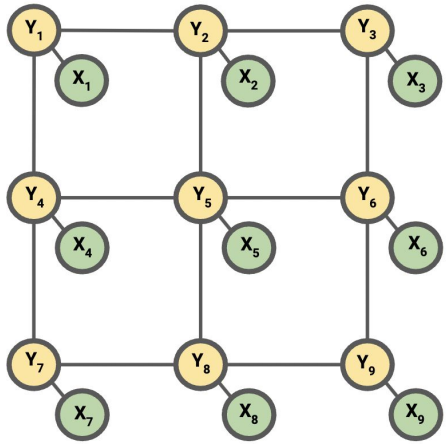
- Project Pitches #1: Groups 1-5

Lecture 5 (February 14) — Intro to Deep Generative Models

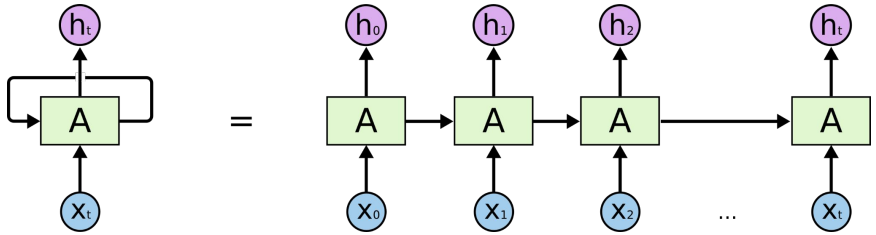
Lecture 5 (February 14) — Intro to Deep Generative Models

Lecture: Introduction to deep (probabilistic) generative models, model paradigms, learning in these models, optimal representations and objective functions.

- Modeling paradigms: autoregressive vs diffusion vs flow models.
- Maximum-likelihood / Kullback-Leibler divergence-based Learning.
- Energy-based models
- Early autoregressive models.



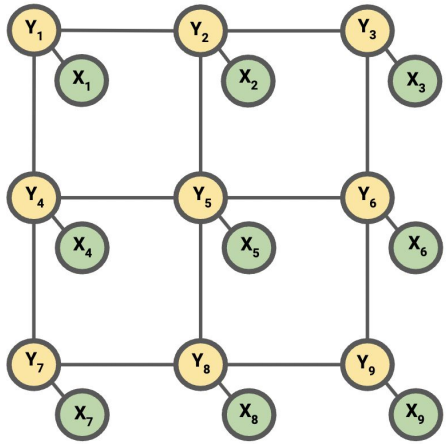
X_i: noisy pixels
Y_i: "true" pixels



Lecture 5 (February 14) — Intro to Deep Generative Models

Lecture: Introduction to deep (probabilistic) generative models, model paradigms, learning in these models, optimal representations and objective functions.

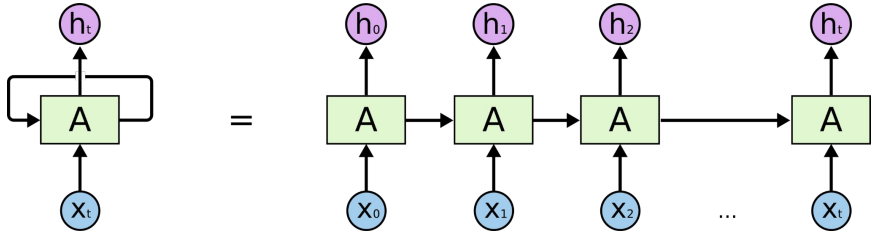
- Modeling paradigms: autoregressive vs diffusion vs flow models.
- Maximum-likelihood / Kullback-Leibler divergence-based Learning.
- Energy-based models
- Early autoregressive models.



X_i: noisy pixels
Y_i: "true" pixels

After:

- Project Pitches #2: Groups 6-10

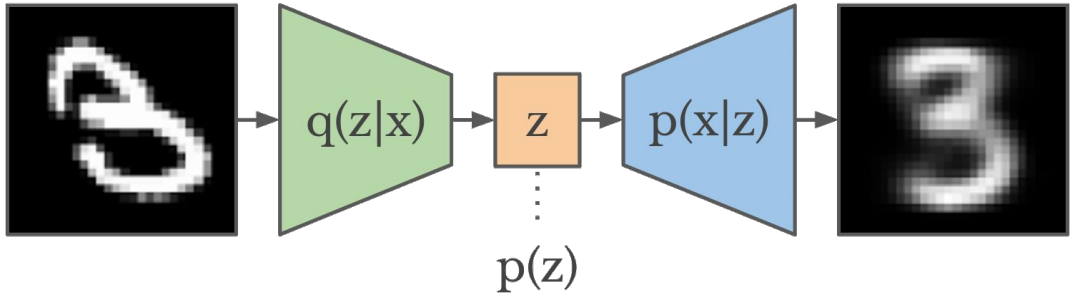


Lecture 6 (February 21) — VAEs and GANs

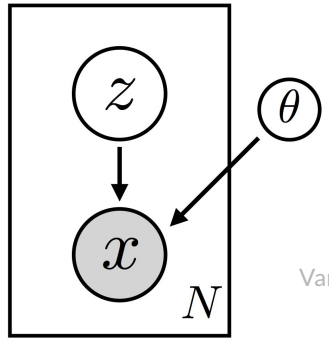
Lecture 6 (February 21) — VAEs and GANs

Lecture: Two historically iconic deep generative models — variational autoencoders (VAEs) and generative adversarial networks (GANs).

- Variational autoencoders
  - Relation to other graphical models.
  - Relation to (classic) autoencoders.
- Generative adversarial networks
  - Extensions in recent times.



"Building Variational Auto-Encoders in TensorFlow", Danijar Hafner

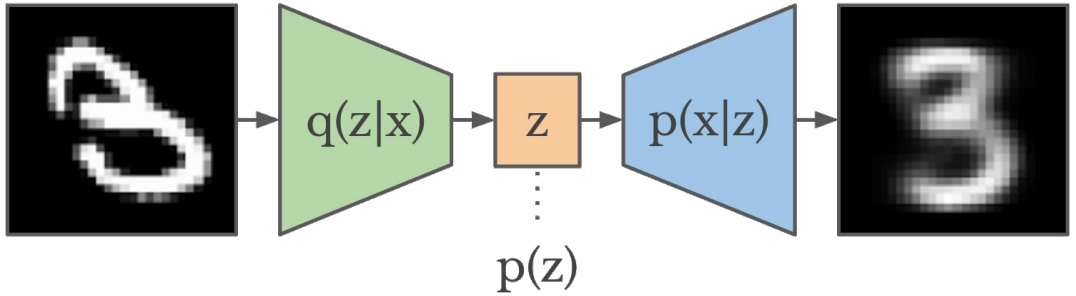


Variational Autoencoder PGM, Pyro documentation

Lecture 6 (February 21) — VAEs and GANs

Lecture: Two historically iconic deep generative models — variational autoencoders (VAEs) and generative adversarial networks (GANs).

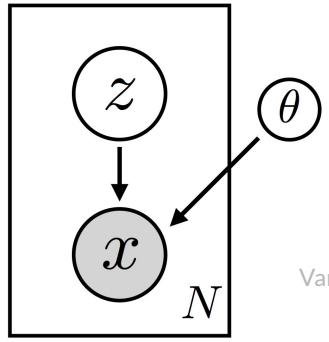
- Variational autoencoders
  - Relation to other graphical models.
  - Relation to (classic) autoencoders.
- Generative adversarial networks
  - Extensions in recent times.



"Building Variational Auto-Encoders in TensorFlow", Danijar Hafner

After:

- Project feedback/discussion via short in-class meetings with each group.



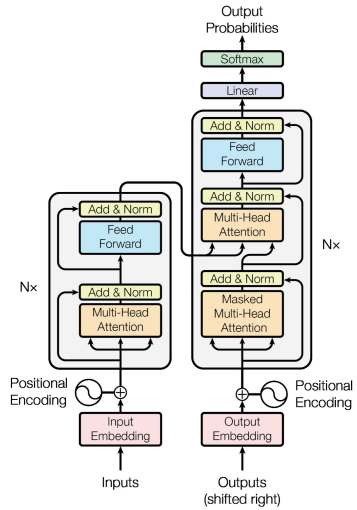
Variational Autoencoder PGM, Pyro documentation

Lecture 7 (February 28) — Autoregressive Models

Lecture 7 (February 28) — Autoregressive Models

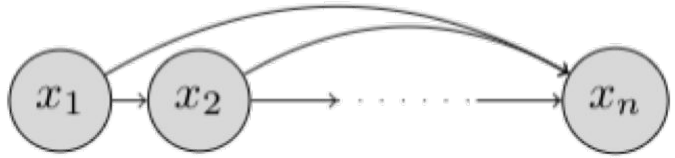
Lecture: Autoregressive generative models, the transformer neural architecture, and its use in large language modeling.

- Neural autoregressive density estimation
- Recurrent neural networks
- Attention-based models
- Transformer models
- Modern autoregressive transformer LLMs



"Attention Is All You Need", Vaswani et al., 2017

Once upon a … [EOS]

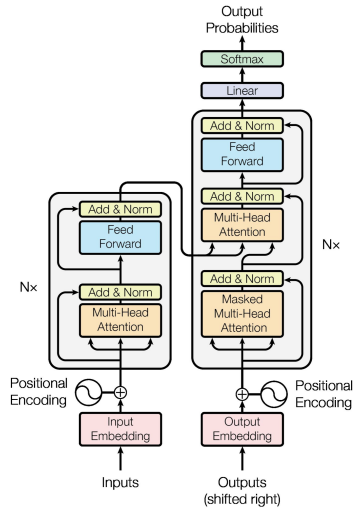


Source: Stefano Ermon, Deep Graphical Models

Lecture 7 (February 28) — Autoregressive Models

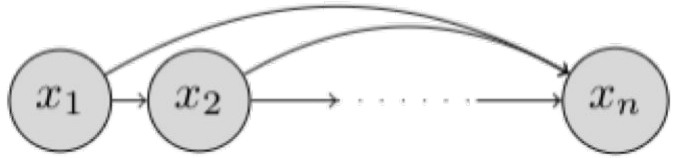
Lecture: Autoregressive generative models, the transformer neural architecture, and its use in large language modeling.

- Neural autoregressive density estimation
- Recurrent neural networks
- Attention-based models
- Transformer models
- Modern autoregressive transformer LLMs



"Attention Is All You Need", Vaswani et al., 2017

Once upon a … [EOS]



Source: Stefano Ermon, Deep Graphical Models

After:

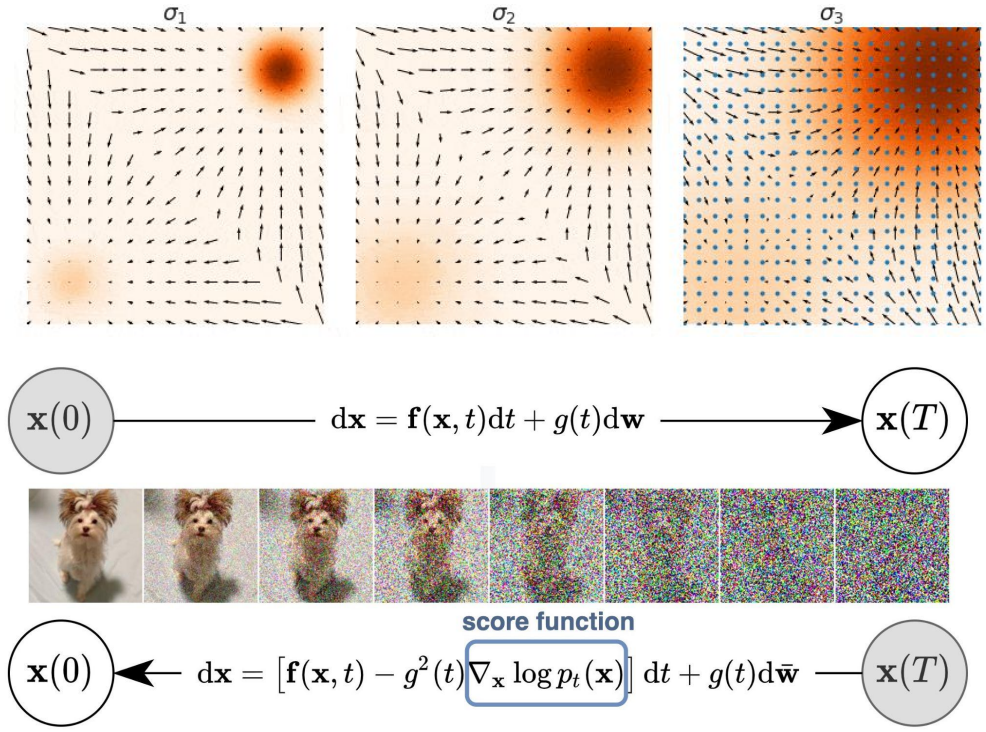
- Paper presentations from students.

Lecture 8 (March 7) — Diffusion Models

Lecture 8 (March 7) — Diffusion Models

Lecture: Score-based generative models and diffusion models.

- The (Stein) score function, score matching.
- Score-based generative models.
- Sampling with Langevin Monte Carlo.
- Denoising diffusion probabilistic models.
- Learning with variational inference.



Source: Yang Song, "Generative Modeling by Estimating Gradients of the Data Distribution"

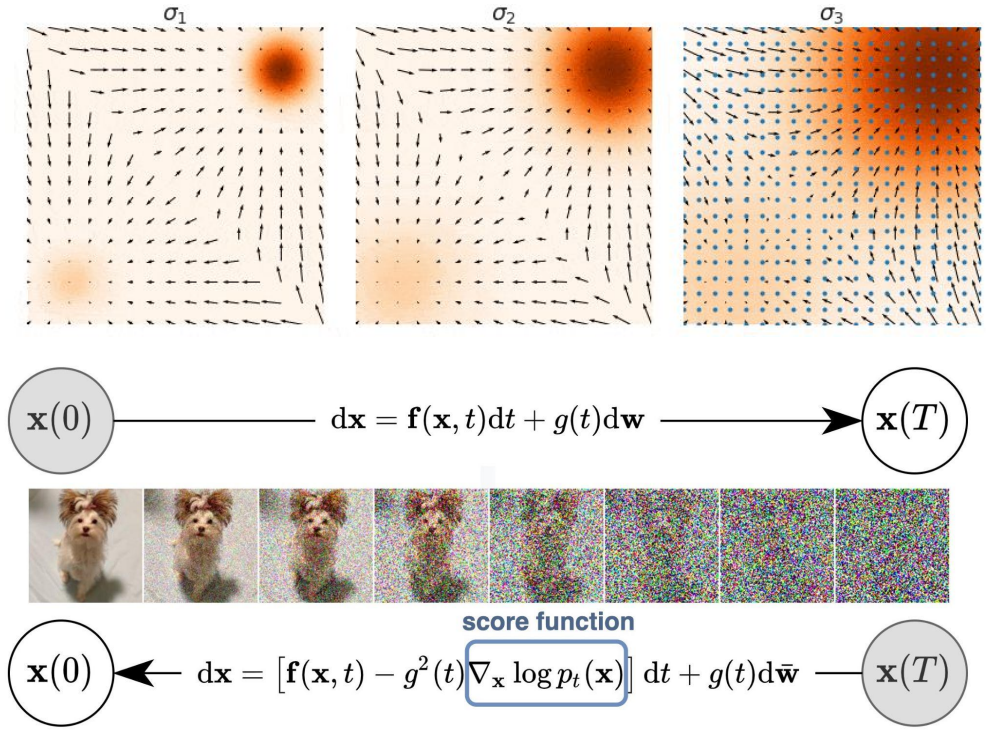
Lecture 8 (March 7) — Diffusion Models

Lecture: Score-based generative models and diffusion models.

- The (Stein) score function, score matching.
- Score-based generative models.
- Sampling with Langevin Monte Carlo.
- Denoising diffusion probabilistic models.
- Learning with variational inference.

After:

- Paper presentations from students.



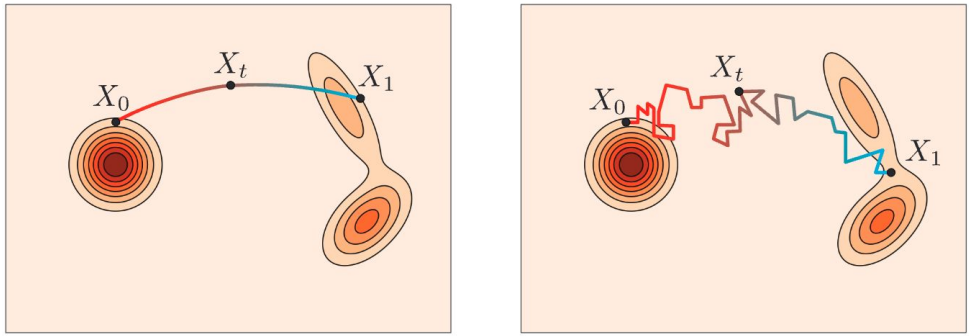
Source: Yang Song, "Generative Modeling by Estimating Gradients of the Data Distribution"

Lecture 9 (March 14) — Flow-based Models

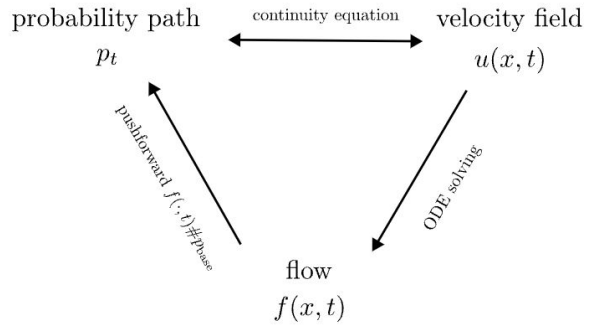
Lecture 9 (March 14) — Flow-based Models

Lecture: Normalizing flows, continuous normalizing flows, neural ODEs, flow matching, and more.

- Normalizing flows
- Continuous normalizing flows
- Neural ordinary differential equations
- Conditional flow matching, and flow matching
- Extensions: discrete flow matching, generator matching.



(a) Flow **(b)** Diffusion



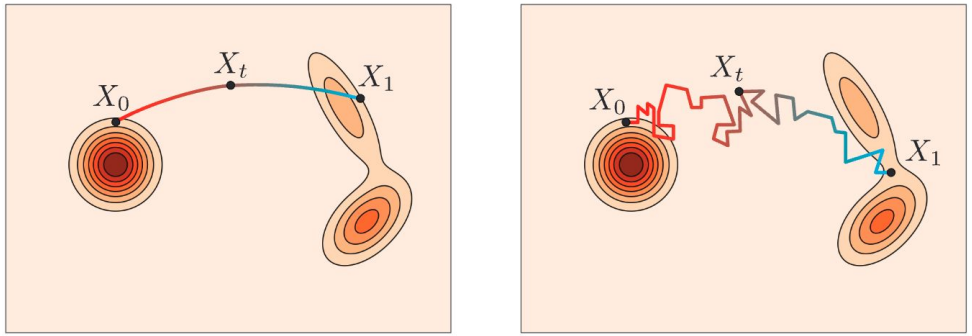
"Flow Matching Guide and Code", Lipman et al., 2024

"A Visual Dive into Conditional Flow Matching", 2024

Lecture 9 (March 14) — Flow-based Models

Lecture: Normalizing flows, continuous normalizing flows, neural ODEs, flow matching, and more.

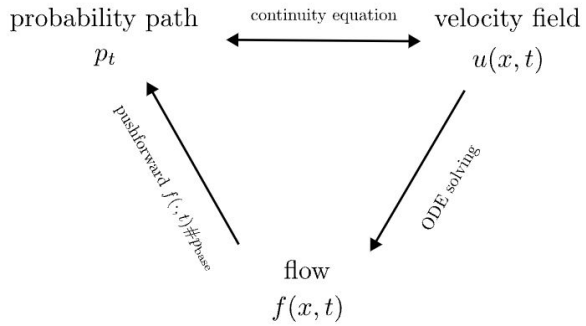
- Normalizing flows
- Continuous normalizing flows
- Neural ordinary differential equations
- Conditional flow matching, and flow matching
- Extensions: discrete flow matching, generator matching.



(a) Flow **(b)** Diffusion

After:

- Paper presentations from students.



"Flow Matching Guide and Code", Lipman et al., 2024

"A Visual Dive into Conditional Flow Matching", 2024

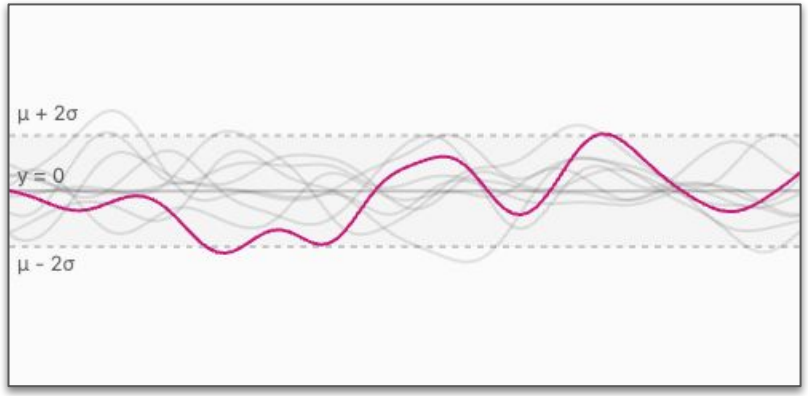
Note: March 21st is Spring Break 

Lecture 10 (March 28) — Predictive Uncertainty Quantification

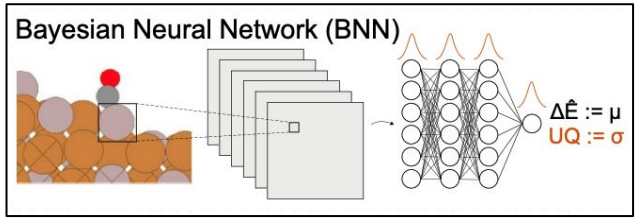
Lecture 10 (March 28) — Predictive Uncertainty Quantification

Lecture: Predictive UQ models, including Gaussian processes (GPs), Bayesian neural networks (BNNs), ensembles/dropout, UQ in LLMs, and relevant metrics.

- Classic predictive UQ: Bayesian parametric models, Gaussian processes (GPs).
- Neural/deep predictive UQ: Bayesian neural networks (BNN), ensembles, MC-dropout.
- Uncertainty quantification in LLMs.
- Relevant metrics: calibration, proper scoring rules.



A Visual Exploration of Gaussian Processes, distill.pub

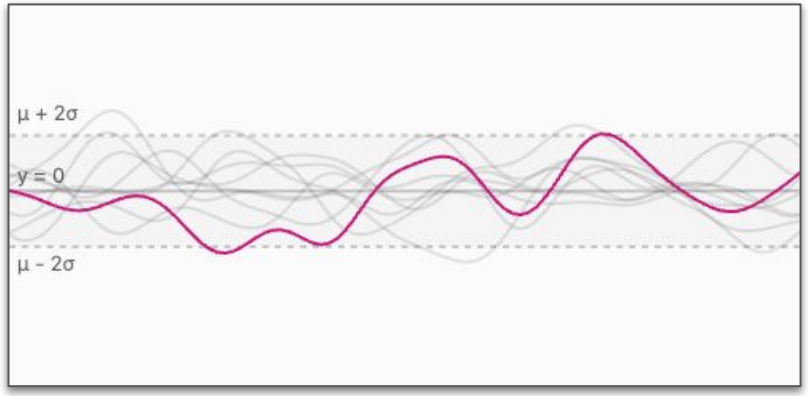


Tran, Neiswanger, et al., MLST. 2020

Lecture 10 (March 28) — Predictive Uncertainty Quantification

Lecture: Predictive UQ models, including Gaussian processes (GPs), Bayesian neural networks (BNNs), ensembles/dropout, UQ in LLMs, and relevant metrics.

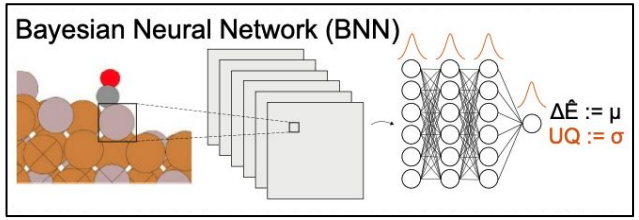
- Classic predictive UQ: Bayesian parametric models, Gaussian processes (GPs).
- Neural/deep predictive UQ: Bayesian neural networks (BNN), ensembles, MC-dropout.
- Uncertainty quantification in LLMs.
- Relevant metrics: calibration, proper scoring rules.



A Visual Exploration of Gaussian Processes, distill.pub

After:

- Paper presentations from students.



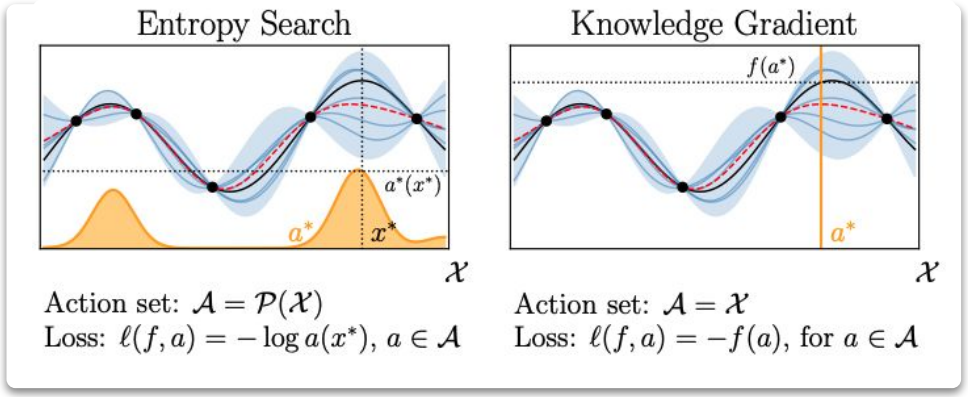
Tran, Neiswanger, et al., MLST. 2020

Lecture 11 (April 4) — Active Learning and Optimization

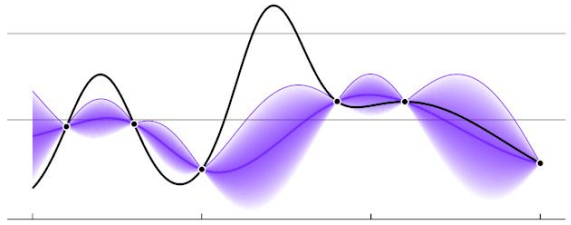
Lecture 11 (April 4) — Active Learning and Optimization

Lecture: Using predictive UQ models for active learning and sequential decision making, including Bayesian optimization and optimal experimental design.

- Decision making under uncertainty.
- Active learning.
- Bayesian optimization.
- Optimal experimental design.
- Extensions: Bayesian algorithm execution.



"Generalizing Bayesian Optimization with Decision-theoretic Entropies", 2022

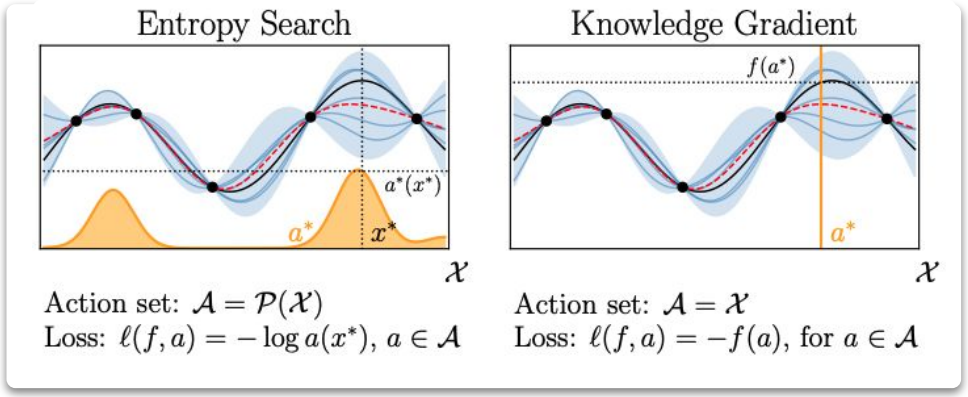


Bayesian Optimization, Wikipedia

Lecture 11 (April 4) — Active Learning and Optimization

Lecture: Using predictive UQ models for active learning and sequential decision making, including Bayesian optimization and optimal experimental design.

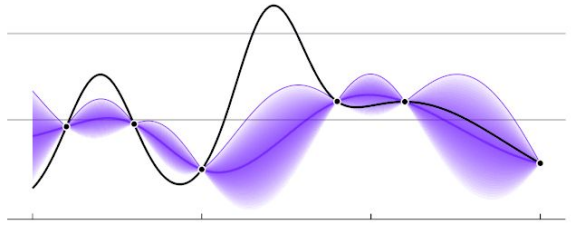
- Decision making under uncertainty.
- Active learning.
- Bayesian optimization.
- Optimal experimental design.
- Extensions: Bayesian algorithm execution.



"Generalizing Bayesian Optimization with Decision-theoretic Entropies", 2022

After:

- Paper presentations from students.



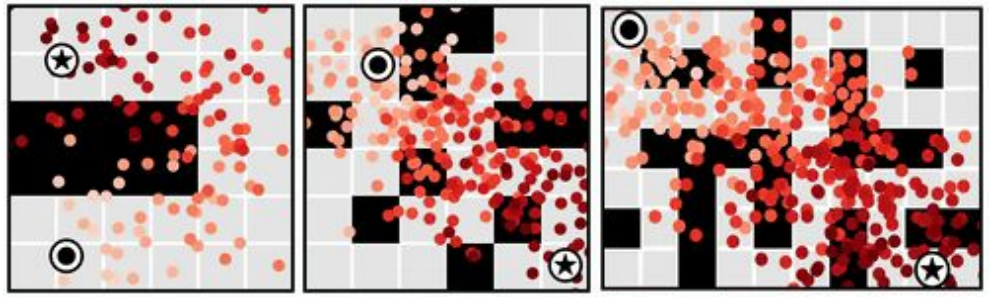
Bayesian Optimization, Wikipedia

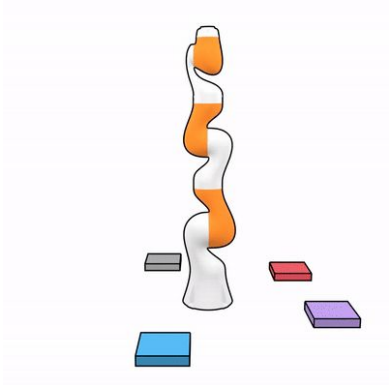
Lecture 12 (April 11) — Generative Models in Decision Making

Lecture 12 (April 11) — Generative Models in Decision Making

Lecture: Using generative models (diffusion, VAE, LLMs, etc), in decision making and optimization procedures.

- High-dimensional BayesOpt with VAE.
- Diffusion planning.
- Vision-language-action models.
- Decision transformer.
- Decision making with LLMs.



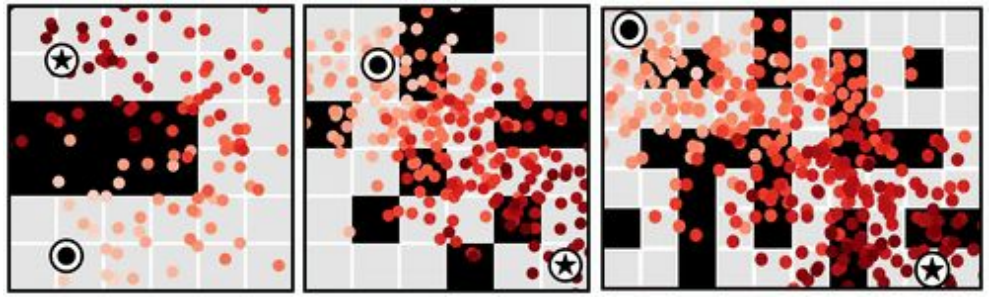


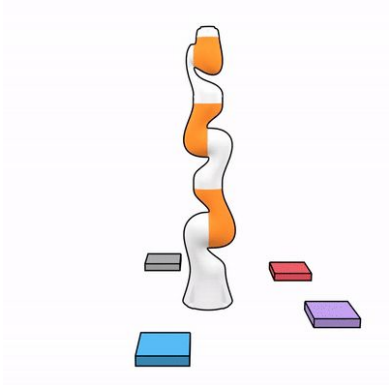
Source: Planning with Diffusion for Flexible Behavior Synthesis.

Lecture 12 (April 11) — Generative Models in Decision Making

Lecture: Using generative models (diffusion, VAE, LLMs, etc), in decision making and optimization procedures.

- High-dimensional BayesOpt with VAE.
- Diffusion planning.
- Vision-language-action models.
- Decision transformer.
- Decision making with LLMs.





Source: Planning with Diffusion for Flexible Behavior Synthesis.

After:

- Paper presentations from students.

Lecture 13 (April 18) — Mystery Class

Lecture: Special topic, to be announced :-).

Lecture 13 (April 18) — Mystery Class

Lecture: Special topic, to be announced :-).

After:

- Finish all student paper presentations.

Lecture 14 (April 25) — Final Presentations #1

Full day of final presentations of class projects! (Teams 1-5)

- Each team will give a presentation.
- ~30 minutes long.
- Describing their full project and results.

Lecture 14 (May 2) — Final Presentations #2

Full day of final presentations of class projects! (Teams 6-10)

- Each team will give a presentation.
- ~30 minutes long.
- Describing their full project and results.

Probability Review

Probability Review

Goal: give a brief background/review of probability that will be useful in this class.

Probability Review

Goal: give a brief background/review of probability that will be useful in this class.

Terminology & notation:

- From elementary probability theory (with a few mentions of measure-theoretic probability theory).
- Which are used throughout modern machine learning work.

Probability Review

Goal: give a brief background/review of probability that will be useful in this class.

Terminology & notation:

- From elementary probability theory (with a few mentions of measure-theoretic probability theory).
- Which are used throughout modern machine learning work.

Will review:

- Probability space, conditional probability, chain rule, independence (for events).
- Random variables, random vectors.
- Cumulative distribution function (CDF), probability density/mass functions (PDF, PMF)
- Joint probability distributions, expectations, chain rule, etc.

Probability Review — Probability Space

Definition. Probability Space $(\Omega, \mathcal{F}, P)$

Probability Review — Probability Space

Definition. Probability Space $(\Omega, \mathcal{F}, P)$

Sample Space Ω

Event Space $\mathcal{F}$

Probability Function (Measure) P

Probability Review — Probability Space

Definition. Probability Space $(\Omega, \mathcal{F}, P)$

Experiment: “roll a die and flip a coin”

Probability Review — Probability Space

Definition. Probability Space $(\Omega, \mathcal{F}, P)$

Sample Space Ω

- The set of all outcomes of a random experiment

Experiment: "roll a die and flip a coin"

Ω

H-1, H-2, H-3,
H-4, H-5, H-6,
T-1, T-2, T-3,
T-4, T-5, T-6,

Probability Review — Probability Space

Definition. Probability Space $(\Omega, \mathcal{F}, P)$

An example event:

The coin lands on heads and the die lands on an odd number.

Event Space $\mathcal{F} \subseteq 2^{\Omega}$

- A set whose elements A ("events") are subsets of Ω.

Probability Review — Probability Space

Definition. Probability Space $(\Omega, \mathcal{F}, P)$

An example event:

The coin lands on heads and the die lands on an odd number.

Event Space $\mathcal{F} \subseteq 2^{\Omega}$

- A set whose elements A ("events") are subsets of Ω.
- Satisfies three properties:
  - contains empty set
  - closed under complements
  - closed under countable unions

$$\emptyset \in \mathcal{F}$$

$$A \in \mathcal{F} \implies \Omega \setminus A \in \mathcal{F}$$

$$A_1, A_2, \ldots \in \mathcal{F} \implies \cup_i A_i \in \mathcal{F}$$

Probability Review — Probability Space

Definition. Probability Space $(\Omega, \mathcal{F}, P)$

Probability Function (Measure) P

- A function $P : \mathcal{F} \to \mathbb{R}$.

Probability Review — Probability Space

Definition. Probability Space $(\Omega, \mathcal{F}, P)$

Probability Function (Measure) P

- A function $P : \mathcal{F} \to \mathbb{R}$.
- Satisfies three properties:
  - Non-negativity
  - Normalization
  - Finite-additivity

"Axioms of Probability"

$$P(A) \geq 0, \text{ for all } A \in \mathcal{F}$$
$$P(\Omega) = 1$$
$$\text{If } A_1, A_2, \ldots \text{ are disjoint} \implies P(\cup_i A_i) = \sum_i P(A_i)$$

Probability Review — Probability Space

Definition. Probability Space $(\Omega, \mathcal{F}, P)$

Measure Theoretic Brief

$\mathcal{F}$ is a σ-algebra.

$(\Omega, \mathcal{F})$ is a measurable space.

P is a probability measure.

Probability Function (Measure) P

- A function $P : \mathcal{F} \to \mathbb{R}$.
- Satisfies three properties:
  - Non-negativity
  - Normalization
  - Finite-additivity

"Axioms of Probability"

$P(A) \geq 0, \text{ for all } A \in \mathcal{F}$

$P(\Omega) = 1$

$\text{If } A_1, A_2, \ldots \text{ are disjoint } \implies P(\cup_i A_i) = \sum_i P(A_i)$

Probability Review — Probability Space

Some properties of a probability space $(\Omega, \mathcal{F}, P)$:

Probability Review — Probability Space

Some properties of a probability space $(\Omega, \mathcal{F}, P)$:

- $A \subseteq B \implies P(A) \leq P(B)$
- $P(A \cap B) \leq \min(P(A), P(B))$
- Union Bound: $P(A \cup B) \leq P(A) + P(B)$
- $P(\Omega - A) = 1 - P(A)$
- Law of Total Probability: If $A_1, \ldots, A_k$ are disjoint, and $\bigcup_{i=1}^{k} A_i = \Omega$, then $\sum_{i=1}^{k} P(A_i) = 1$

Probability Review — Conditional Probability (*Events*)

Definition. Conditional probability of any event A given event B:

(where B has non-zero probability)

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

"Probability measure of the event A, assuming the occurrence of event B"

Probability Review — Conditional Probability (*Events*)

Definition. Conditional probability of any event A given event B:

(where B has non-zero probability)

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

Example:

Probability that the die lands on an odd number, given that the die lands on >2.

"Probability measure of the event A, assuming the occurrence of event B"

Probability Review — Chain Rule (*Events*)

Definition. Let $A_1, \ldots, A_k$ be events, where $P(A_i) > 0$. Then the chain rule states that:

$$P(A_1 \cap A_2 \cap \cdots \cap A_k)$$
$$= P(A_1)P(A_2|A_1)P(A_3|A_2 \cap A_1) \cdots P(A_k|A_1 \cap A_2 \cap \cdots \cap A_{k-1})$$

⇒ derived by applying conditional probability multiple times!

Probability Review — Independence (*Events*)

Definition. Two events A and B are called independent if

$$P(A \cap B) = P(A)P(B)$$

or

$$P(A \mid B) = P(A)$$

"Observing B does not have any effect on the probability of A"

Probability Review — Random Variables

Probability Review — Random Variables

Suppose we just care about a (real-valued) function of outcomes $\omega \in \Omega$.

E.g. if Ω is outcomes of 5 coin flips, we might care about "number of heads that occur", or "length of longest run of tails" ⇒ both are real-valued functions.

3 (number of heads)

And we care about the induced distribution over a numeric/real space, given by this function.

Probability Review — Random Variables

Definition. A Random Variable is a (measurable) function $X : \Omega \to \mathbb{R}$. We can define the probability of X in the following way.

Note: "measurable function" ensures that this is an event in $\mathcal{F}$ with a well-defined probability

Probability Review — Random Variables

Definition. A Random Variable is a (measurable) function $X : \Omega \to \mathbb{R}$. We can define the probability of X in the following way.

Note: "measurable function" ensures that this is an event in $\mathcal{F}$ with a well-defined probability

For discrete random variables:

$$P(X = k) = P(\{\omega : X(\omega) = k\})$$

And for continuous random variables:

$$P(a \leq X \leq b) = P(\{\omega : a \leq X(\omega) \leq b\})$$

Probability Review — Random Variables

Definition. A Random Variable is a (measurable) function $X : \Omega \to \mathbb{R}$. We can define the probability of X in the following way.

Note: "measurable function" ensures that this is an event in $\mathcal{F}$ with a well-defined probability

$$\mathbb{P}(\{\omega : X(\omega) = k\})$$

$$\mathbb{P}(\{\omega : a \leq X(\omega) \leq b\})$$

Doob, J.
Quoted in Statistical Science

"While writing my book [Stochastic Processes] I had an argument with Feller. He asserted that everyone said 'random variable' and I asserted that everyone said 'chance variable.'

We obviously had to use the same name in our books, so we decided the issue by a stochastic procedure. That is, we tossed for it and he won."

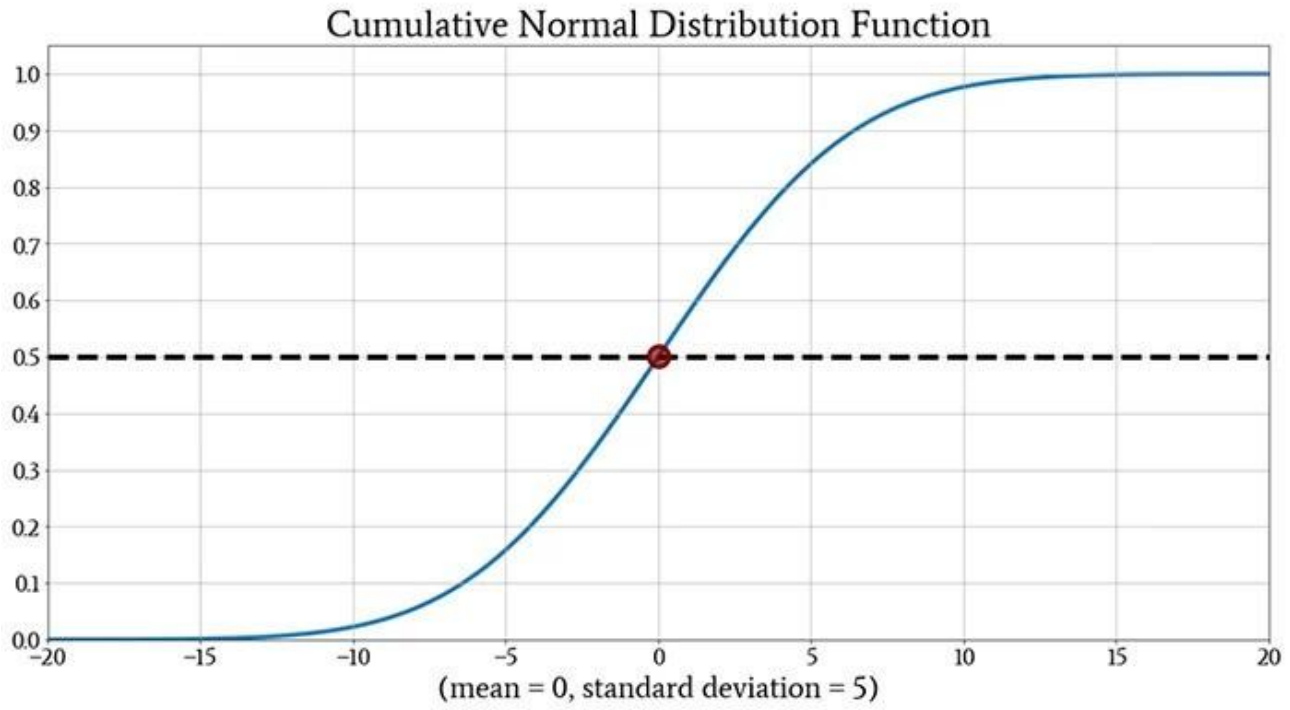
Probability Review — CDF, PMF, PDF

It can useful to define alternative functions to specify probability measures of random variables — e.g. CDF, PMF, and PDF.

Probability Review — Cumulative Distribution Function (CDF)

Probability Review — Cumulative Distribution Function (CDF)

Definition. Cumulative Distribution Function (CDF) is a function $F_X : \mathbb{R} \to [0, 1]$ which specifies a probability measure as $F_X(x) = P(X \leq x)$.



Source: Texas Instruments

Probability Review — Cumulative Distribution Function (CDF)

Definition. Cumulative Distribution Function (CDF) is a function $F_X : \mathbb{R} \to [0, 1]$ which specifies a probability measure as $F_X(x) = P(X \leq x)$.

- $0 \leq F_X(x) \leq 1$
- $\lim_{x \to -\infty} F_X(x) = 0$
- $\lim_{x \to \infty} F_X(x) = 1$
- $x \leq y \implies F_X(x) \leq F_X(y)$

Probability Review — Probability Mass Function (PMF)

Probability Review — Probability Mass Function (PMF)

Definition. When X is a discrete random variable, a Probability Mass Function (PMF) is a function $p_X : \Omega \to \mathbb{R}$ such that $p_X(x) = P(X = x)$.

Note: $\text{Val}(X)$ is the set of possible sample-space values that X can take on.

Properties:

- $0 \leq p_X(x) \leq 1$
- $\sum_{x \in \text{Val}(X)} p_X(x) = 1$
- $\sum_{x \in A} p_X(x) = P(X \in A)$

Probability Review — Probability Density Function (PDF)

Definition. When X is a continuous random variable, and the CDF is differentiable everywhere, a Probability Density Function (PDF) is $f_X : \mathbb{R} \to \mathbb{R}$ such that

$$f_X(x) = \frac{dF_X(x)}{dx}$$

For very small Δx, we have $P(x \leq X \leq x + \Delta x) \approx f_X(x)\Delta x$

Probability Review — Probability Density Function (PDF)

Definition. When X is a continuous random variable, and the CDF is differentiable everywhere, a Probability Density Function (PDF) is $f_X : \mathbb{R} \to \mathbb{R}$ such that

$$f_X(x) = \frac{dF_X(x)}{dx}$$

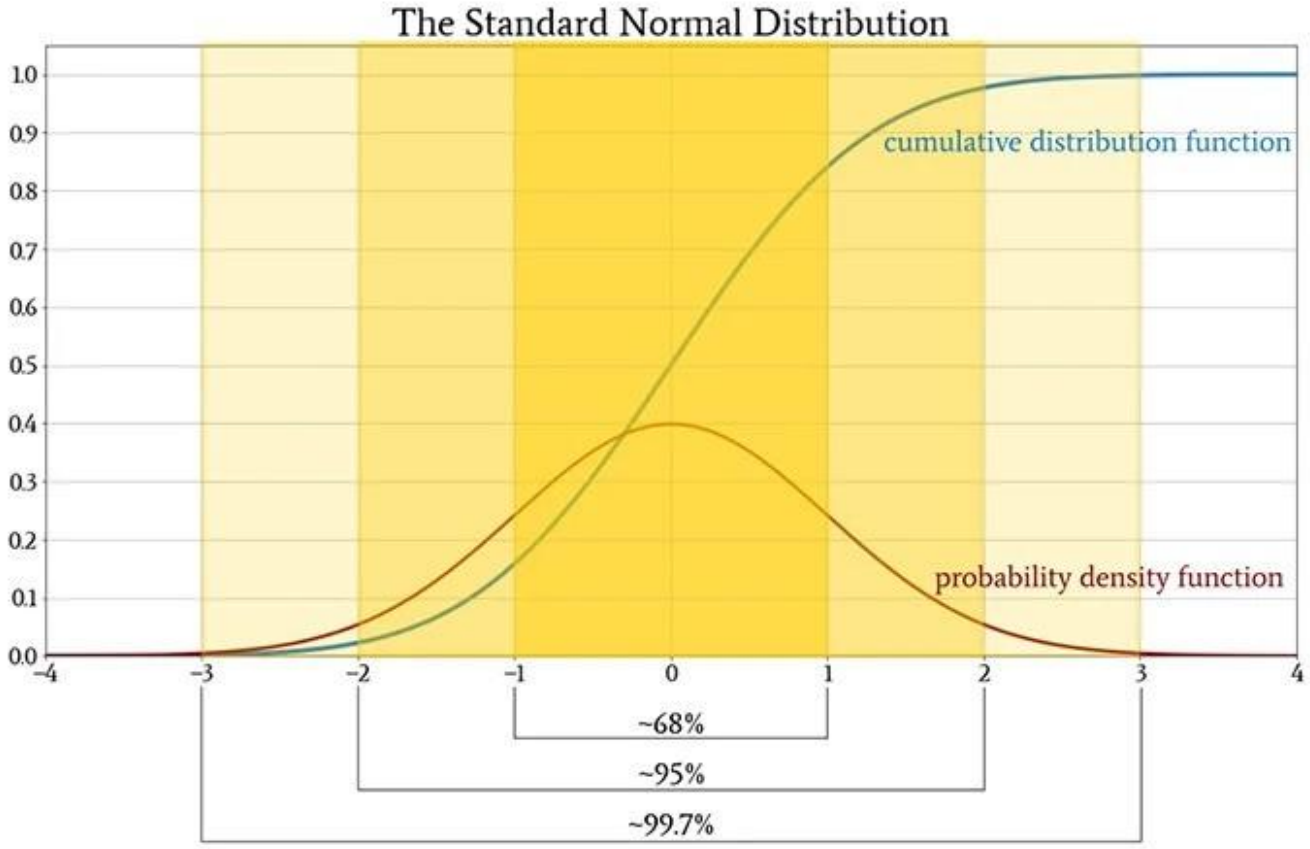
For very small Δx, we have $P(x \leq X \leq x + \Delta x) \approx f_X(x)\Delta x$

Properties:

- $f_X(x) \geq 0$
- $\int_{-\infty}^{\infty} f_X(x)dx = 1$
- $\int_{x \in A} f_X(x)dx = P(X \in A)$

Probability Review — Probability Density Function (PDF)

Definition. When X is a continuous random variable, and the CDF is differentiable everywhere, a Probability Density Function (PDF) is $f_X : \mathbb{R} \to \mathbb{R}$ such that



Source: Texas Instruments

Probability Review — Example Distributions

We will encounter many common distributions in this class, including:

Continuous

- Normal (Gaussian) distribution
- Uniform distribution (continuous)
- Gamma distribution
- Beta distribution
- Exponential distribution
- Chi-squared distribution
- Fisher-Snedecor distribution

Discrete

- Poisson distribution
- Binomial distribution
- Geometric distribution
- Hypergeometric distribution
- Discrete uniform distribution

Probability Review — Expectation

Probability Review — Expectation

Definition. Let $g(X)$ be an arbitrary test function $g : \mathbb{R} \to \mathbb{R}$.

Then for a discrete random variable

$$\mathbb{E}\left[g(X)\right] = \sum_{x \in \mathrm{Val}(x)} g(x) p_X(x)$$

And for a continuous random variable

$$\mathbb{E}\left[g(X)\right] = \int_{-\infty}^{\infty} g(x) f_X(x) dx$$

Probability Review — Expectation

Definition. Let $g(X)$ be an arbitrary test function $g : \mathbb{R} \to \mathbb{R}$.

Then for a discrete random variable

$$\mathbb{E}\left[g(X)\right] = \sum_{x \in \mathrm{Val}(x)} g(x) p_X(x)$$

And for a continuous random variable **(alternative notation)**

$$\mathbb{E}_{x \sim f_X}\left[g(x)\right] = \int_{-\infty}^{\infty} g(x) f_X(x) dx$$

Probability Review — Variance

Probability Review — Variance

Definition. The variance is a measure of how concentrated a random variable is around its mean, and is defined as

$$\operatorname{Var}[X] = \mathbb{E}\left[(X - \mathbb{E}[X])^2\right]$$

We can also write the variance as

$$\begin{aligned}
\mathbb{E}\left[(X - \mathbb{E}[X])^2\right] &= \mathbb{E}\left[X^2 - 2\mathbb{E}[X]X + \mathbb{E}[X]^2\right] \\
&= \mathbb{E}\left[X^2\right] - 2\mathbb{E}[X]\mathbb{E}[X] + \mathbb{E}[X]^2 \\
&= \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2
\end{aligned}$$

Probability Review — Properties of Expectation and Variance

A few properties of note for the expectation and variance:

Probability Review — Properties of Expectation and Variance

A few properties of note for the expectation and variance:

- $\mathbb{E}[a] = a$ for any constant $a \in \mathbb{R}$
- $\mathbb{E}[af(X)] = a\mathbb{E}[f(X)]$ for any constant $a \in \mathbb{R}$
- Linearity of Expectation: $\mathbb{E}[f(X) + g(X)] = \mathbb{E}[f(X)] + \mathbb{E}[g(X)]$
- $\text{Var}[a] = 0$ for any constant $a \in \mathbb{R}$
- $\text{Var}[af(X)] = a^2\text{Var}[f(X)]$ for any constant $a \in \mathbb{R}$
- If $X \perp Y$, then $\text{Var}[f(X) + g(X)] = \text{Var}[f(X)] + \text{Var}[g(X)]$

Probability Review — Joint Distributions (Two or More Random Variables)

Suppose we have two random variables, X and Y, and want to model their outcomes *simultaneously* during a random experiment.

Probability Review — Joint CDF

Probability Review — Joint CDF

Definition. The Joint Cumulative Distribution Function (joint CDF) of random variables X and Y is defined to be

$$F_{XY}(x, y) = P(X \leq x, Y \leq y)$$

By knowing the joint CDF, any event involving X and Y can be calculated!

Probability Review — Joint CDF

Definition. The Joint Cumulative Distribution Function (joint CDF) of random variables X and Y is defined to be

$$F_{XY}(x, y) = P(X \leq x, Y \leq y)$$

The joint CDF is related to the **marginal CDF** for each random variable by:

$$F_X(x) = \lim_{y \to \infty} F_{XY}(x, y)dy$$

$$F_Y(y) = \lim_{x \to \infty} F_{XY}(x, y)dx$$

Probability Review — Joint PDF

Probability Review — Joint PDF

Definition. Let X and Y be random variables with joint CDF $F_{XY}(x, y)$. Then the joint probability density function (joint PDF) is defined to be

$$f_{XY}(x, y) = \frac{\partial^2 F_{XY}(x, y)}{\partial x \partial y}$$

And, similarly to the one-dimensional case, we have that

$$\int_{x,y \in A} f_{XY}(x, y) dx dy = P((X, Y) \in A)$$

Probability Review — Marginal PDFs

Probability Review — Marginal PDFs

Definition. Let X and Y be random variables with joint PDF $f_{XY}(x, y)$. Then the marginal probability density functions (marginal PDFs) are defined to be

$$f_X(x) = \int_{-\infty}^{\infty} f_{XY}(x, y)dy$$

$$f_Y(y) = \int_{-\infty}^{\infty} f_{XY}(x, y)dx$$

Often say “marginalize out y” (top), or “marginalize out x” (bottom).

Probability Review — Conditional Distributions

Probability Review — Conditional Distributions

Seeks to answer: "What is the probability distribution over X when we know that Y has a certain value?"

Definition. The conditional distribution of X given that $Y = y$ is defined to be

$$f_{X|Y}(x \mid y) = \frac{f_{XY}(x, y)}{f_Y(y)}$$

assuming that $f_Y(y) \neq 0$.

Probability Review — Conditional Distributions

Seeks to answer: "What is the probability distribution over X when we know that Y has a certain value?"

Definition. The conditional distribution of X given that $Y = y$ is defined to be

$$f_{X_1|X_2,\ldots,X_n}(x_1 \mid x_2, \ldots, x_n) = \frac{f_{X_1,X_2,\ldots,X_n}(x_1, x_2, \ldots, x_n)}{f_{X_2,\ldots,X_n}(x_2, \ldots, x_n)}$$

assuming that the denominator is not equal to 0.

For higher dimensions!

Probability Review — Conditional Distributions

Seeks to answer: "What is the probability distribution over X when we know that Y has a certain value?"

Definition. The conditional distribution of X given that $Y = y$ is defined to be

$$p(x_1 \mid x_2, \ldots, x_n) = \frac{p(x_1, x_2, \ldots, x_n)}{p(x_2, \ldots, x_n)}$$

assuming that the denominator is not equal to 0.

Common notation for PDFs!

Probability Review — Conditional Distributions

Seeks to answer: "What is the probability distribution over X when we know that Y has a certain value?"

Note that the conditional PDF implies the following factorization:

$$f_{X|Y}(x \mid y) = \frac{f_{XY}(x, y)}{f_Y(y)} \quad \iff$$

$$f_{XY}(x, y) = f_{X|Y}(x \mid y) f_Y(y) = f_{Y|X}(y \mid x) f_X(x)$$

This will show up a lot!

Probability Review — Bayes Rule

Wikipedia

An important expression of conditional distribution of random variables!

Probability Review — Bayes Rule

Wikipedia

An important expression of conditional distribution of random variables!

Definition. Bayes Rule is defined to be

$$f_{Y|X}(y \mid x) = \frac{f_{XY}(x,y)}{f_X(x)} = \frac{f_{X|Y}(x \mid y) f_Y(y)}{\int_{-\infty}^{\infty} f_{X|Y}(x \mid y') f_Y(y') dy'}$$

Probability Review — Bayes Rule

Wikipedia

An important expression of conditional distribution of random variables!

Definition. Bayes Rule is defined to be

Likelihood

Prior

$$f_{Y|X}(y \mid x) = \frac{f_{XY}(x,y)}{f_X(x)} = \frac{f_{X|Y}(x \mid y) f_Y(y)}{\int_{-\infty}^{\infty} f_{X|Y}(x \mid y') f_Y(y') dy'}$$

Marginal / Evidence

Probability Review — Expectation (for Joint PDF)

Probability Review — Expectation (for Joint PDF)

Suppose we have two random variables and a test function $g : \mathbb{R}^2 \to \mathbb{R}$. Then the expected value can be written

$$\mathbb{E}_{x,y\sim f_{XY}}\left[g(x,y)\right] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y) f_{XY}(x,y) dxdy$$

Notation often shortened to:

$$\mathbb{E}_{x,y}\left[g(x,y)\right] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y) f_{XY}(x,y) dxdy$$

Probability Review — Covariance

Probability Review — Covariance

Definition. The covariance of two random variables X and Y is defined to be

$$\operatorname{Cov}[X, Y] = \mathbb{E}\left[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])\right]$$

Describes whether two variables tend to move in the same direction (positive covariance) or opposite directions (negative covariance).

Probability Review — Covariance

Definition. The covariance of two random variables X and Y is defined to be

$$\operatorname{Cov}[X, Y] = \mathbb{E}\left[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])\right]$$

Which alternatively can be written

$$\begin{aligned}
\operatorname{Cov}[X, Y] &= \mathbb{E}\left[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])\right] \\
&= \mathbb{E}\left[XY - X\mathbb{E}[Y] - Y\mathbb{E}[X] + \mathbb{E}[X]\mathbb{E}[Y]\right] \\
&= \mathbb{E}\left[XY\right] - \mathbb{E}[X]\mathbb{E}[Y] - \mathbb{E}[Y]\mathbb{E}[X] + \mathbb{E}[X]\mathbb{E}[Y] \\
&= \mathbb{E}\left[XY\right] - \mathbb{E}[X]\mathbb{E}[Y]
\end{aligned}$$

Probability Review — Chain Rule (for Joint PDFs)

Probability Review — Chain Rule (for Joint PDFs)

Similar to the expression above, for a joint pdf we can write the chain rule as:

Using more-common notation for PDFs

Joint PDF factorization

(repeatedly!)

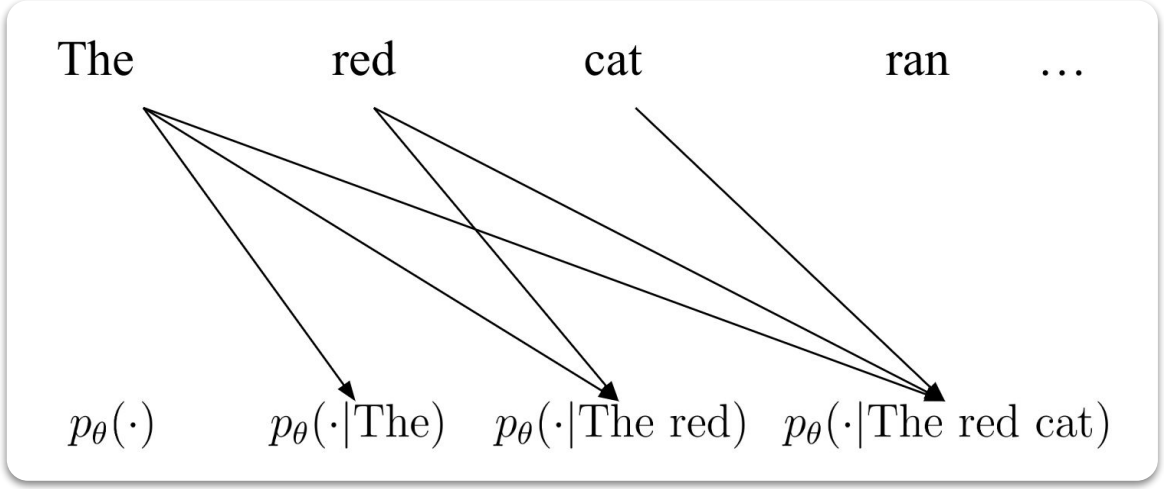
$$
\begin{aligned}
p(x_1, x_2, \ldots, x_n) &= p(x_n \mid x_1, x_2, \ldots, x_{n-1}) p(x_1, x_2, \ldots, x_{n-1}) \\
&= p(x_n \mid x_1, x_2, \ldots, x_{n-1}) p(x_{n-1} \mid x_1, x_2, \ldots, x_{n-2}) p(x_1, x_2, \ldots, x_{n-2}) \\
&= \ldots \\
&= p(x_1) \prod_{i=2}^{n} p(x_i \mid x_1, \ldots, x_{i-1})
\end{aligned}
$$

Full joint PDF factorization

Probability Review — Chain Rule (for Joint PDFs)

$$
\begin{aligned}
p(x_1, x_2, \ldots, x_n) &= p(x_n \mid x_1, x_2, \ldots, x_{n-1}) p(x_1, x_2, \ldots, x_{n-1}) \\
&= p(x_n \mid x_1, x_2, \ldots, x_{n-1}) p(x_{n-1} \mid x_1, x_2, \ldots, x_{n-2}) p(x_1, x_2, \ldots, x_{n-2}) \\
&= \ldots \\
&= p(x_1) \prod_{i=2}^{n} p(x_i \mid x_1, \ldots, x_{i-1})
\end{aligned}
$$

Autoregressive Models (LLMs)!



Probability Review — Independence (for Joint PDFs)

Probability Review — Independence (for Joint PDFs)

Using more-common notation for PDFs

We say that random variables $X_1, X_2, \ldots, X_n$ are independent if

$$p(x_1, x_2, \ldots, x_n) = p(x_1)p(x_2)\cdots p(x_n)$$

Probability Review — Independence (for Joint PDFs)

Using more-common notation for PDFs

We say that random variables $X_1, X_2, \ldots, X_n$ are independent if

$$p(x_1, x_2, \ldots, x_n) = p(x_1)p(x_2)\cdots p(x_n)$$

Contrast this with the previous factorization (no independence assumption):

$$p(x_1, x_2, \ldots, x_n) = p(x_1)\prod_{i=2}^{n} p(x_i \mid x_1, \ldots, x_{i-1})$$

⇒ intuitively, independence means that knowing the value of one variable does not have any effect on the conditional probability distribution of the other variable.

Probability Review — Random Vectors

When working with n random variables, often convenient to describe them as a **random vector**

$$X : \Omega \to \mathbb{R}^n \qquad X = [X_1, X_2, \ldots, X_n]^\top$$

Probability Review — Random Vectors

When working with n random variables, often convenient to describe them as a **random vector**

$$X : \Omega \to \mathbb{R}^n \qquad X = [X_1, X_2, \ldots, X_n]^\top$$

Where we can write the expected value, given a test function $g : \mathbb{R}^n \to \mathbb{R}$, as

$$\begin{aligned} E[g(X)] = E_x[g(x)] &= \int_{\mathbb{R}^n} g(x_1, x_2, \ldots, x_n) p(x_1, x_2, \ldots, x_n)\, dx_1 dx_2 \cdots dx_n \\ &= \int_{\mathbb{R}^n} g(x) p(x)\, dx, \qquad\qquad x = (x_1, \ldots, x_n) \in \mathbb{R}^n \end{aligned}$$

Probability Review — Covariance Matrix

Probability Review — Covariance Matrix

A covariance matrix is the n x n square matrix with entries $\Sigma_{ij} = \mathrm{Cov}[X_i, X_j]$ written as

$$\Sigma = \begin{bmatrix} \mathrm{Cov}[X_1, X_1] & \cdots & \mathrm{Cov}[X_1, X_n] \\ \vdots & \ddots & \vdots \\ \mathrm{Cov}[X_n, X_1] & \cdots & \mathrm{Cov}[X_n, X_n] \end{bmatrix}$$

$$= \begin{bmatrix} E[X_1^2] - E[X_1]E[X_1] & \cdots & E[X_1 X_n] - E[X_1]E[X_n] \\ \vdots & \ddots & \vdots \\ E[X_n X_1] - E[X_n]E[X_1] & \cdots & E[X_n^2] - E[X_n]E[X_n] \end{bmatrix}$$

$$= E[XX^\top] - E[X]E[X]^\top = \cdots = E\big((X - E[X])(X - E[X])^\top\big).$$

Next Class

Next Class

Lecture: PGMs — using ideas from both probability and graph theory to derive efficient machine learning algorithms

After:

- Class introductions and research interests.
- Everyone should share their name, and a couple of research interests. We will record this in a doc and share it with the class.
  - The goal is to help facilitate matching for group projects!