STA414
Statistical Methods for Machine Learning II

January 2020

David Duvenaud and Jesse Bettencourt

Motivating Questions:

- How could I build a system to automatically fill in missing parts of an image, given examples?
- Which test would give me the most useful information about a patient?
- How can I handle missing data?
- How can I figure out how good a player is from their wins and losses?

Lecture structure:

- **First two hours:** Going through concepts, mostly matching course notes. Medium pace.

- **Last hour:** Tutorial - worked examples. Slow pace. Lots of time for questions. Feel free to skip / leave.

- **Lecturers:** Each week after this one, either me or Jesse will cover both sections + tutorial.

Today

- Course information and overview
  - Expectations, course structure, evaluations
  - Learning objectives for course
- Overview of probabilistic machine learning
  - Examples
  - Tools of the trade
- **Tutorial**: Installing Julia, basics of Git

Learning Outcomes: Today

- Know what topics are and aren't in the course.
- An idea of if you have the background + how hard the material will be.
- What you should be able to do with this knowledge.
- Know how to set up a computing environment

Scope of course

- Designing, fitting, and interpreting parametric probabilistic models.
    - Conditioning, marginalizing, Normalized versus unnormalized distributions, Graphical models
    - Neural nets, gradient-based optimization, automatic differentiation
    - Approximate inference, sampling, variational inference
- A bit of simple decision theory
- Standard software tools: numerics, autodiff, git

Evaluation

- **Assignment 0**: 10% (Friday, Jan 24)
    - Onboarding. Basic distributions, sampling, linear algebra, autodiff, unit tests.
- **Assignment 1**: 13.3% (Friday, Feb 7)
    - Deriving and fitting high-dimensional probabilistic models. (Probably)
- **Midterm**: 20% (Around Thursday, Feb ~13)
    - Basics of graphical models, conditioning, sampling, fitting. (Probably)
- **Assignment 2**: 13.3% (Friday, Mar 13)
    - Fitting multi-factor latent variable models. (Probably)
- **Assignment 3**: 13.3% (Friday, Apr 3)
    - Fitting neural net generative models (e.g. variational autoencoder). (Probably)
- **Final Exam**: 30% (TBD)

Tools of the trade: Probability

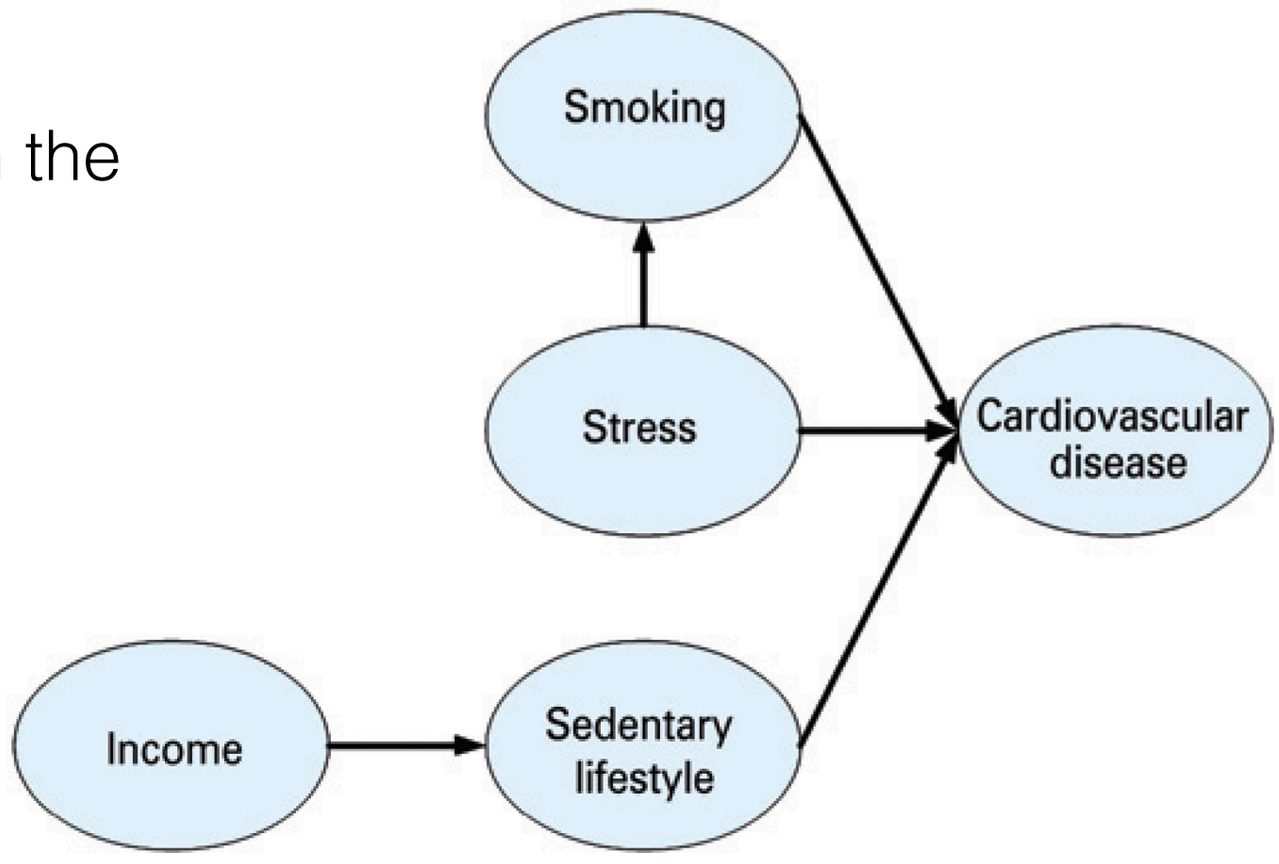
- Probabilities represent uncertainty about a fixed but unknown quantity, conditioned on some information

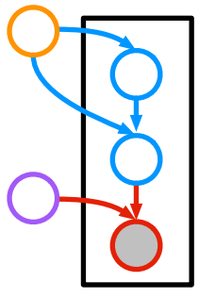
- Inference and prediction is easy!: "Just write down the joint probability of everything, and integrate out everything you don't know." - MacKay

- No need to pretend to identify parameters, except for computational efficiency

Tools of the trade: Graphical Models

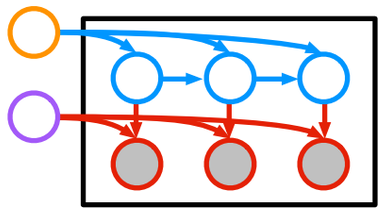
- Giant joint pdfs are hard to reason about
- Conditional independencies often the most important fact about a joint distribution
- Can encode and reason about conditional independence using graphs
- Lots of fun algorithms
- De-emphasized, since often simpler to assume everything is connected





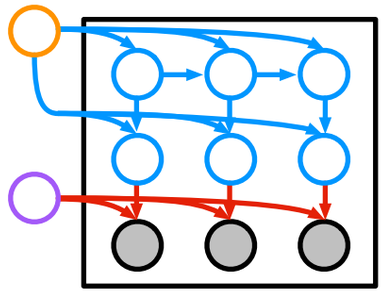
[1]

Gaussian mixture model



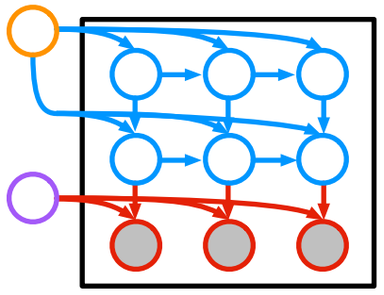
[2]

Linear dynamical system



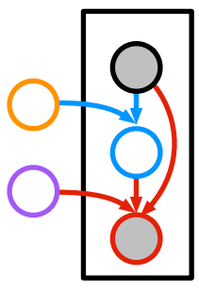
[3]

Hidden Markov model



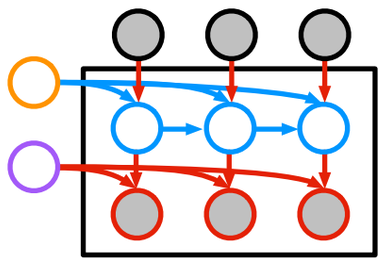
[4]

Switching LDS



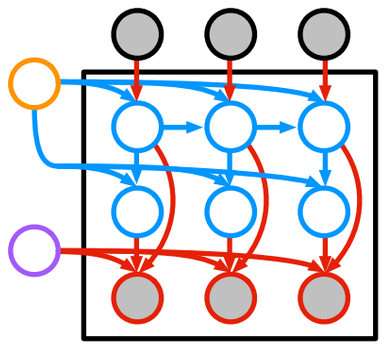
[5]

Mixture of Experts



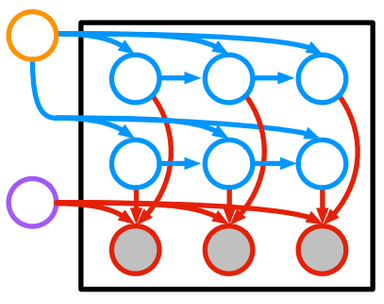
[2]

Driven LDS



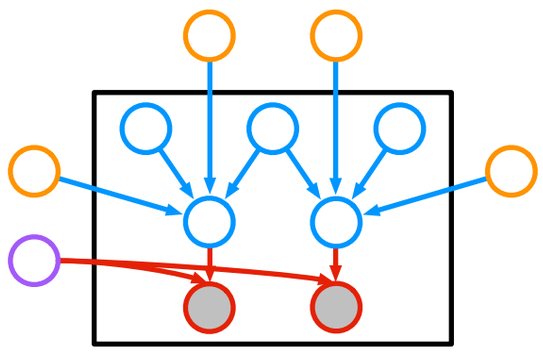
[6]

IO-HMM



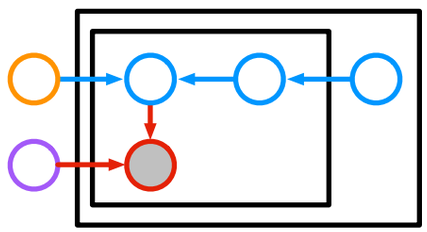
[7]

Factorial HMM



[8,9]

Canonical correlations analysis



[10]

admixture / LDA / NMF

[1] Palmer, Wipf, Kreutz-Delgado, and Rao. Variational EM algorithms for non-Gaussian latent variable models. NIPS 2005.
[2] Ghahramani and Beal. Propagation algorithms for variational Bayesian learning. NIPS 2001.
[3] Beal. Variational algorithms for approximate Bayesian inference, Ch. 3. U of London Ph.D. Thesis 2003.
[4] Ghahramani and Hinton. Variational learning for switching state-space models. Neural Computation 2000.
[5] Jordan and Jacobs. Hierarchical Mixtures of Experts and the EM algorithm. Neural Computation 1994.
[6] Bengio and Frasconi. An Input Output HMM Architecture. NIPS 1995.
[7] Ghahramani and Jordan. Factorial Hidden Markov Models. Machine Learning 1997.
[8] Bach and Jordan. A probabilistic interpretation of Canonical Correlation Analysis. Tech. Report 2005.
[9] Archambeau and Bach. Sparse probabilistic projections. NIPS 2008.
[10] Hoffman, Bach, Blei. Online learning for Latent Dirichlet Allocation. NIPS 2010.

Courtesy of Matthew Johnson

Tools of the trade: Neural Networks

- Not profound or especially mysterious: Just a large nonlinear parametric function.
- Can basically fit anything if we overparameterize enough and use gradients.
- Main issues: Overfitting, non-differentiable objectives, hard to debug
- Show autograd demo

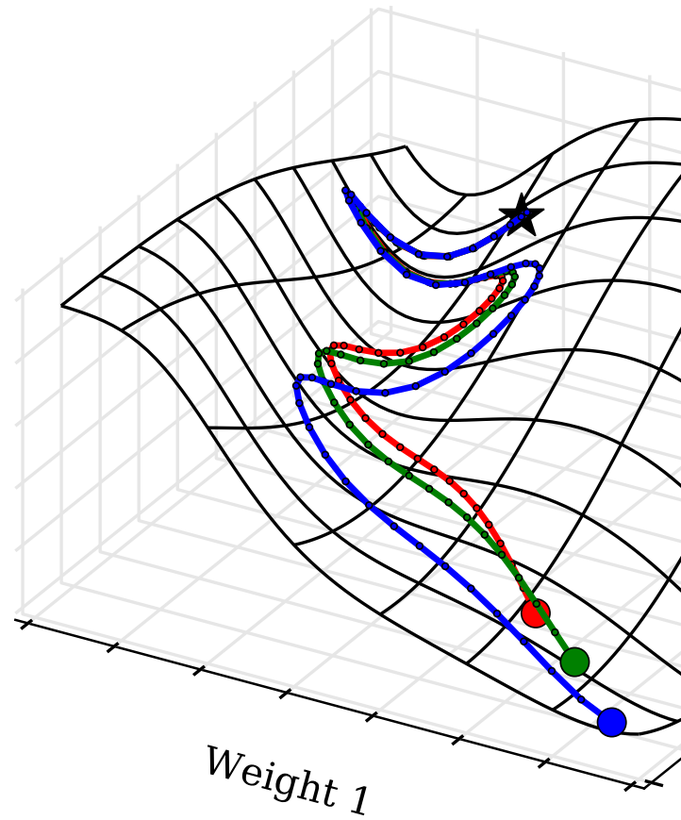
Source: xkcd

Tools of the trade: Gradient-based optimization

- Unconstrained, high-dimensional, stochastic, first-order gradient descent is surprisingly applicable

- Hinton: SGD "works much better than anyone had any right to expect".

- More parameters -> more progress before getting stuck.

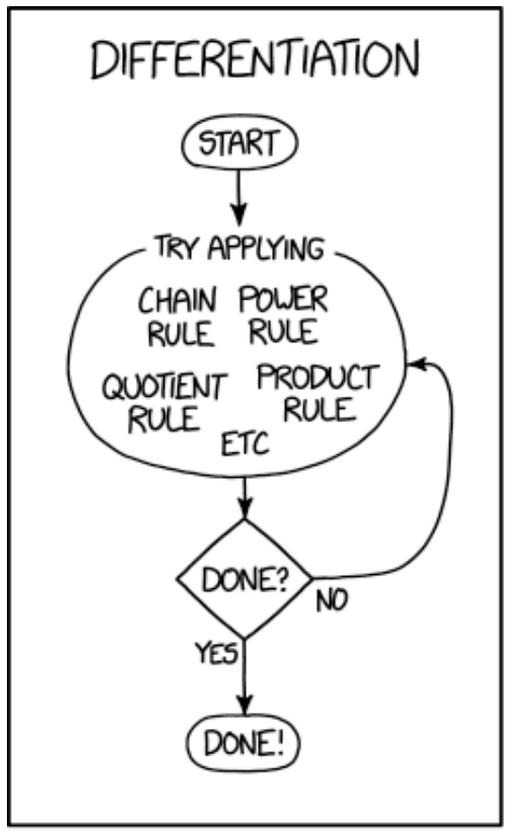


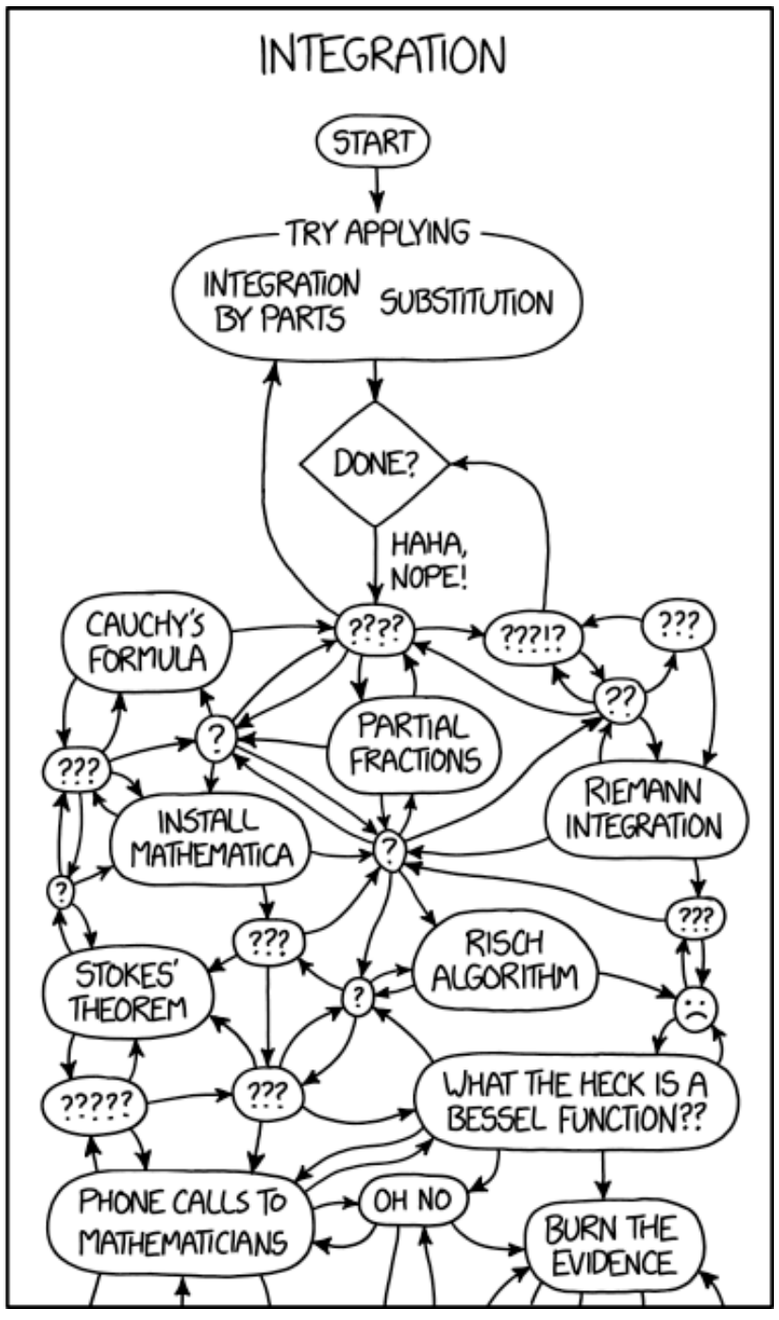
Tools of the trade: Automatic Differentiation

- Reverse-mode grads has same asymptotic time cost as original function

- Biggest change in last 10 years of ML practice

- Vector-Jacobian products are cheap

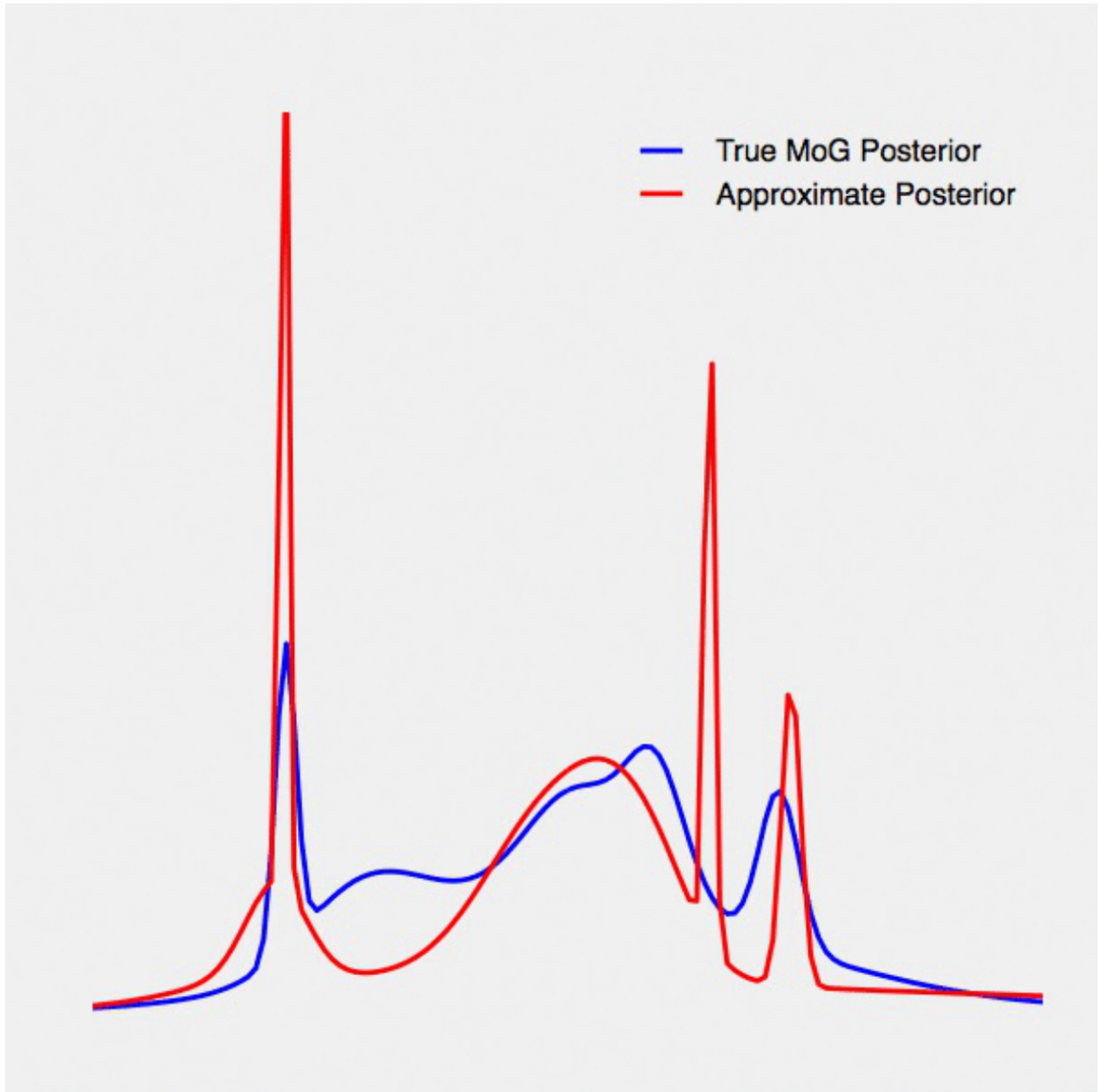




Source: xkcd

Tools of the trade: Approximate Inference

- Gradient based methods:
  - Variational inference
  - MCMC
- Jointly optimize + integrate.
- Show autograd demo

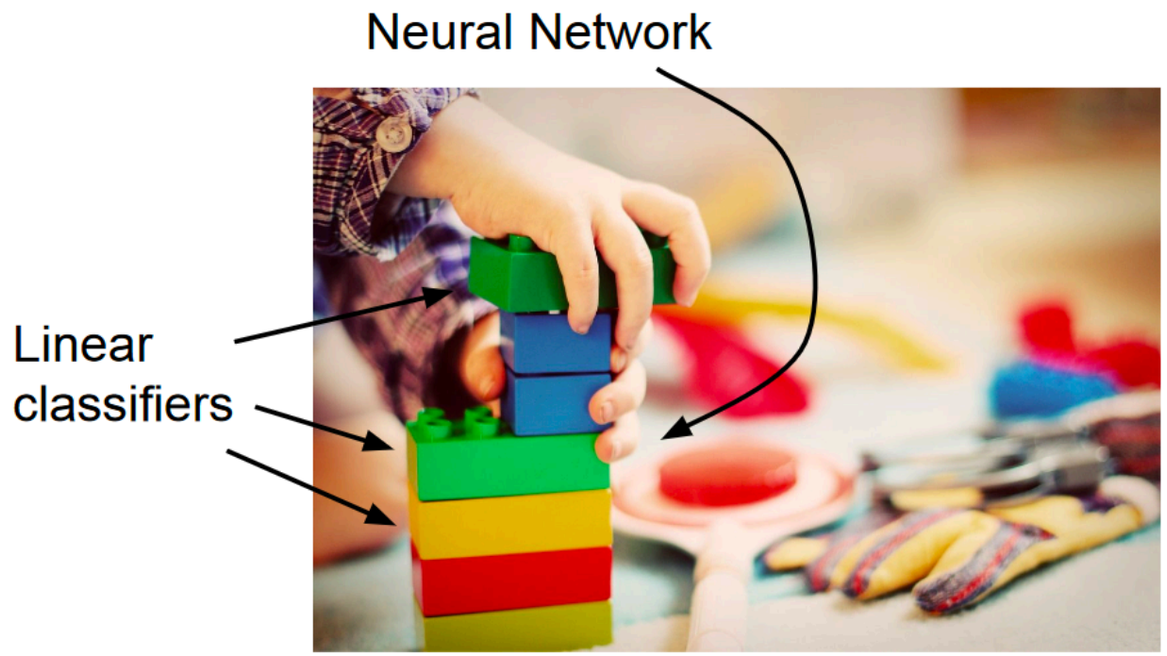


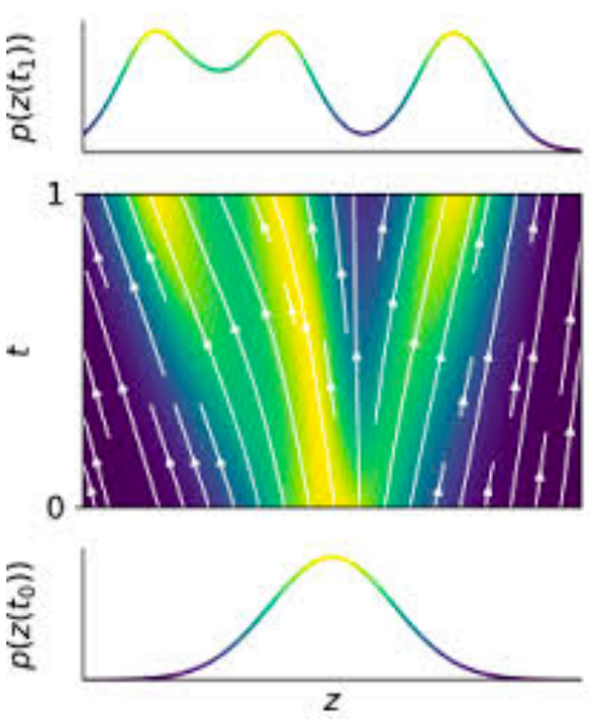
What can you build with these tools?

- Naive Bayes, Mixture of Gaussians, Logistic Regression, Bayesian Linear Regression, Hidden Markov Models, Factor Analysis

- Neural network classifiers, LSTMs, RNNs, Transformers, Convnets, Neural ODEs

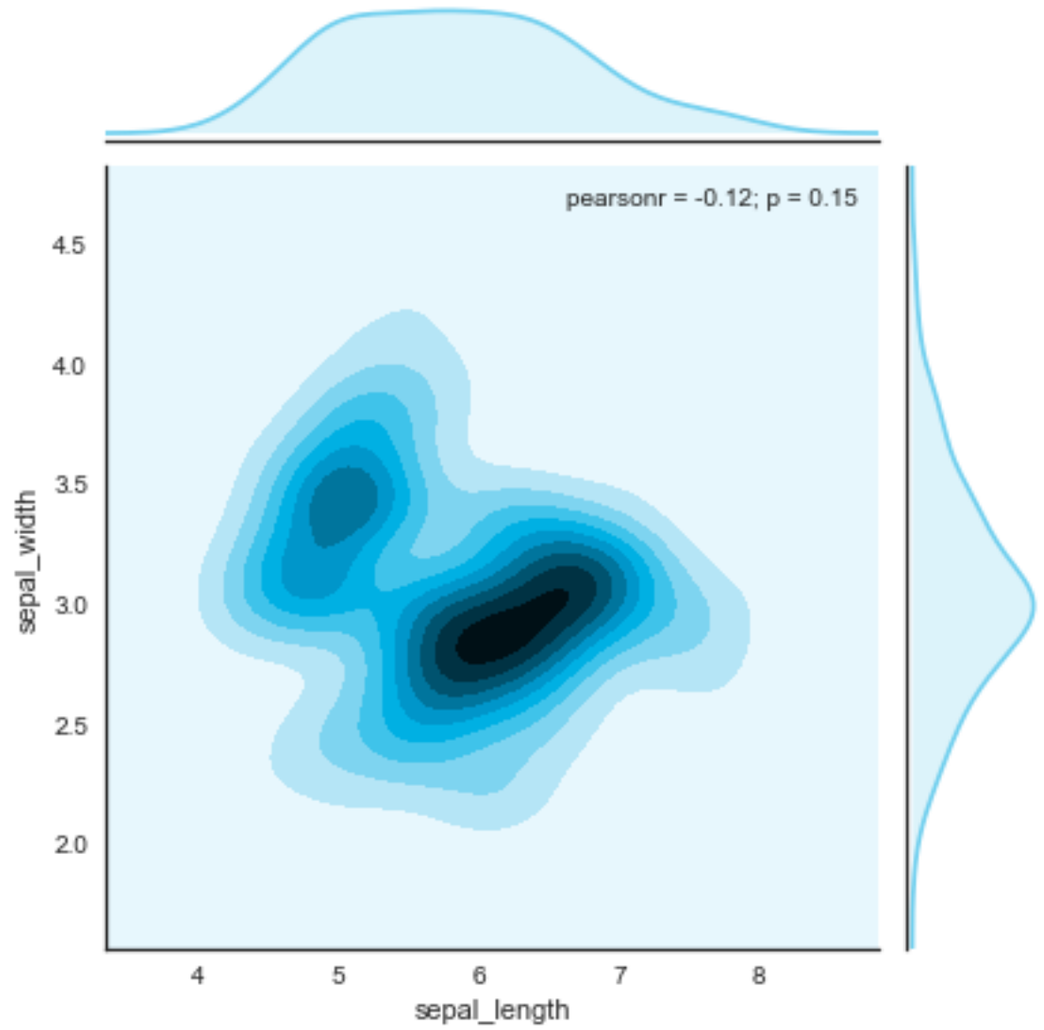
- Variational Autoencoders, Generative Adversarial Networks, Normalizing Flows





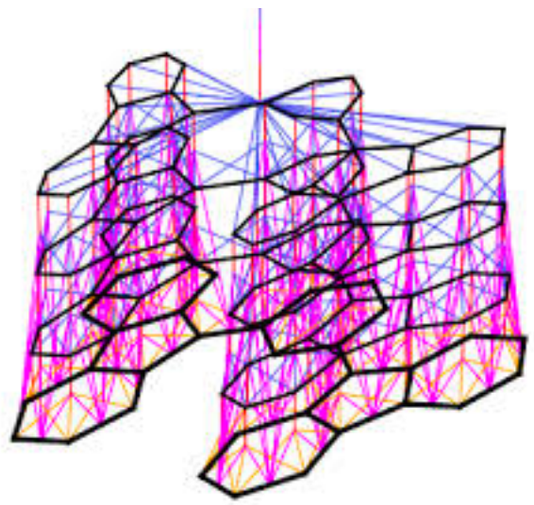
What can do with these models?

- Extend existing models.
  - E.g. What if we know the age of only some of our users?
- Sanity check data
  - E.g. Which piece of data in this form is most surprising?



What's not in scope?

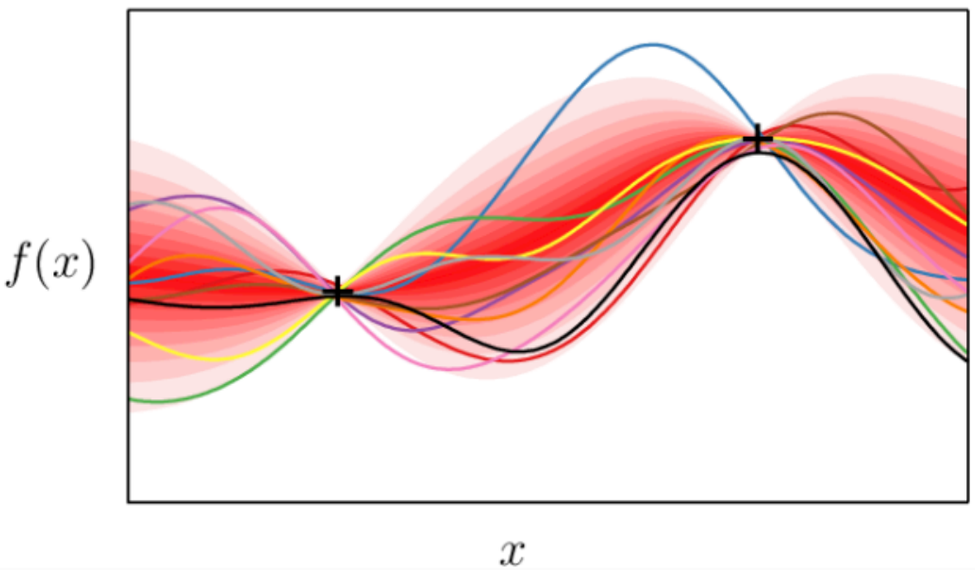
- Statistical Learning Theory. See e.g. Dan Roy or Murat Erdogdu's courses.



- Fancy neural network architectures. See e.g. Roger Grosse's course

- Logic-based AI, reasoning, discrete search. See e.g. Sheila McLiraith or Faheim Baccus.

- Nonparametrics, e.g. kernel density estimation, k-NN, Gaussian processes, support vector machines, Indian Buffet processes.



Tools of the trade: Julia and Python

- Julia: Simple, unified interface with autodiff. Decent error messages.

  - More support from course materials.

- Python: Allowed, but initially can't use frameworks' network layers, initializers, or optimizers.

  - Suggested: Jax, PyTorch

  - Gotchas: need to learn both Python and a framework on top. Bad error messages.

Intimidated?

- Look at HW0. Only need to fill in blanks. Will release tomorrow.

- Will provide starter code / skeleton for at least most of the assignments.

- Stick around for intro + tutorial

Why not R?

- Main reason: No reverse-mode autodiff!
  - Radford Neal is working on this.
- Other reasons:
  - Me, Jesse and some TAs don't know much R.
  - Students write slow nested loops (OK in Julia)
  - Limited GPU support, limited composability.

Tools of the trade: Git

- Version control is table stakes for industry, collaboration, your own sanity.

- Github demos add a lot to a resume.

- Assignments will be due through Github classroom.

GitHub

Extra Resources

- No required textbook. All tested material in lecture notes on website.
- David MacKay (2003) *Information Theory, Inference, and Learning Algorithms.* Great intro, dated on methods.
- Christopher M. Bishop (2006) *Pattern Recognition and Machine Learning.* Great intro, dated on methods.
- Kevin Murphy (2012), *Machine Learning: A Probabilistic Perspective*. Up-to-date, encyclopaedic.
- Trevor Hastie, Robert Tibshirani, Jerome Friedman (2009) *The Elements of Statistical Learning*
- Deep Learning (2016) Goodfellow, Bengio, Courville.

My Origin Story

- https://bayes.wustl.edu/etj/prob/book.pdf

- Derives probability from scratch

- Part manifesto

Probability Theory:
The Logic of Science

by
E. T. Jaynes
Wayman Crow Professor of Physics
Washington University
St. Louis, MO 63130, U. S. A.

ML as a bag of tricks

Special cases:

- K-means
- Kernel Density Estimation
- Support Vector Machines
- Boosting
- Random Forests
- K-Nearest Neighbors

Extensible family:

- Mixture of Gaussians
- Latent variable models
- Gaussian processes
- Deep neural nets
- Bayesian neural nets
- Attention-based models

Regularization as a bag of tricks

Fast special cases:

- Early stopping
- Ensembling
- L2 Regularization
- Gradient noise
- Dropout
- Expectation-Maximization

Extensible family:

- Stochastic variational inference

AI as a bag of tricks

Russel and Norvig's parts of AI:

- Machine learning
- Natural language processing
- Knowledge representation
- Automated reasoning
- Computer vision
- Robotics

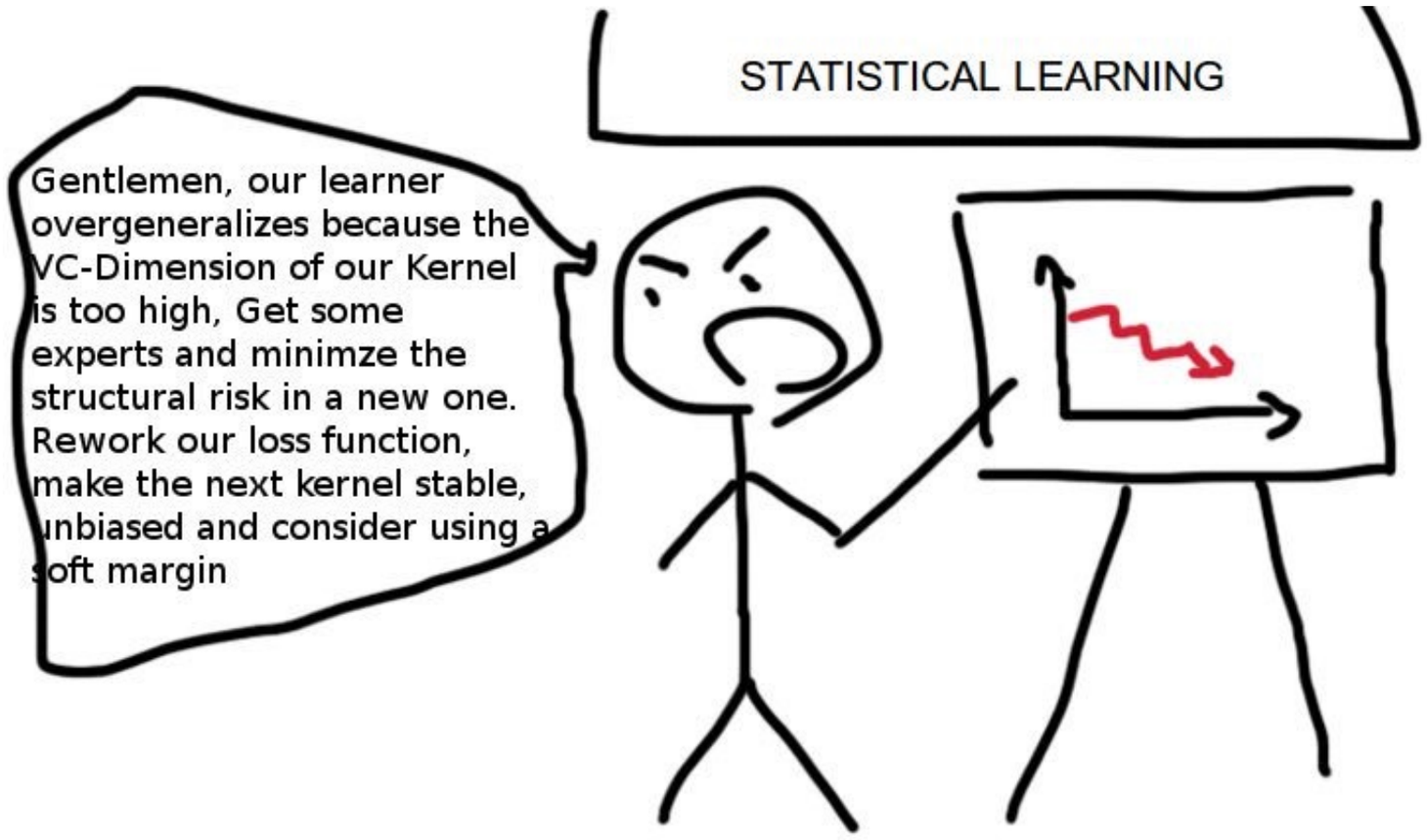
Extensible family:

- Deep probabilistic latent-variable models + decision theory
- a.k.a. Model-based Reinforcement learning

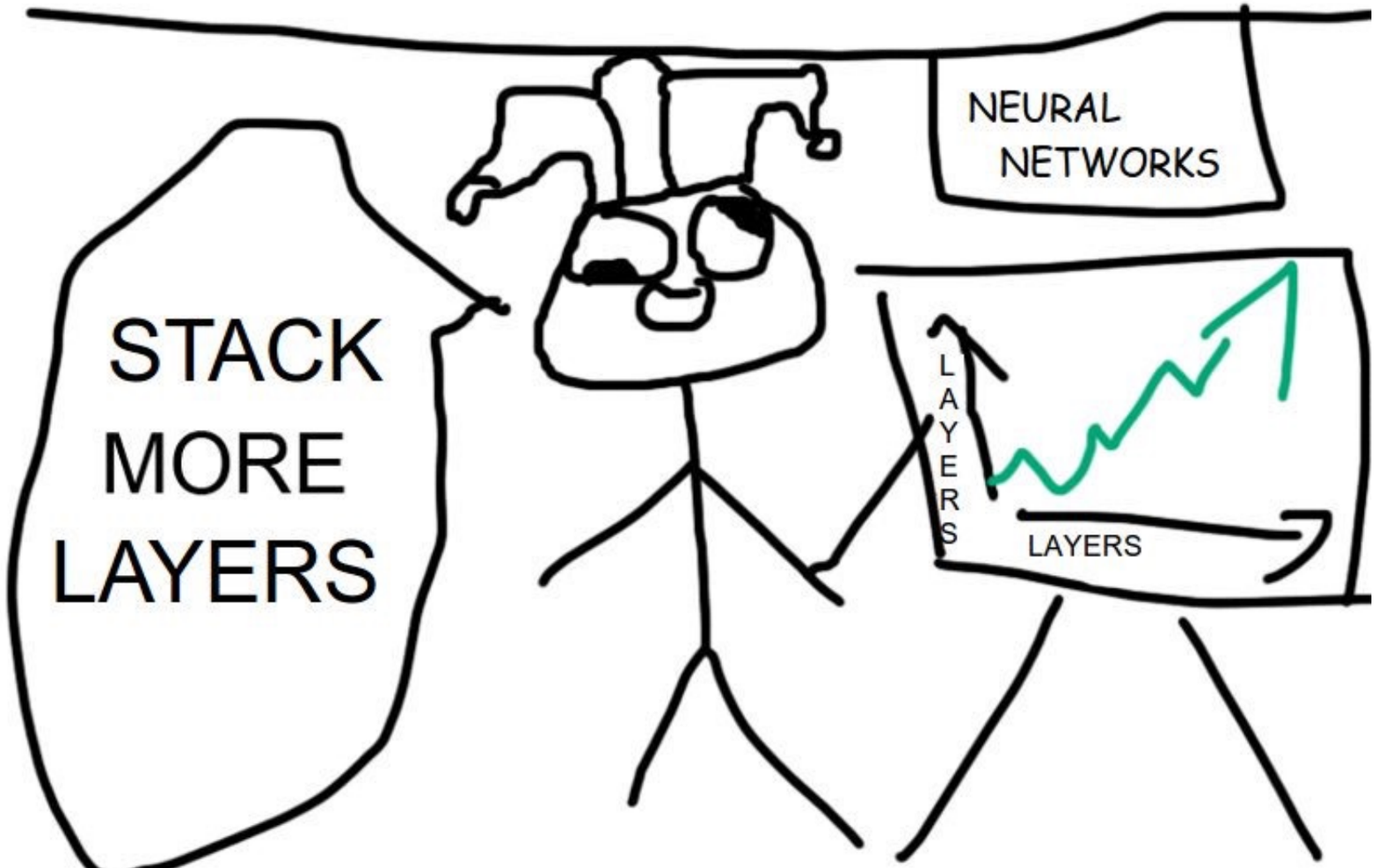
Stats vs Machine Learning

- Statistician: Look at dataset, consider the problem, design an interpretable model
    - Want guarantees, few assumptions, explanations
- ML: Mostly only predictions. Must handle new data automatically.
    - No way around making assumptions. Just make model big enough, hopefully it includes something close to the truth.
        - Model needs to have a million parameters somewhere, reality is messy.
    - Can’t use guarantees or bounds in practice, so empirically choose model details
- Probabilistic ML: Distinguish model from fitting algorithm

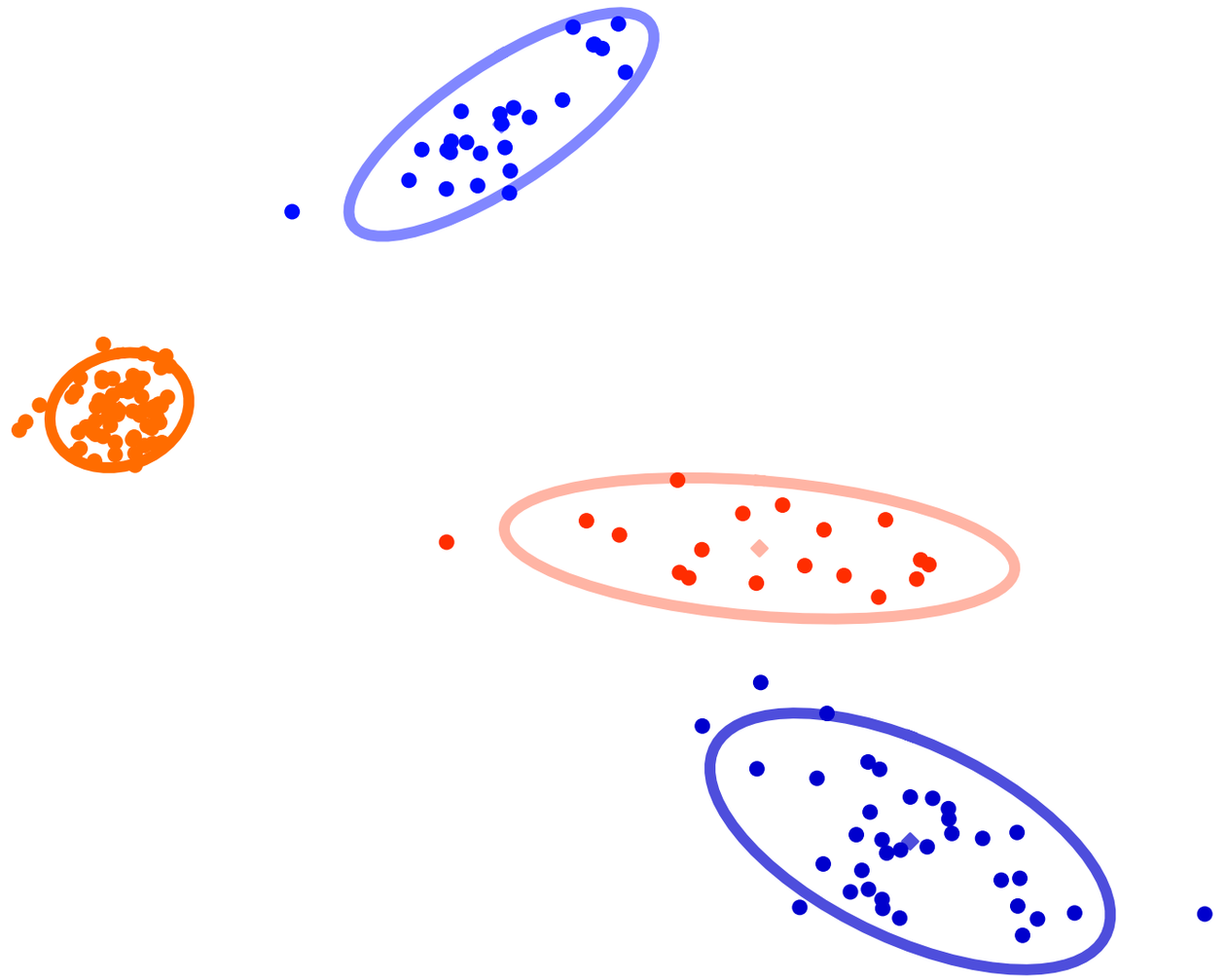
Statistical Learning vs Deep learning

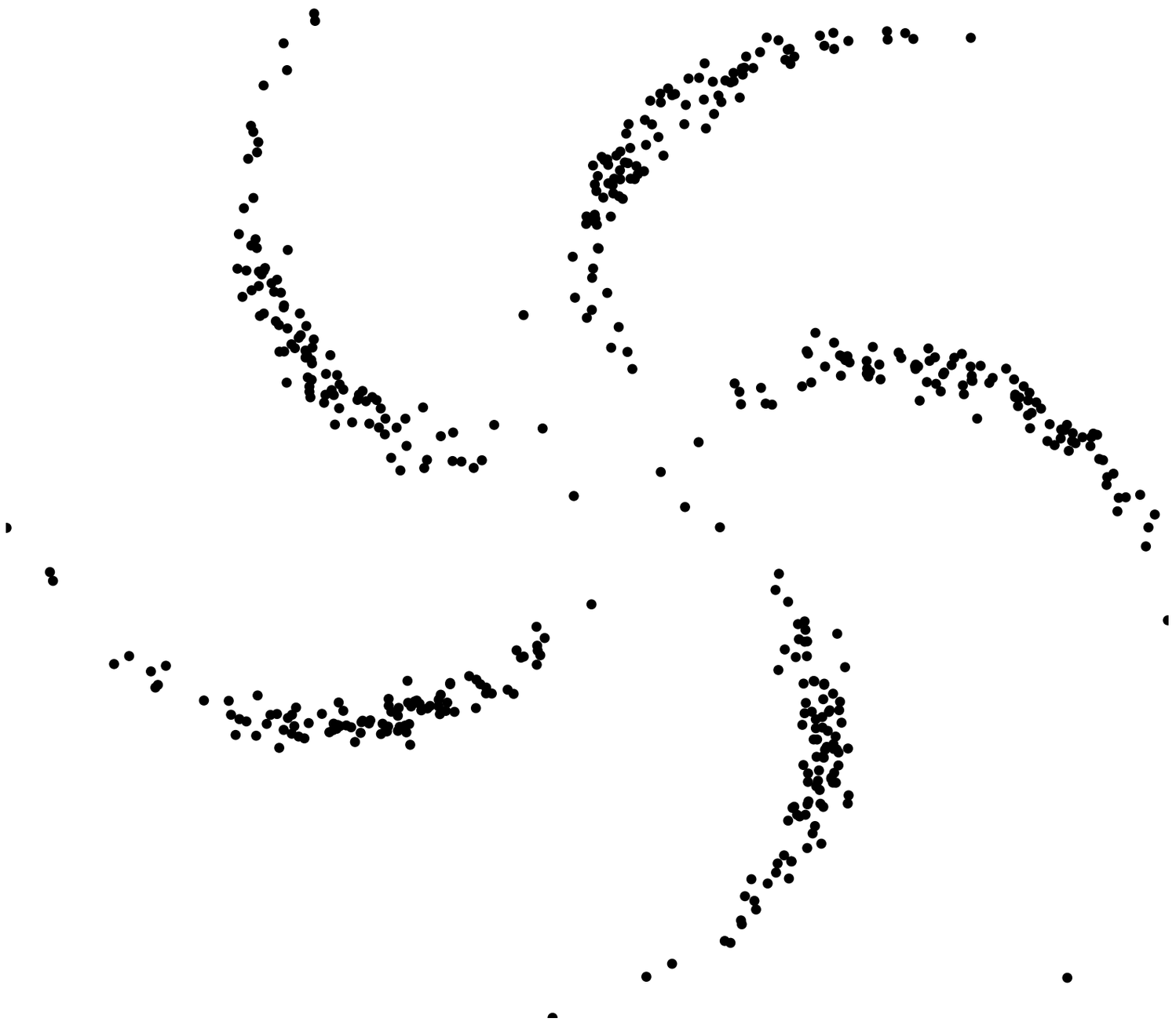


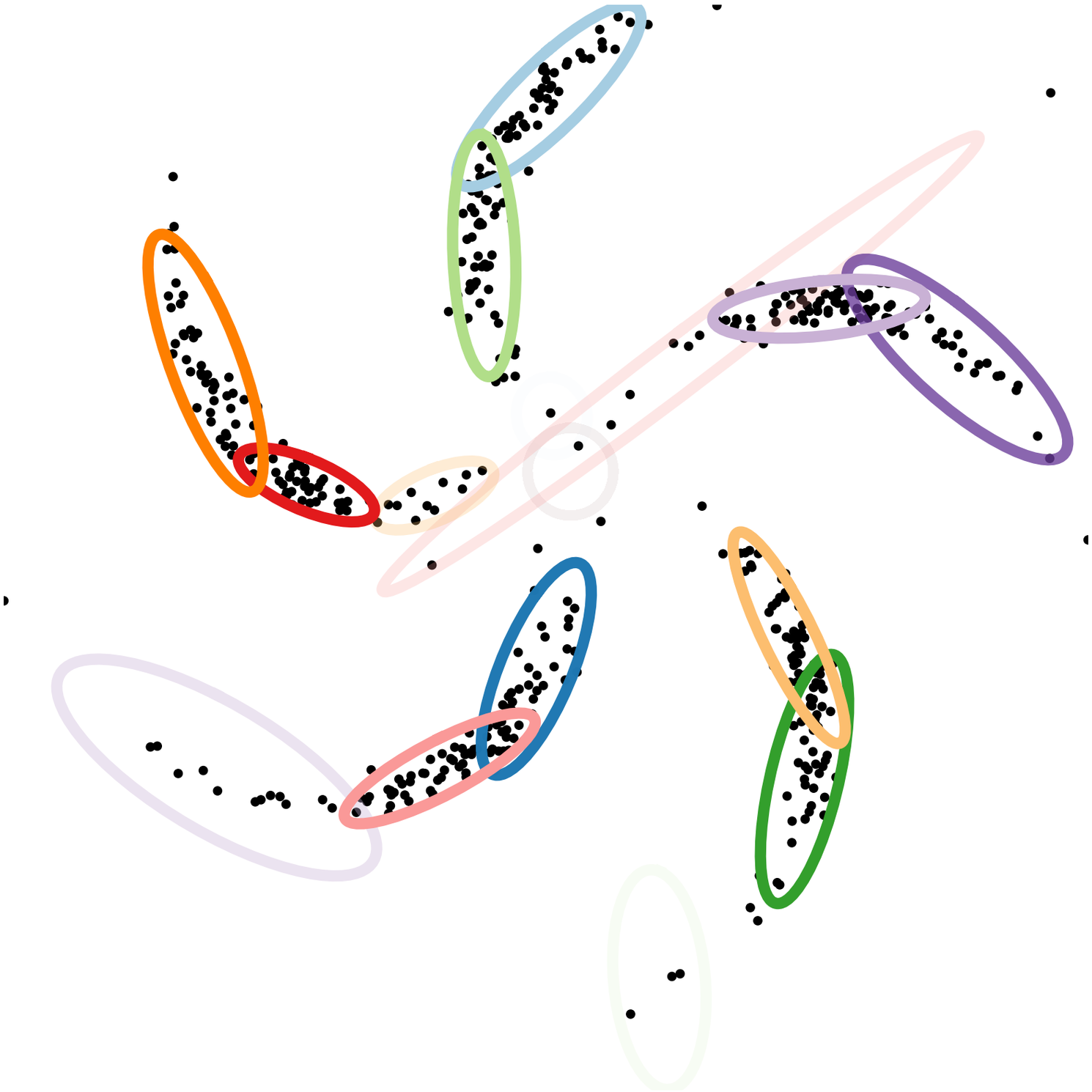
- Conceptually, a lot going on, mathematically and algorithmically simpler

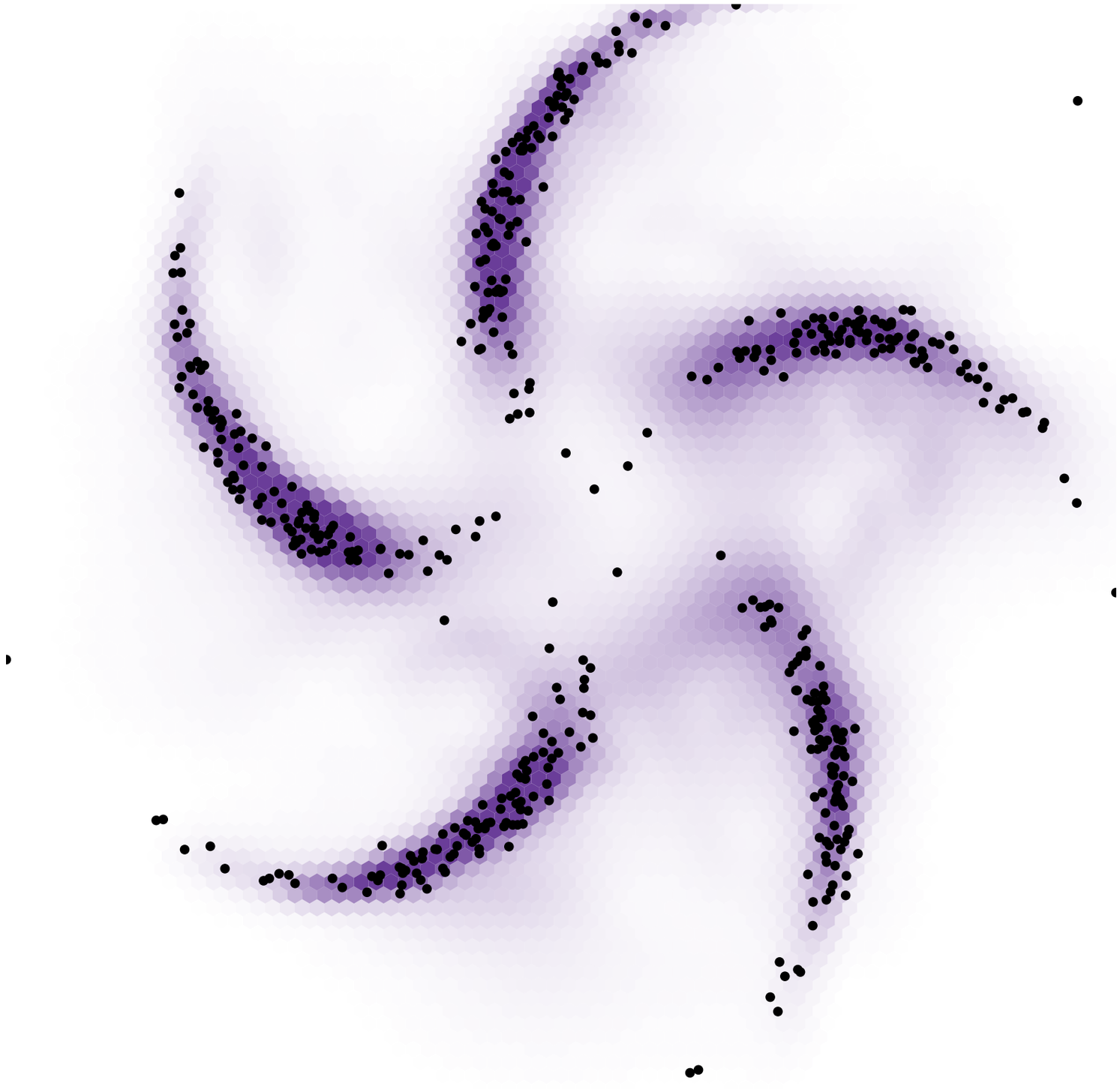


Examples









Probabilistic graphical models	Deep learning
+ structured representations	– neural net “goo”
+ priors and uncertainty	– difficult parameterization
+ data and computational efficiency	– can require lots of data
– rigid assumptions may not fit	+ flexible
– feature engineering	+ feature learning
– top-down inference	+ recognition networks

Differentiable models

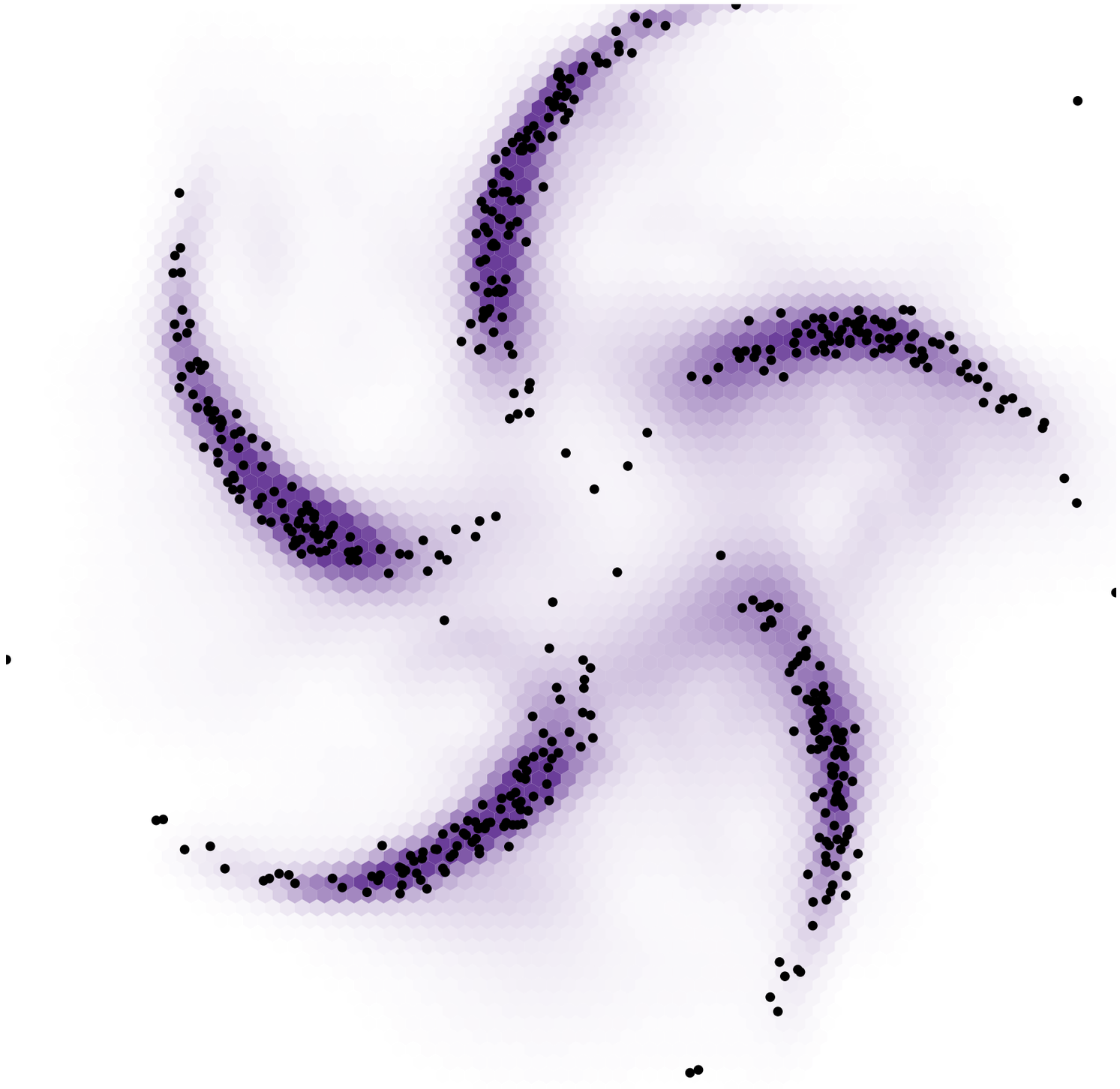
- Model distributions implicitly by a variable pushed through a deep net:

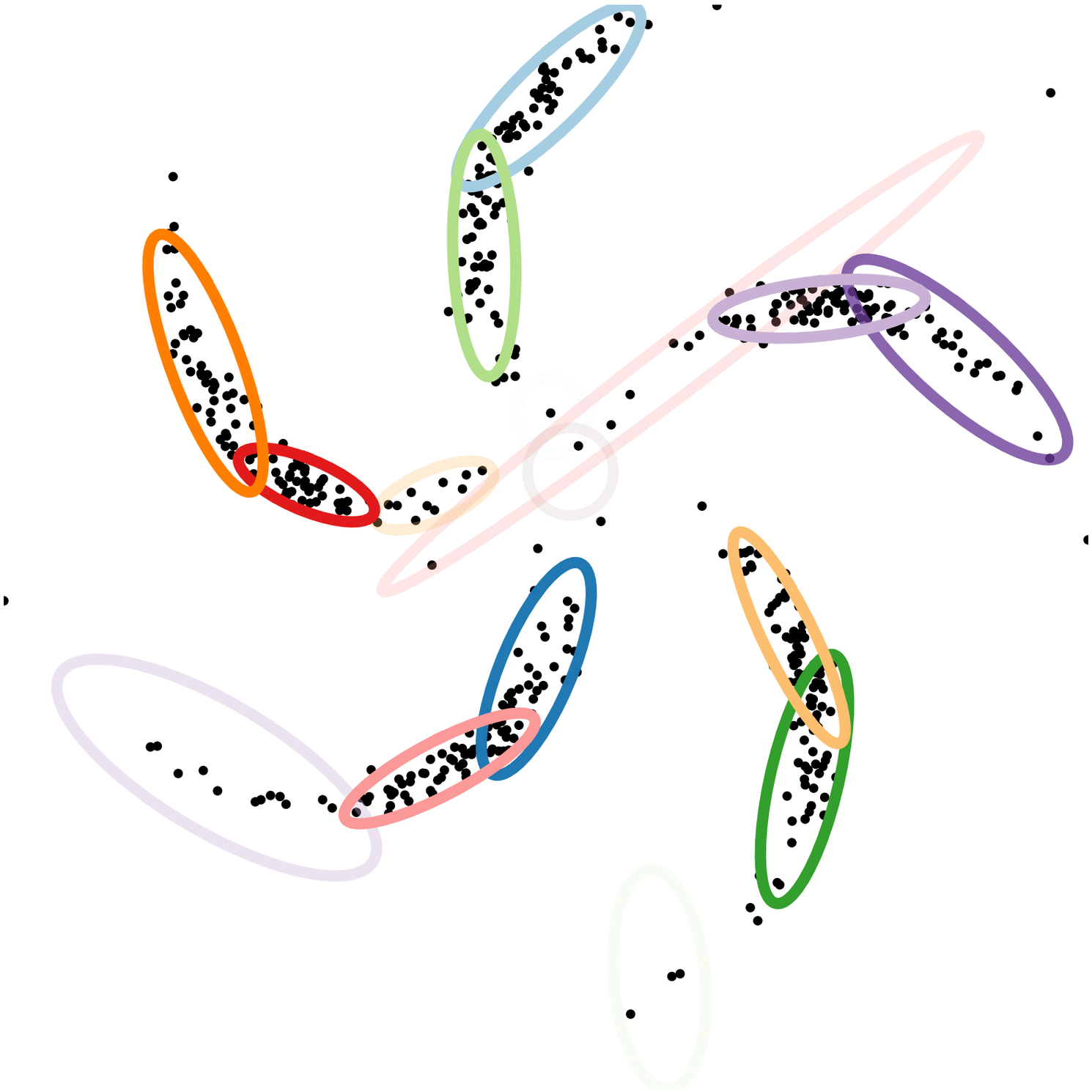
$$y = f_\theta(x)$$

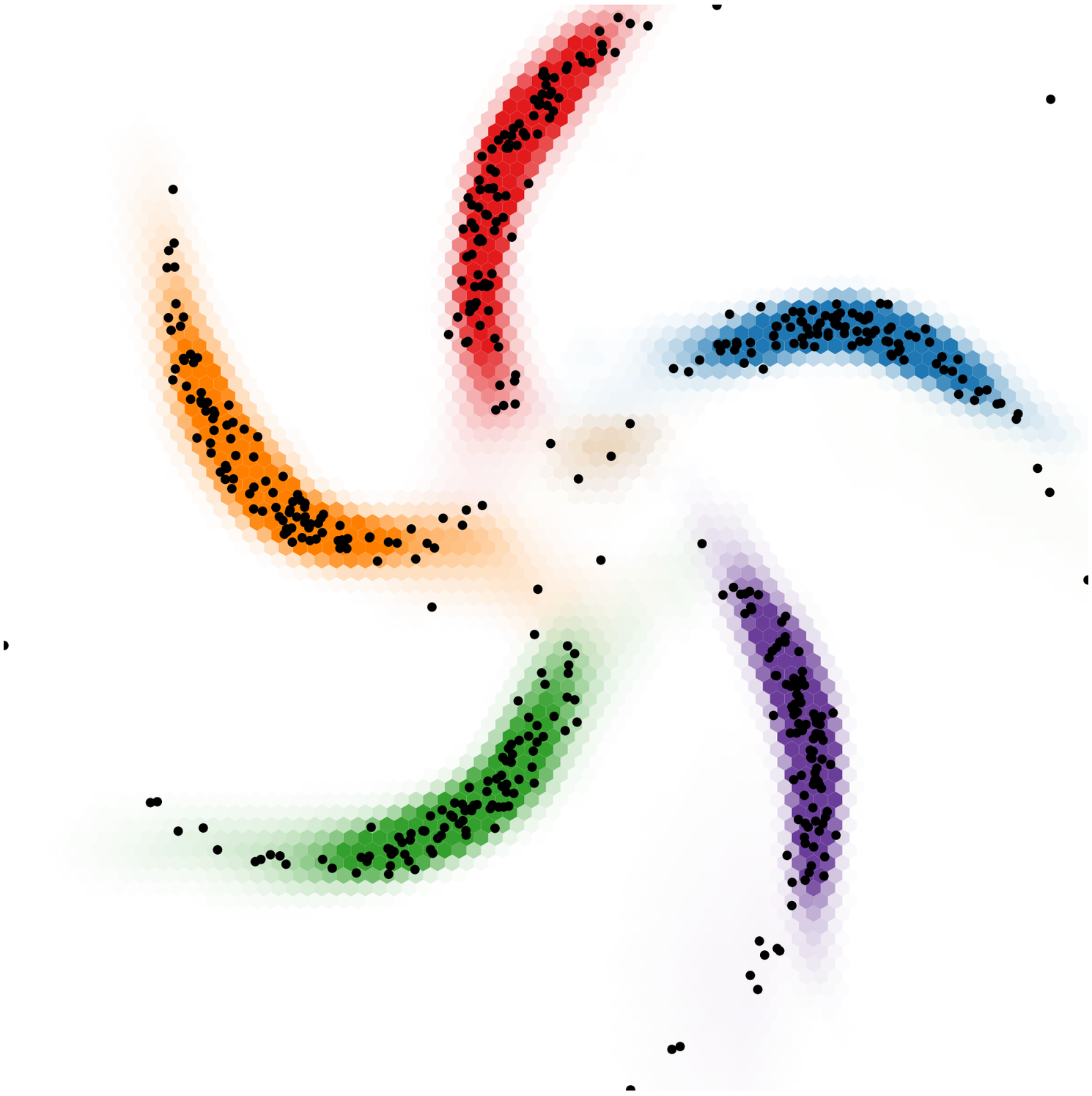
- Approximate intractable distribution by a tractable distribution parameterized by a deep net:

$$p(y|x) = \mathcal{N}(y|\mu = f_\theta(x), \Sigma = g_\theta(x))$$

- Optimize all parameters using stochastic gradient descent

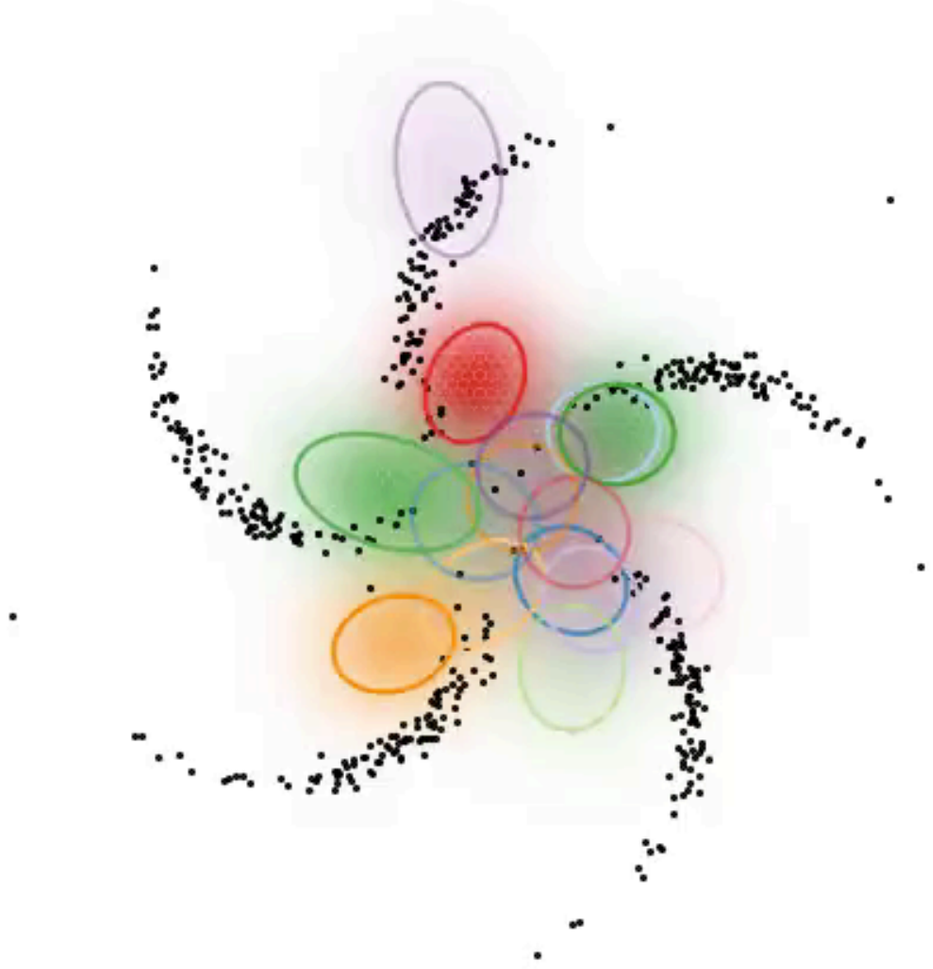




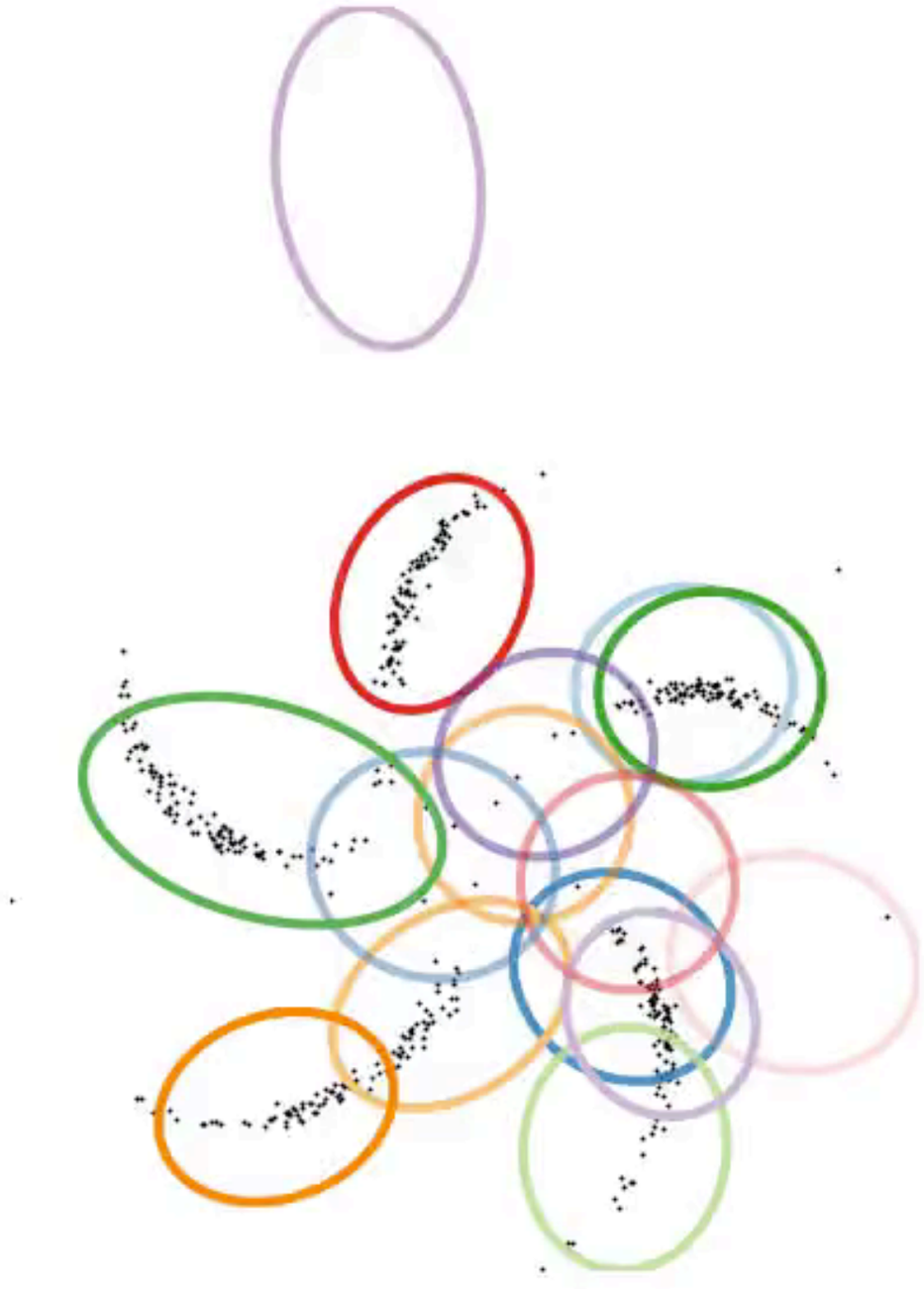


Modeling idea: graphical models on latent variables,
neural network models for observations

Composing graphical models with neural networks for structured representations and fast inference. Johnson, Duvenaud, Wiltschko, Datta, Adams, NIPS 2016

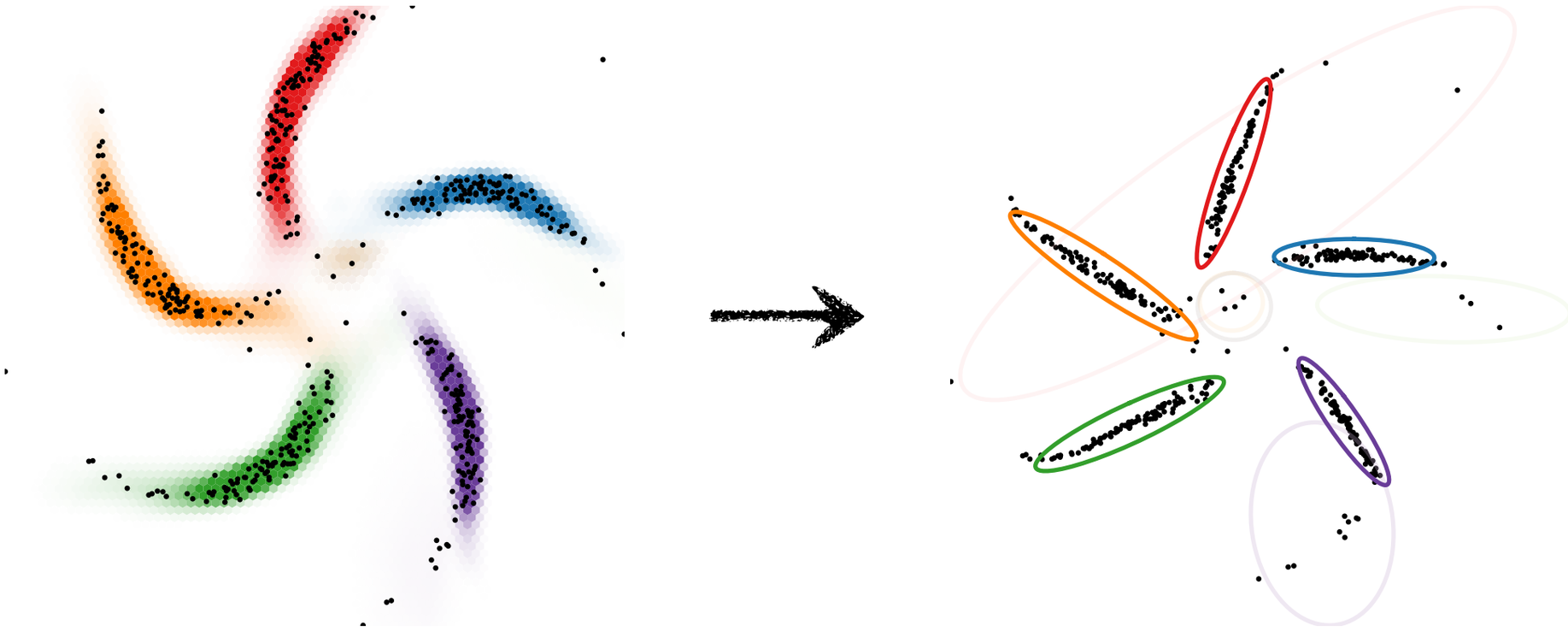


data space

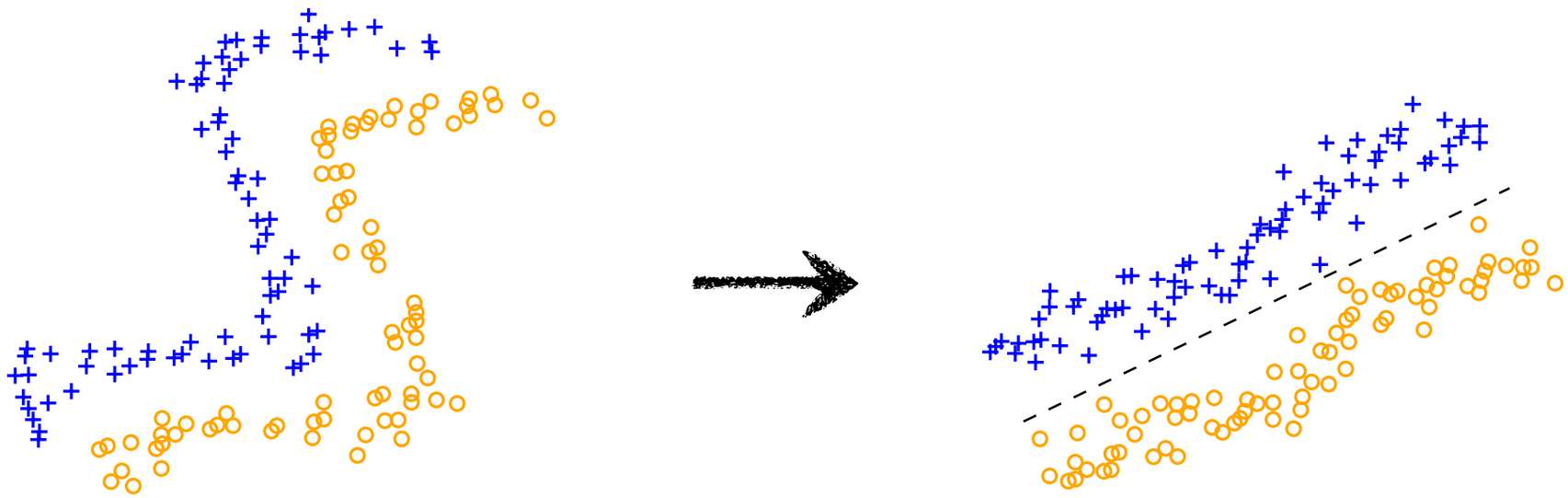


latent space

unsupervised learning



supervised learning



Courtesy of Matthew Johnson

Types of Learning

- **Supervised Learning**: Given input-output pairs (x,y) the goal is to predict correct output y given a new input x.

- **Unsupervised Learning**: Given unlabeled data instances x1, x2, x3… build a model of x, which can be used for making predictions, decisions.

- **Semi-supervised Learning**: We are given only a limited amount of (x,y) pairs, but lots of unlabeled x's.

- All just special cases of estimating distributions from data: p(y|x), p(x), p(x, y).

Image Infill

- Just sampling from p(missing pixels | remaining)
- https://www.youtube.com/watch?v=9V7rNoLVmSs

StyleGAN2

- "Just" a big GAN with some training tricks + data preprocessing.
- Representation ends up being intuitive.
- https://www.youtube.com/watch?v=c-NJtV9Jvp0&feature=youtu.be

Analyzing and Improving the Image Quality of StyleGAN

Tero Karras
NVIDIA

Samuli Laine
NVIDIA

Miika Aittala
NVIDIA

Janne Hellsten
NVIDIA

Jaakko Lehtinen
NVIDIA and Aalto University

Timo Aila
NVIDIA

occluded completions original

Pixel Recurrent Neural Networks (2015)
Aaron van den Oord, Nal Kalchbrenner, Koray Kavukcuoglu

Image to Text

a car is parked in
the middle of nowhere .

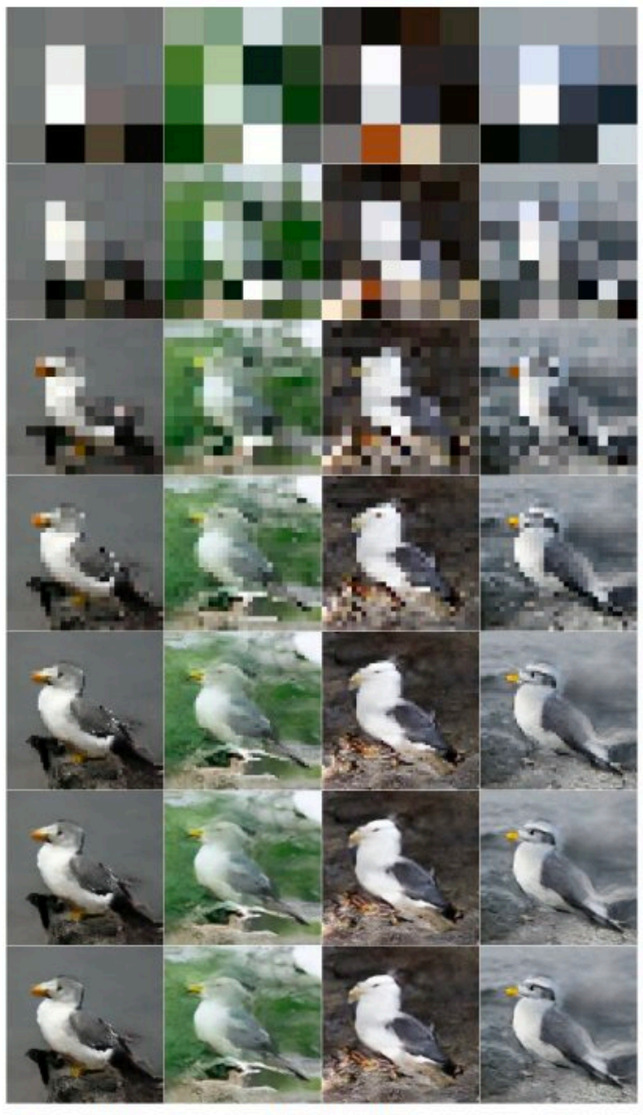
a wooden table and chairs
arranged in a room .

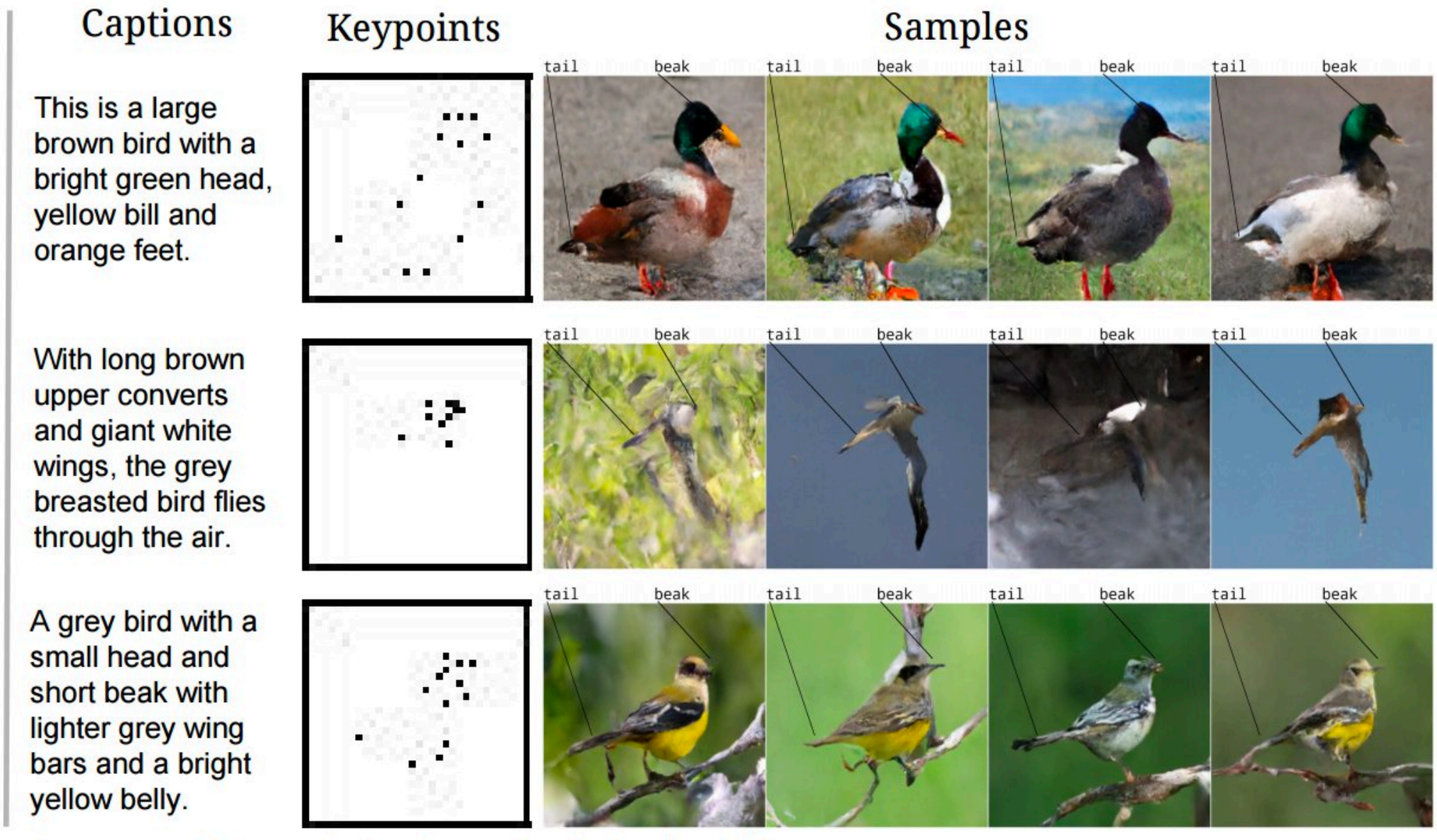
there is a cat sitting on a shelf .

a ferry boat on a marina
with a group of people .

a little boy with a bunch
of friends on the street .

Text to Image





A white large bird with orange legs and gray secondaries and primaries, and a short yellow bill.

- Parallel Multiscale Autoregressive Density Estimation. Reed et al., 2017

Sequential Data: Video

- Stochastic Video Generation with a Learned Prior. Emily Denton, Rob Fergus

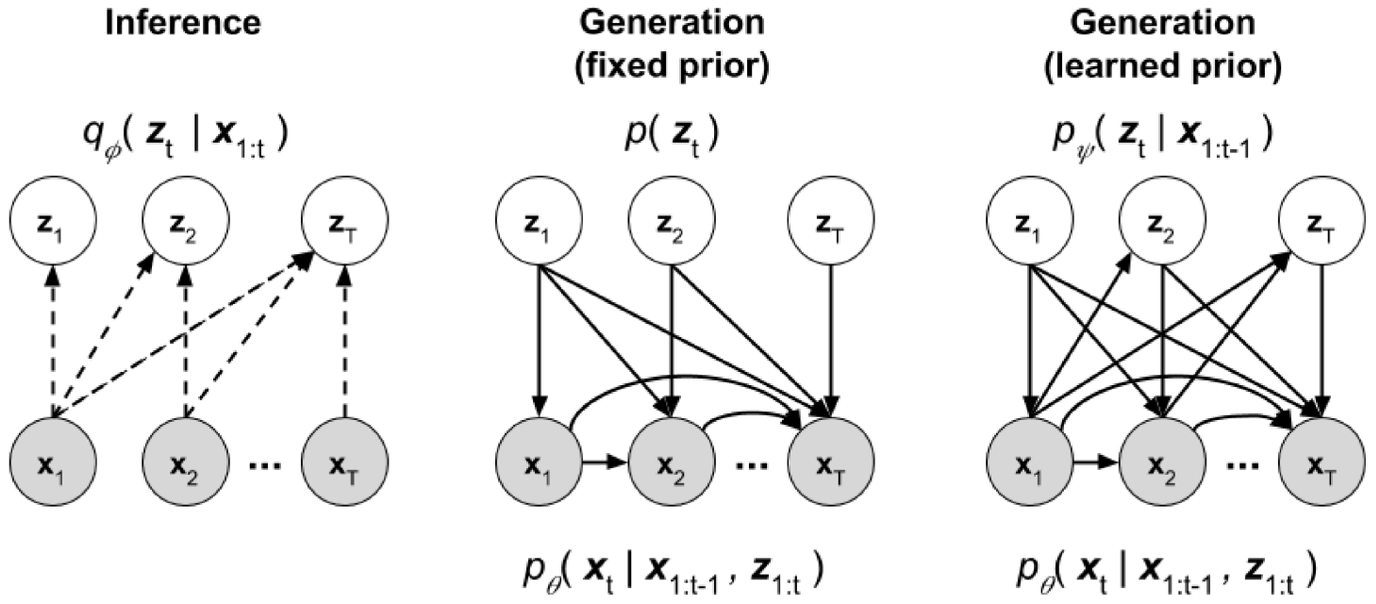
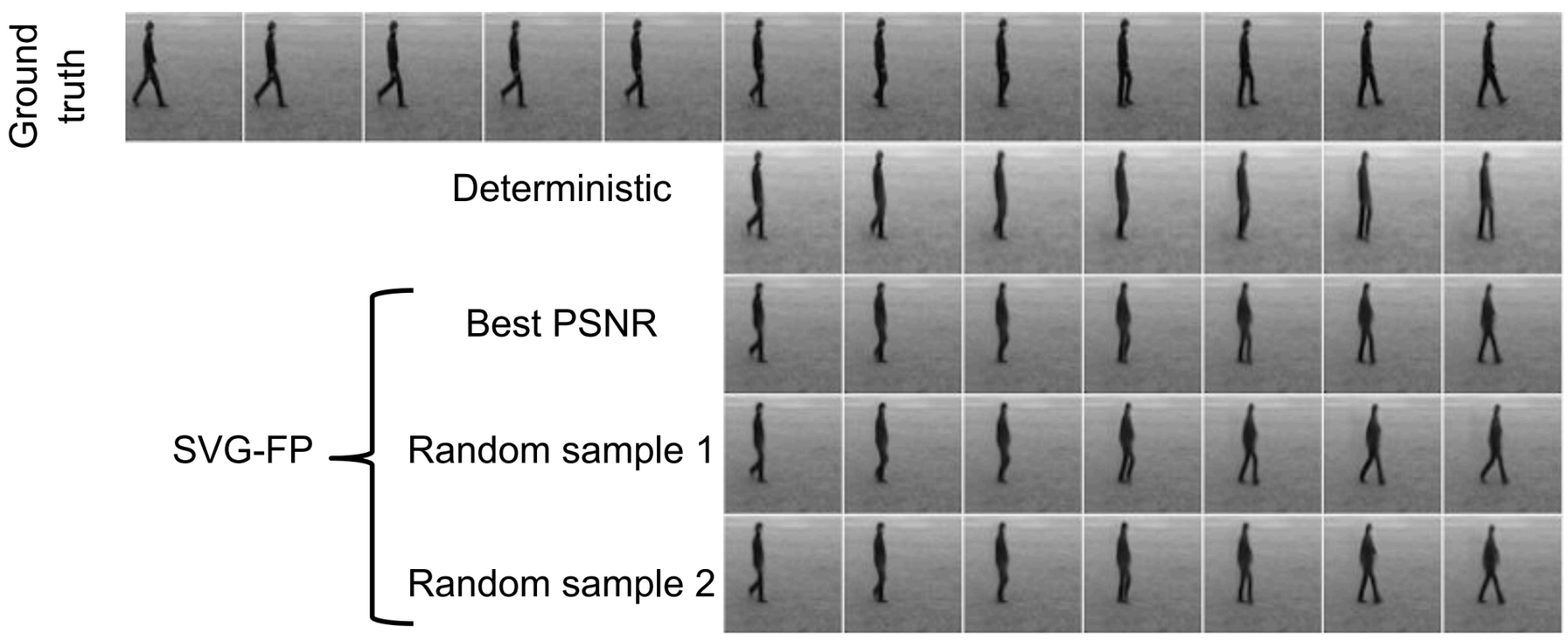


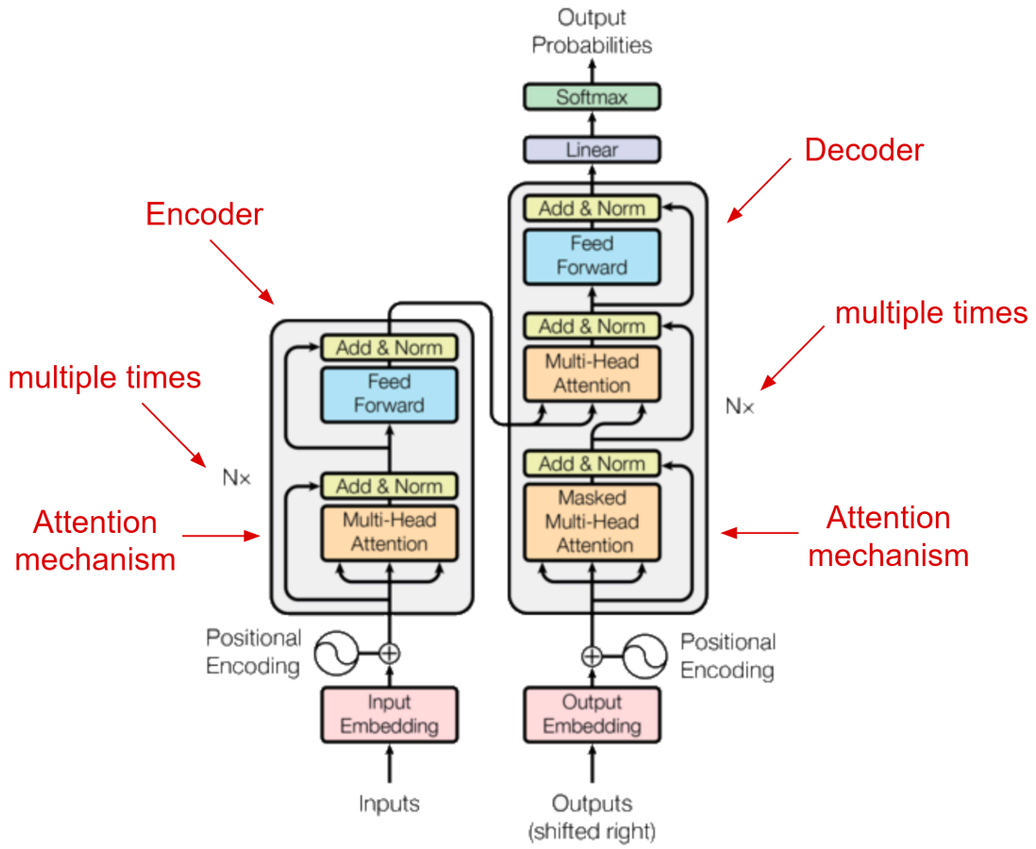
Figure 1. Inference (left) and generation in the SVG-FP (middle) and SVG-LP models (right).

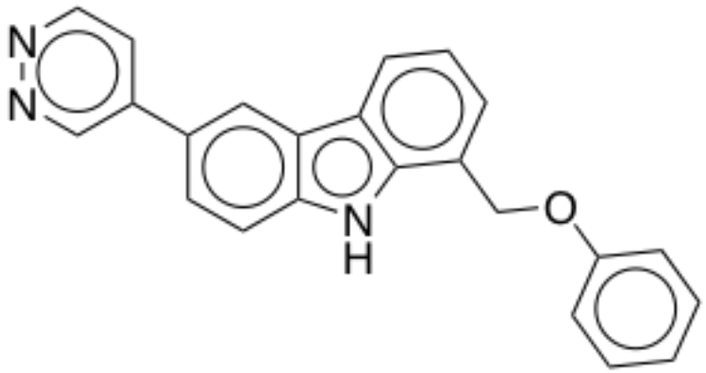


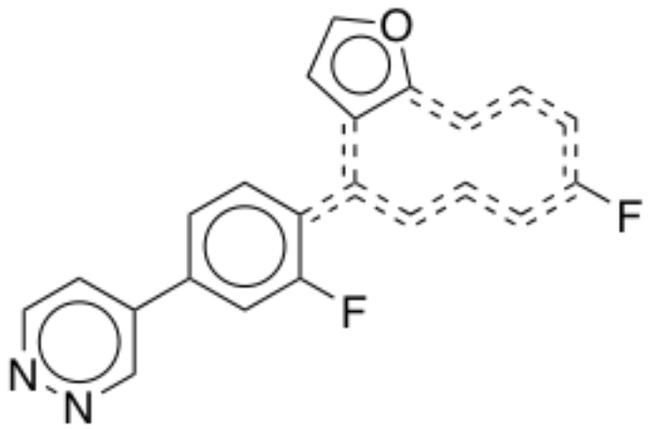
Sequential Data: Text

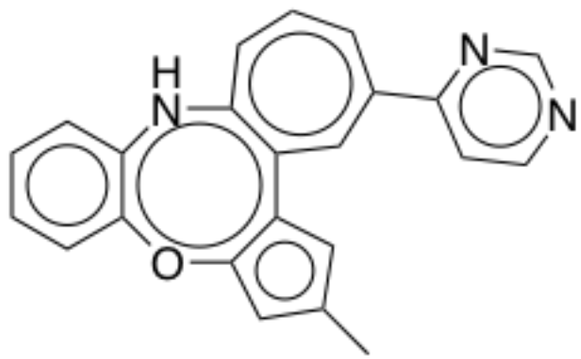
- Attention Is All You Need’ Vaswani et al., 2017

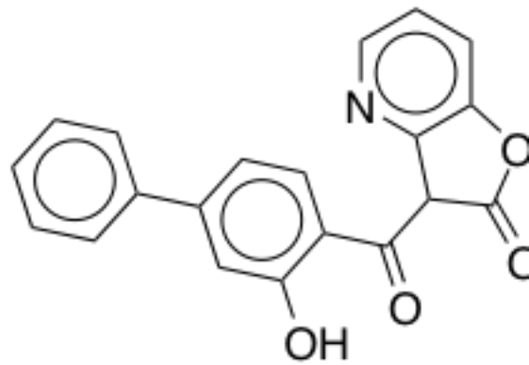
- Variant of RNNs with attention, aka key-query layers

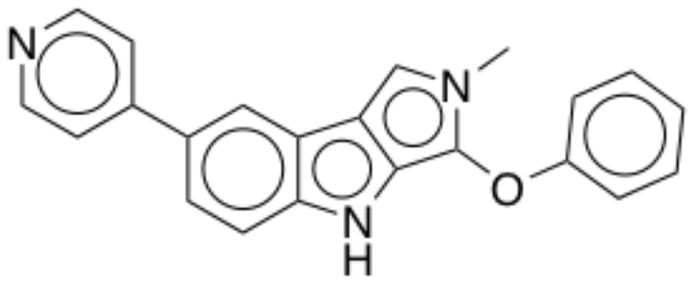


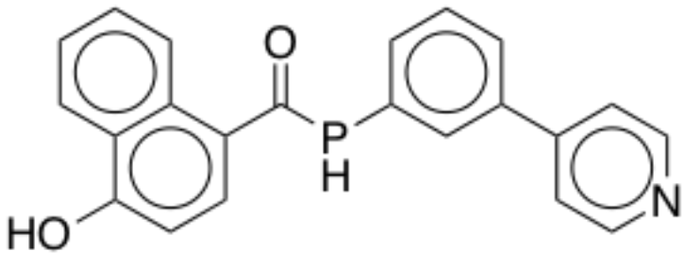


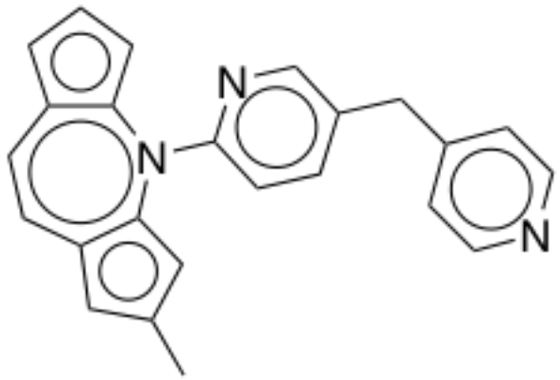


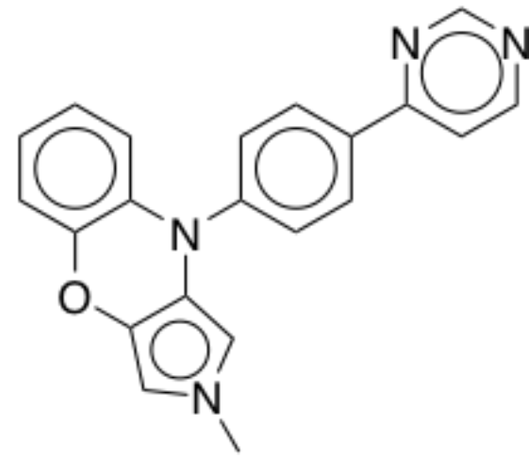


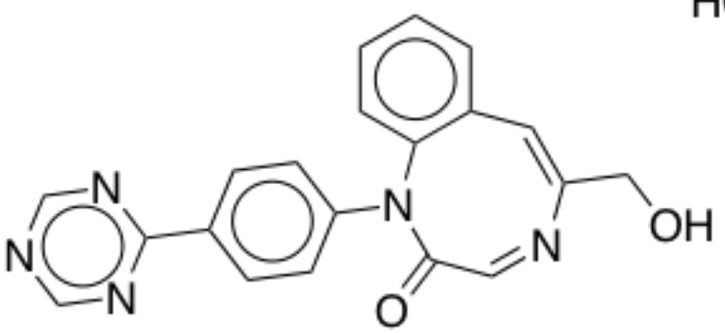


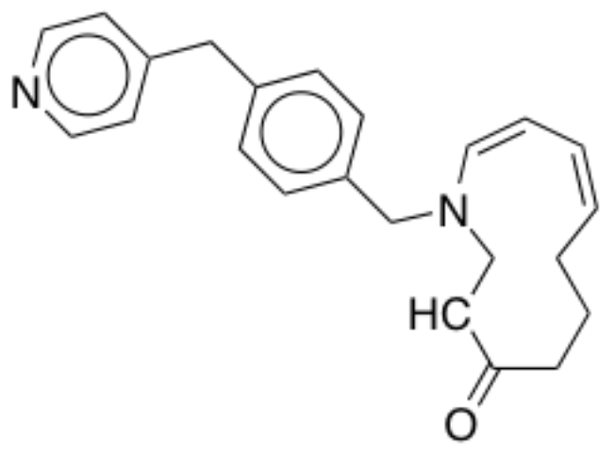


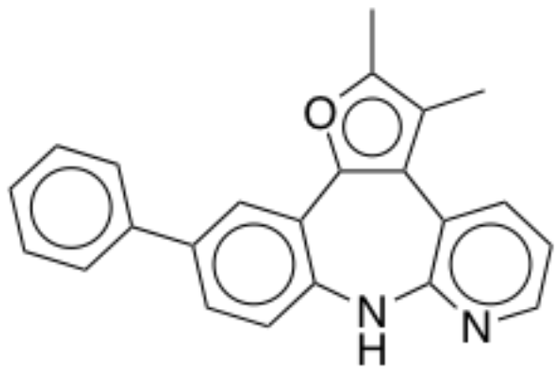


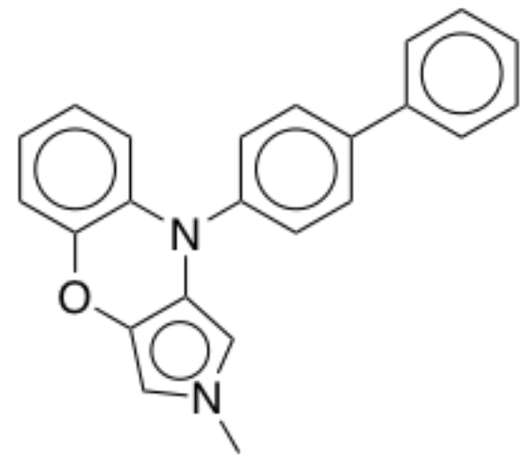


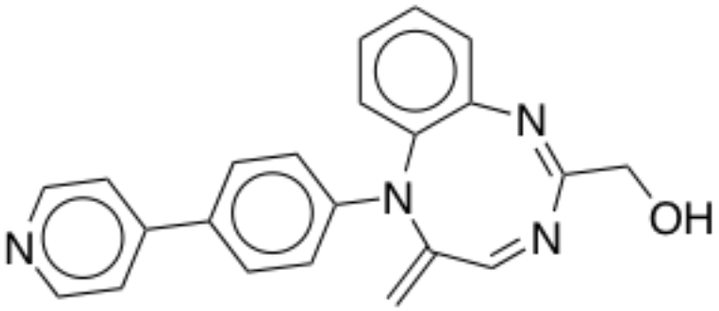


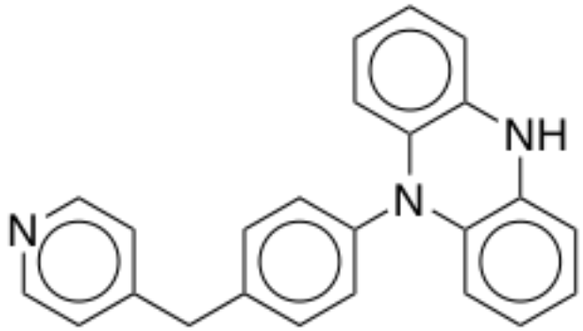


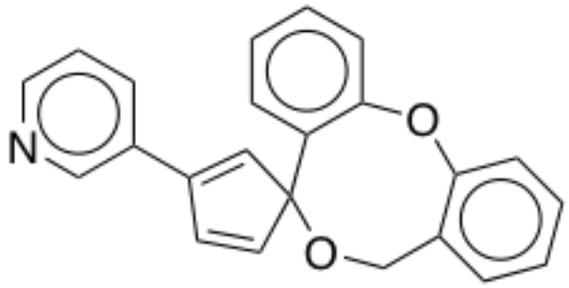


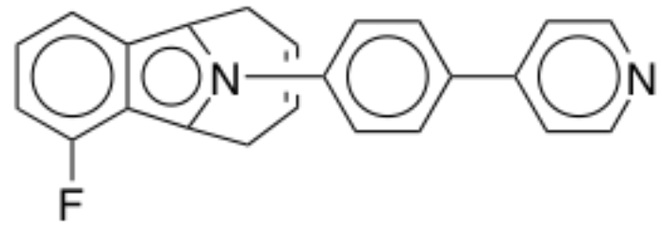


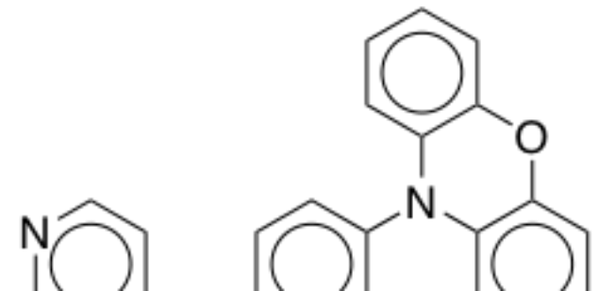


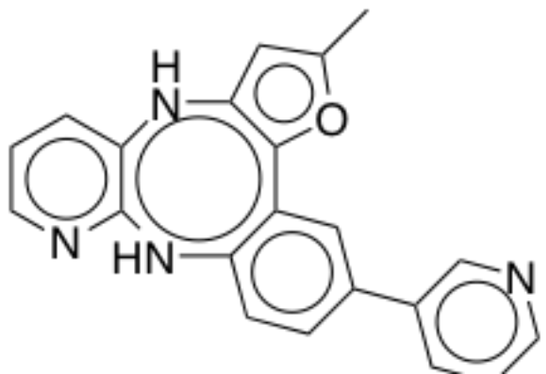


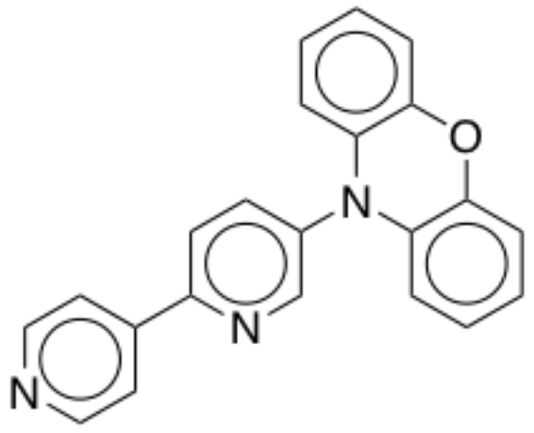


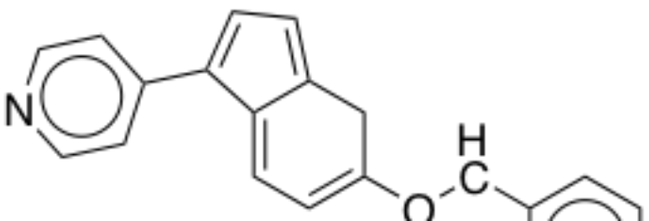


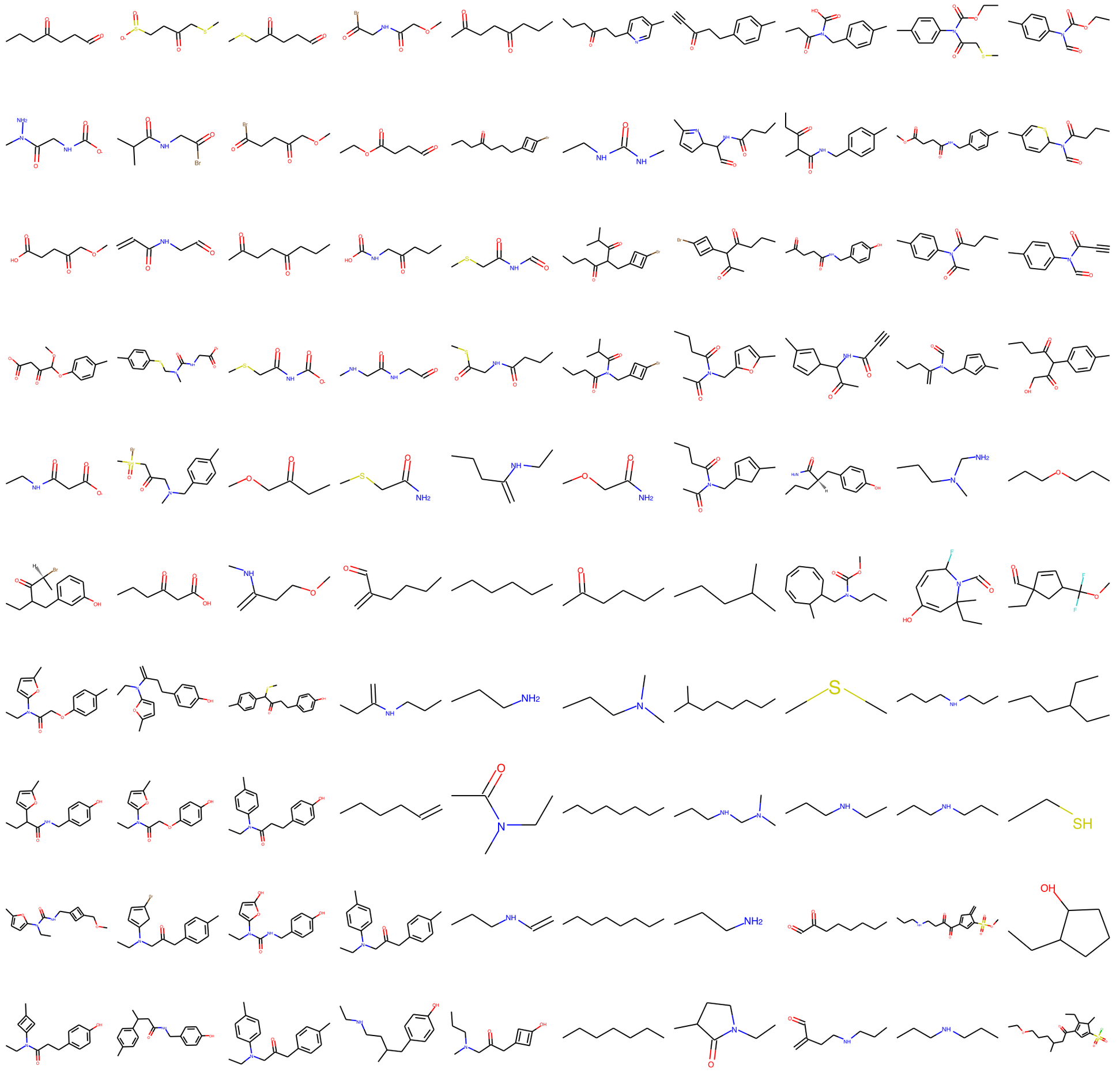










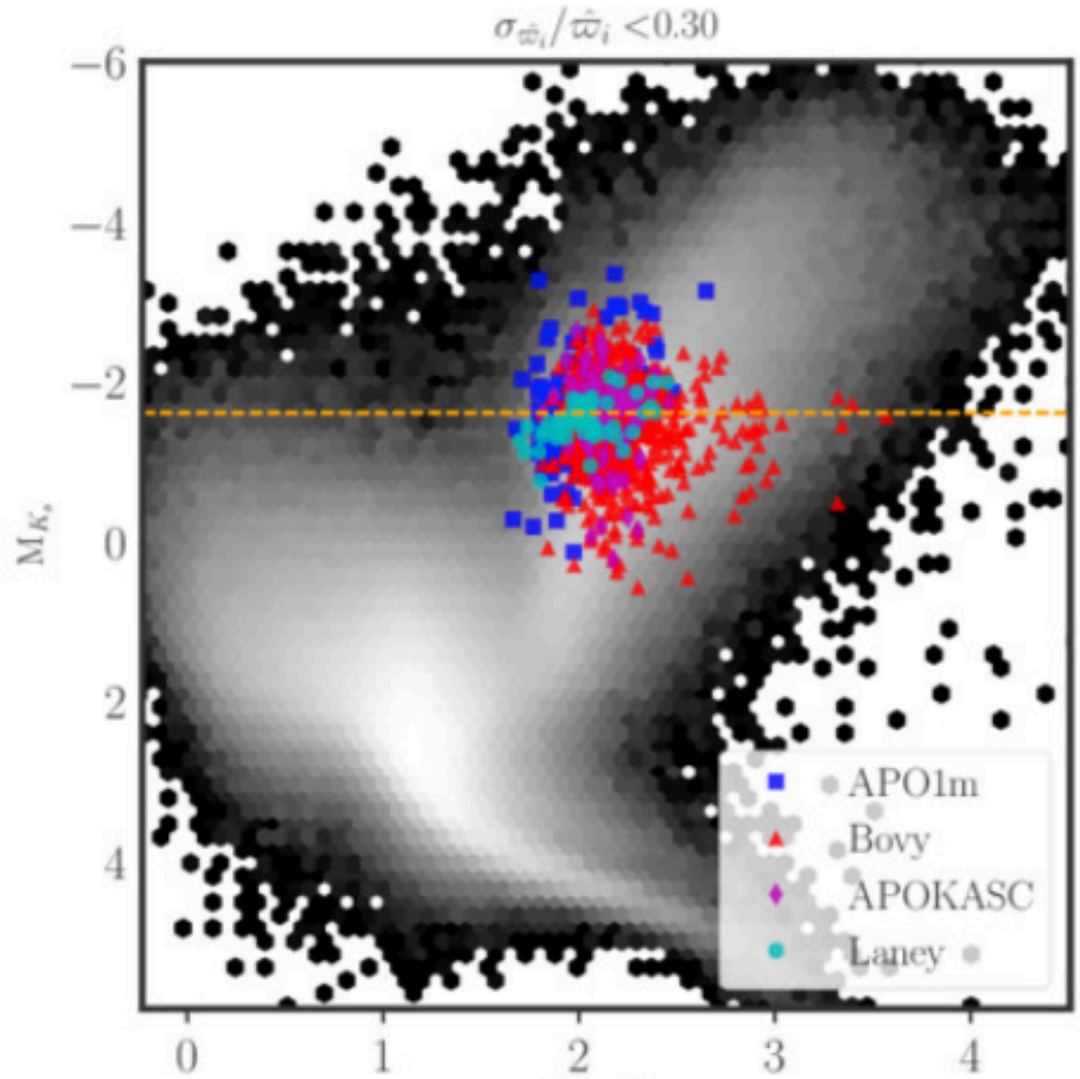


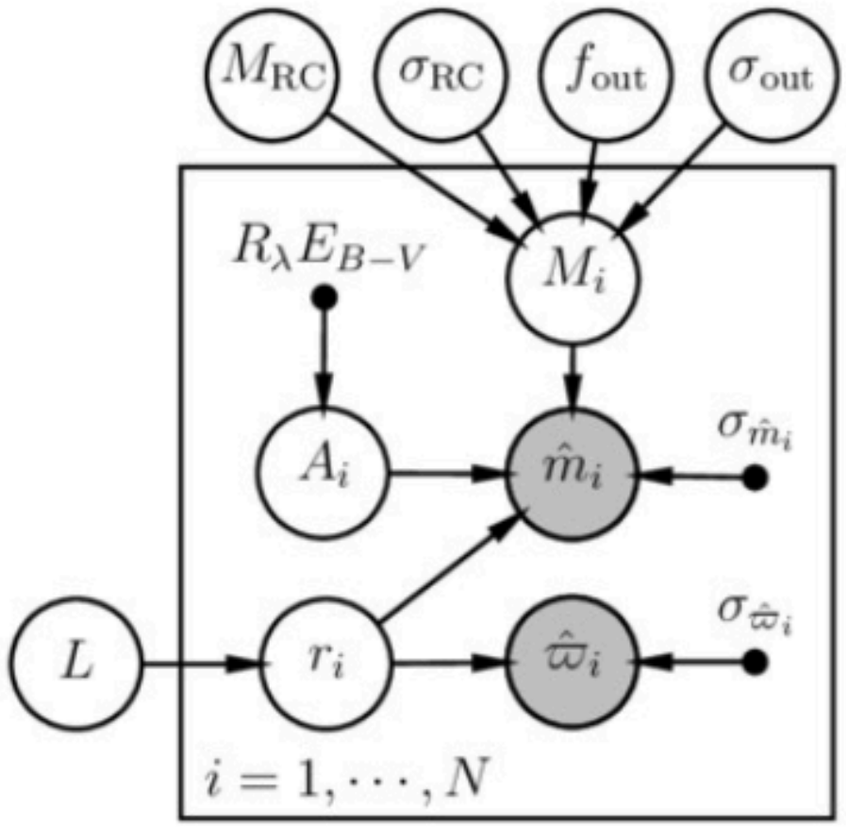
Scientific Data

- Need to marginalize over all the things we don't know

Modeling the photometric red clump

Use existing spectroscopic or astroseismic RC catalogs
Hierarchical probabilistic model: Gaussian for the RC + outliers, marginalizing over dust, parallaxes, observed magnitudes.

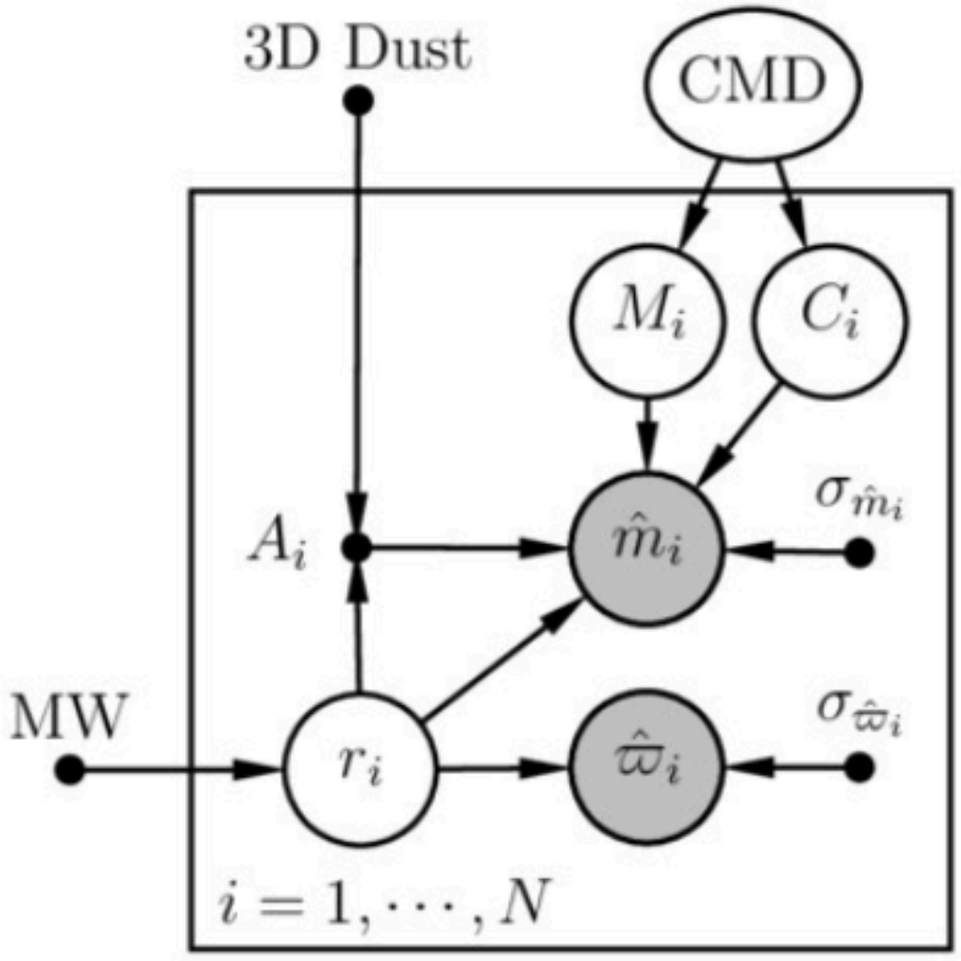




Model+MCMC with stan
Sample joint posterior

https://speakerdeck.com/ixkael/data-driven-models-of-the-milky-way-in-the-gaia-era

TGAS PGM and model



Absolute magnitude:

$$M_V = m_V - 5\log_{10}(\frac{d}{10\ \mathrm{pc}})$$

Parallax & magnitude likelihoods:

$$p(\hat{\varpi}|d, \sigma_\varpi) = \mathcal{N}(\hat{\varpi} - 1/d; \sigma^2_\varpi)$$

$$p(\hat{\vec{m}}|d, \vec{C}, M, \Sigma_{\hat{\vec{m}}})$$
$$= \mathcal{N}\left(\hat{\vec{m}} - \vec{m}(d, \vec{C}, M); \Sigma_{\hat{\vec{m}}}\right)$$

Posterior distribution now tying all objects together, with CMD

https://speakerdeck.com/ixkael/data-driven-models-of-the-milky-way-in-the-gaia-era

Advantages of probabilistic latent-variable models

- **Data-efficient** - automatic regularization, can take advantage of more information
- **Composeable** - e.g. incorporate data corruption model.
- **Handle missing or corrupted data** - no imputation, always integration.
- **Predictive uncertainty** - useful for decision-making.
- **Conditional predictions** - e.g. if brexit happens, the value of the pound will fall
- **Active learning** - What data would be expected to increase our confidence about a prediction?

- **Disadvantages**:
  - intractable integral over latent variables

Reasons to take this course

- Any sort of 'data scientist' job.
- Getting into research in ML. (but it's a gold rush)
- Doing research in another area, but being able to build / tweak / question models. (recommended)
- Not being impressed by "it was done with deep learning / reinforcement learning / AI"

Syllabus

- Course Content
- Collaboration policy
- Communication / extension policy

Discourse

- Piazza sucks
- Unified across CSC412 and STA414
- TAs will monitor, but please answer each other! Great thing for us to mention letters of rec
- Don't share solutions though
- Should get email invite. If auditing, email instructors for link

Emails

- Don't email us directly except for personal logistics
- Instructors email: sta414prof@cs.toronto.edu
- TA email: sta414tas@cs.toronto.edu

Learning Outcomes: Today

- Know what topics are and aren't in the course.
- An idea of if you have the background + how hard the material will be.
- What you should be able to do with this knowledge.
- Know how to set up a computing environment (next)

Sotware Tutorial