CSC412/2506 Probabilistic Learning and Reasoning

Introduction

Today

- Course information
- Overview of ML with examples
- Ungraded, anonymous background quiz

- **Thursday**: Basics of ML vocabulary (cross-validation, objective functions, overfitting, regularization) and basics of probability manipulation

Course Website

- www.cs.toronto.edu/~duvenaud/courses/csc412
- Contains all course information, slides, etc.

Evaluation

- Assignment 1: due Feb 10th worth 15%
- Assignment 2: due March 3rd worth 15%
- Assignment 3: due March 24th worth 20%
- 1-hour Midterm: Feb 23rd worth 20%
- Project: due April 10th worth 30%
- 15% per day of lateness, up to 4 days

Related Courses

- CSC411: List of methods, (K-NN, Decision trees), more focus on computation
- STA302: Linear regression and classical stats
- ECE521: Similar material, more focus on computation
- STA414: Mostly same material, slightly more introductory, more emphasis on theory than coding, exam instead of project
- CSC321: Neural networks - about 30% overlap

Textbooks + Resources

- No required textbook
- Christopher M. Bishop (2006) *Pattern Recognition and Machine Learning.*
- Kevin Murphy (2012), *Machine Learning: A Probabilistic Perspective*.
- Trevor Hastie, Robert Tibshirani, Jerome Friedman (2009) *The Elements of Statistical Learning*
- David MacKay (2003) *Information Theory, Inference, and Learning Algorithms*
- Deep Learning (2016) Goodfellow, Bengio, Courville

Stats vs Machine Learning

- Statistician: Look at the data, consider the problem, and design a model we can understand
    - Analyze methods to give guarantees
    - Want to make few assumptions
- ML: We only care about making good predictions!
    - Let’s make a general procedure that works for lots of datasets
    - No way around making assumptions, let’s just make the model large enough to hopefully include something close to the truth
    - Can’t use bounds in practice, so evaluate empirically to choose model details
    - Sometimes end up with interpretable models anyways

Types of Learning

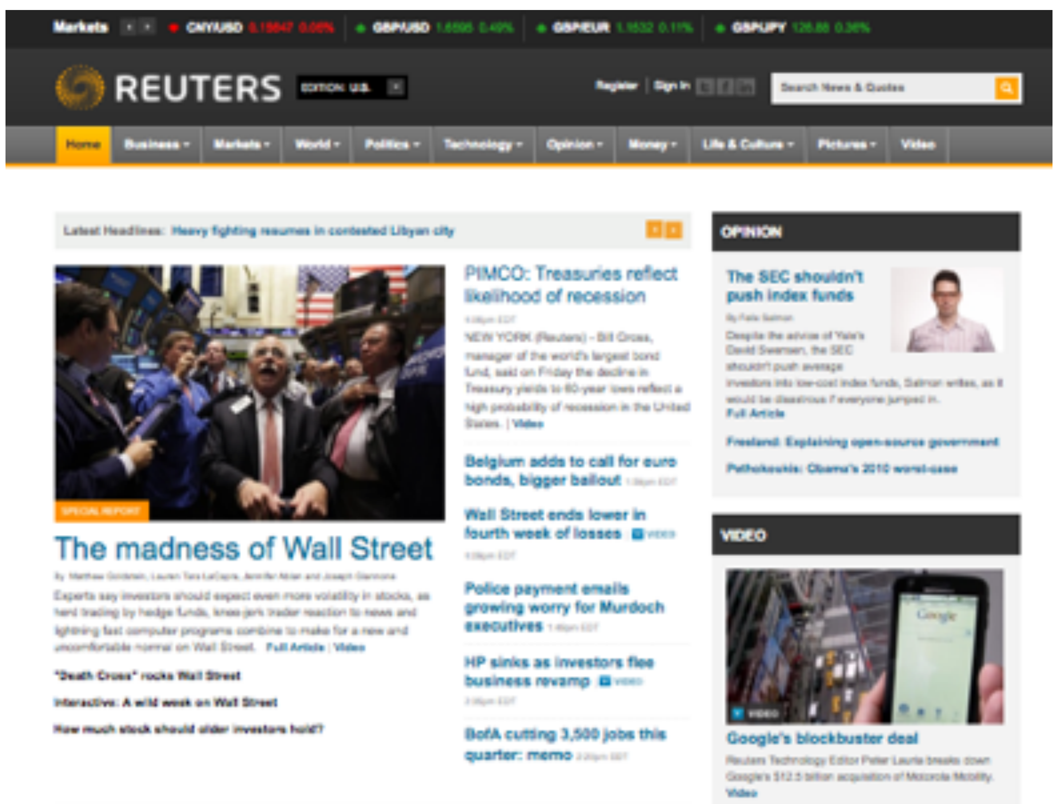
- **Supervised Learning**: Given input-output pairs (x,y) the goal is to predict correct output given a new input.
- **Unsupervised Learning**: Given unlabeled data instances x1, x2, x3… build a statistical model of x, which can be used for making predictions, decisions.
- **Semi-supervised Learning**: 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).
- **Active learning and RL**: Also get to choose actions that influence future information + reward. Can just use basic decision theory.

Finding Structure in Data

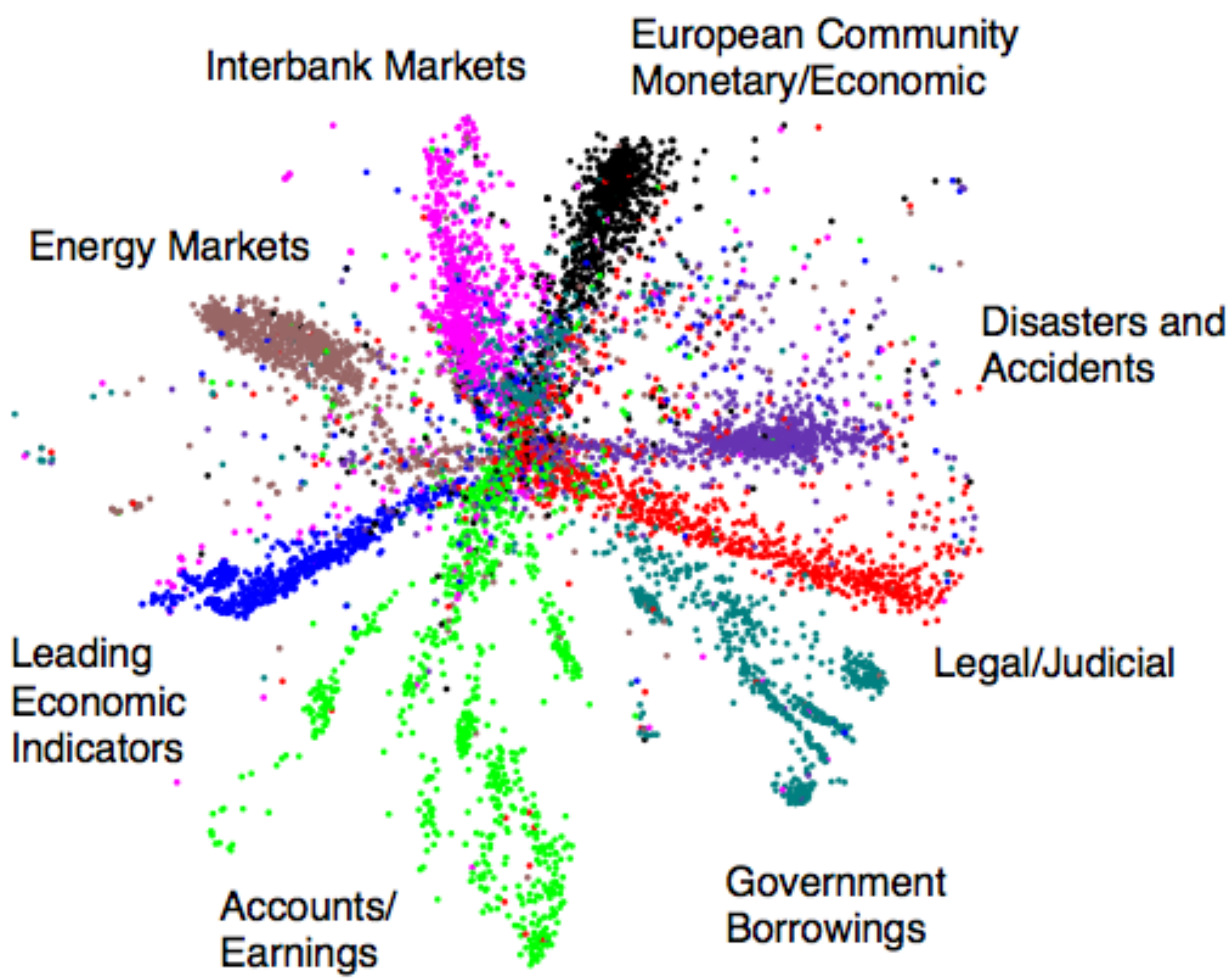
$$P(\mathbf{x}) = \frac{1}{\mathcal{Z}} \sum_{\mathbf{h}} \exp\left[\mathbf{x}^\top \mathbf{W} \mathbf{h}\right]$$

Vector of word counts on a webpage

Latent variables: hidden topics



804,414 newswire stories



Matrix Factorization

Collaborative Filtering/
Matrix Factorization/

Hierarchical Bayesian Model

Rating value of user i for item j

Latent user feature (preference) vector

Latent item feature vector

$$r_{ij}|\mathbf{u}_i, \mathbf{v}_j, \sigma \sim \mathcal{N}(\mathbf{u}_i^\top \mathbf{v}_j, \sigma^2),$$

$$\left.\begin{aligned} \mathbf{u}_i|\sigma_u &\sim \mathcal{N}(\mathbf{0}, \sigma_u^2\mathbf{I}), \quad i = 1, ..., N. \\ \mathbf{v}_j|\sigma_v &\sim \mathcal{N}(\mathbf{0}, \sigma_v^2\mathbf{I}), \quad j = 1, ..., M. \end{aligned}\right\}$$

Latent variables that we infer from observed ratings.

Prediction: predict a rating r^*_{ij} for user i and query movie j.

$$P(r^*_{ij}|\mathbf{R}) = \iint P(r^*_{ij}|\mathbf{u}_i, \mathbf{v}_j)\underbrace{P(\mathbf{u}_i, \mathbf{v}_j|\mathbf{R})}_{\textbf{Posterior over Latent Variables}} d\mathbf{u}_i d\mathbf{v}_j$$

Infer latent variables and make predictions using Bayesian inference (MCMC or SVI).

Finding Structure in Data

Collaborative Filtering/
Matrix Factorization/
Product Recommendation

Learned ``genre''

Netflix dataset:
480,189 users
17,770 movies
Over 100 million ratings.

Fahrenheit 9/11
Bowling for Columbine
The People vs. Larry Flynt Canadian
Bacon
La Dolce Vita

Independence Day
The Day After Tomorrow
Con Air
Men in Black II
Men in Black

Friday the 13th
The Texas Chainsaw Massacre
Children of the Corn
Child's Play
The Return of Michael Myers

- Part of the wining solution in the Netflix contest (1 million dollar prize).

Impact of Deep Learning

- Speech Recognition
- Computer Vision
- Recommender Systems
- Language Understanding
- Drug Discovery and Medical Image Analysis

Multimodal Data

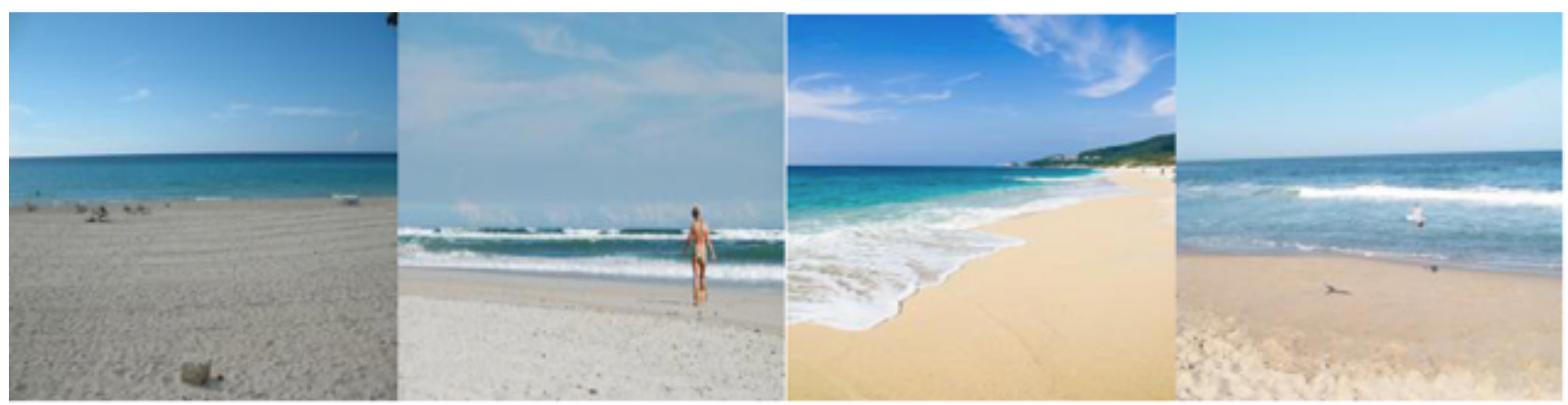
mosque, tower, building, cathedral, dome, castle

ski, skiing, skiers, skiiers, snowmobile

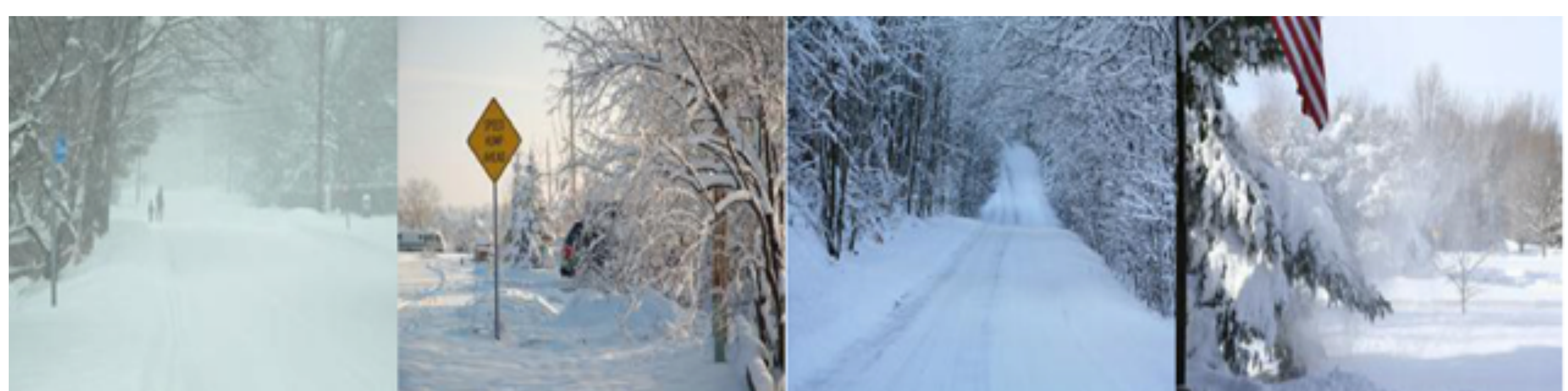
kitchen, stove, oven, refrigerator, microwave

bowl, cup, soup, cups, coffee

beach



snow



Caption Generation

LZ

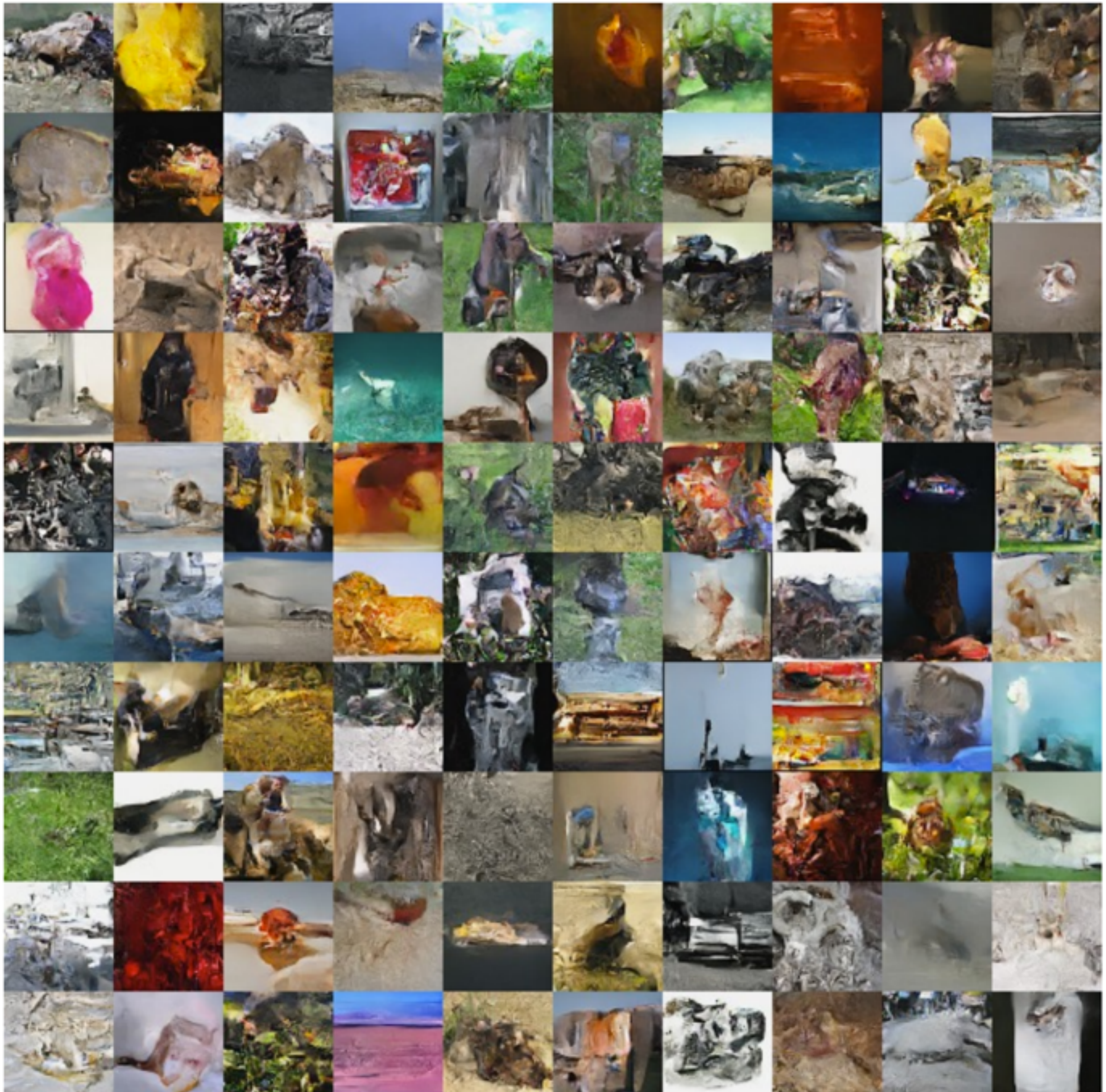
a car is parked in
the middle of nowhere .

a wooden table and chairs
arranged in a room .

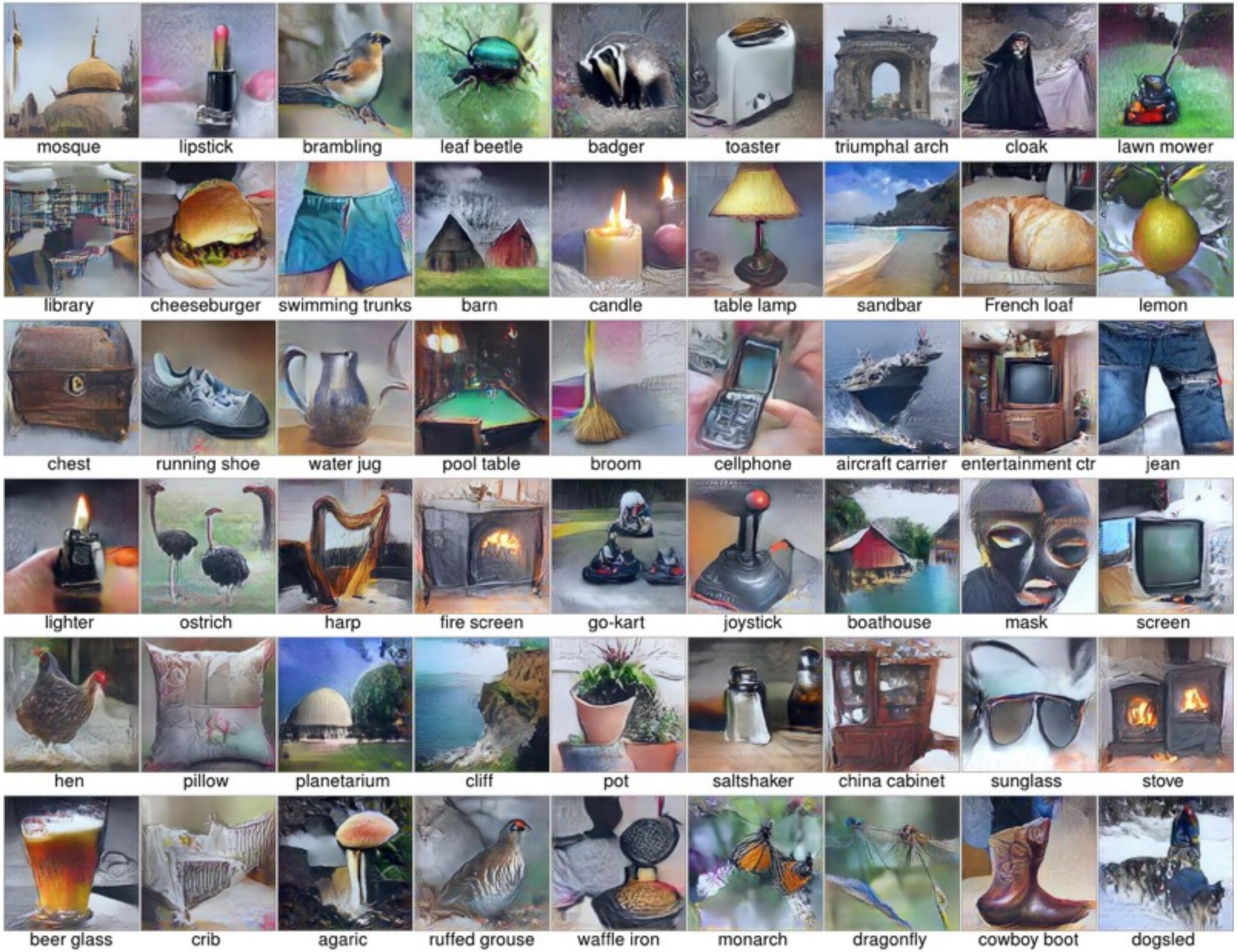
there is a cat sitting on a shelf .

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

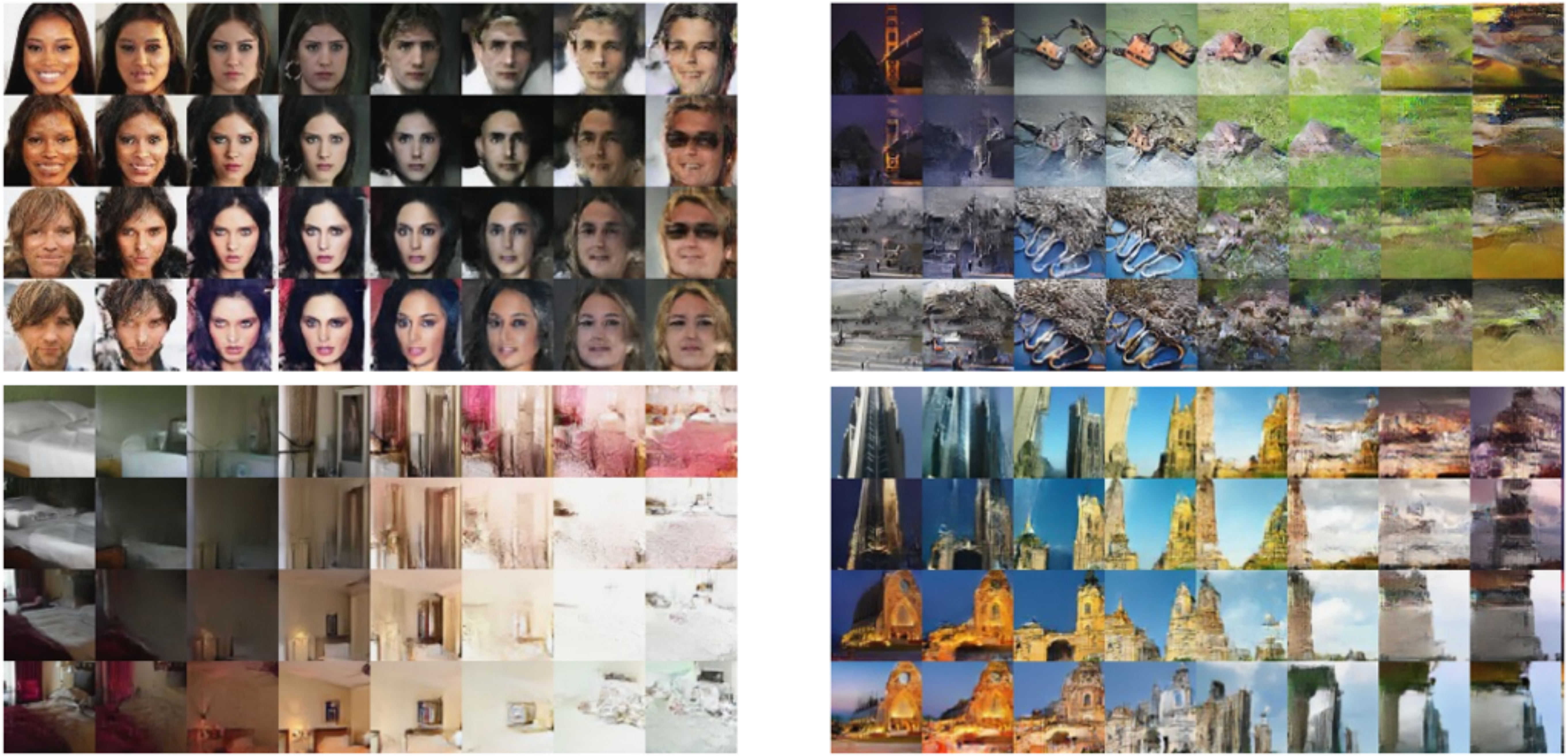
a little boy with a bunch
of friends on the street .



Density estimation using Real NVP. Ding et al, 2016



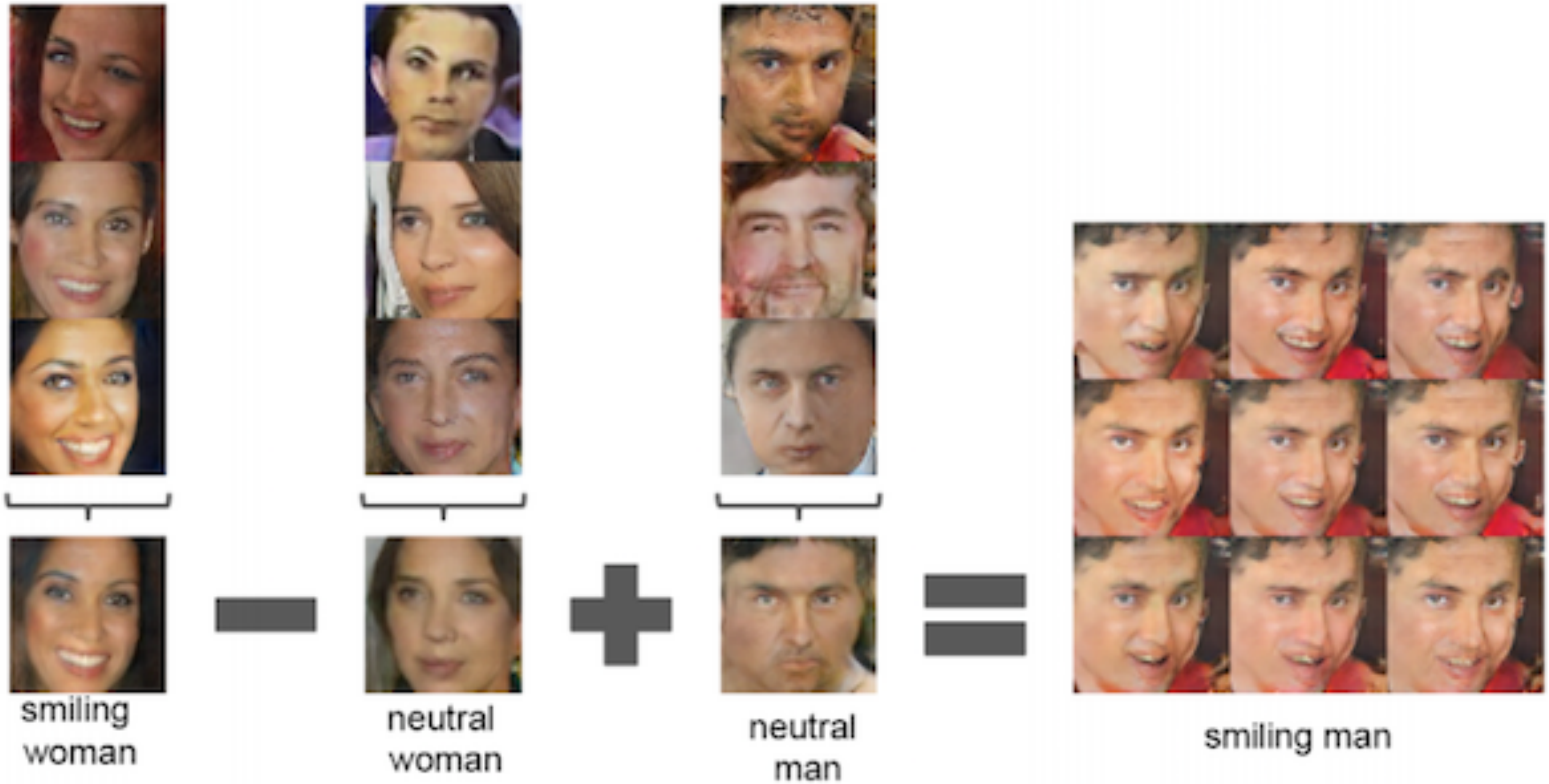
Nguyen A, Dosovitskiy A, Yosinski J, Brox T, Clune J (2016). *Synthesizing the preferred inputs for neurons in neural networks via deep generator networks.* Advances in Neural Information Processing Systems 29



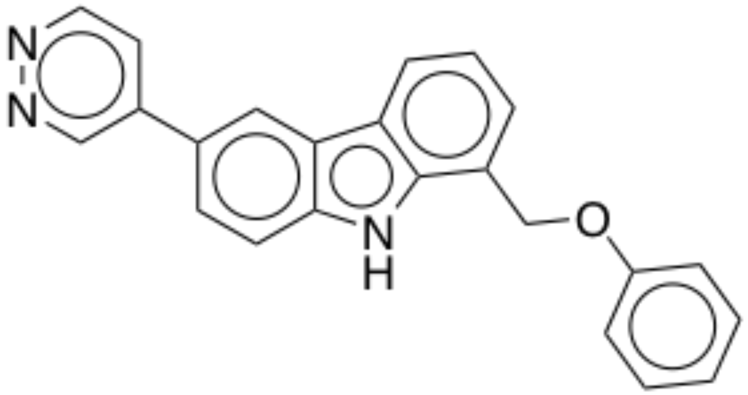
Density estimation using Real NVP. Ding et al, 2016

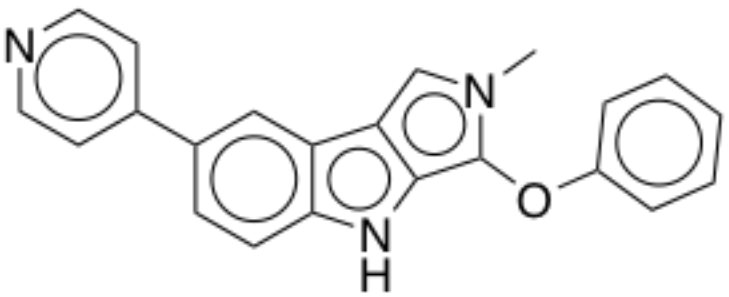
Pixel Recurrent Neural Networks
Aaron van den Oord, Nal Kalchbrenner, Koray Kavukcuoglu

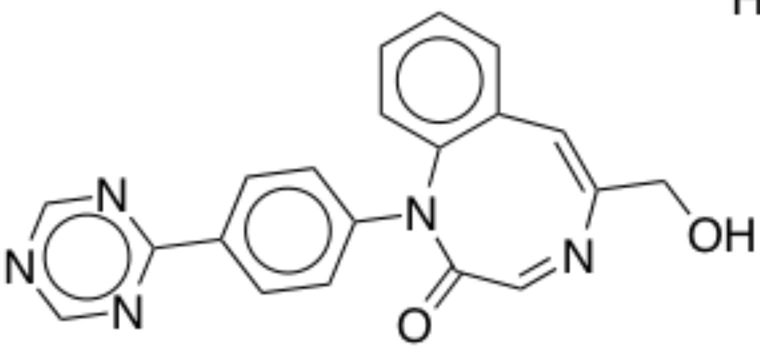
Density estimation using Real NVP. Ding et al, 2016

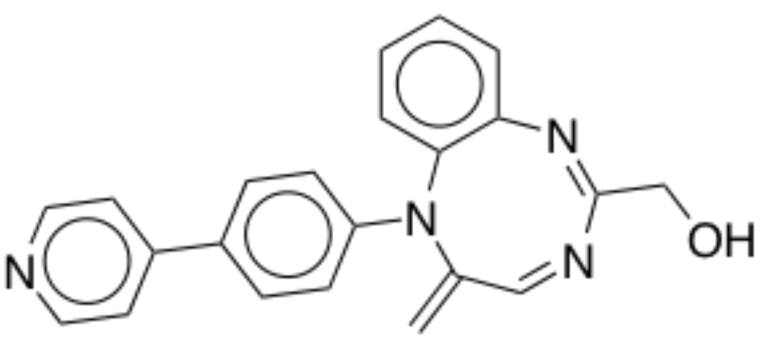


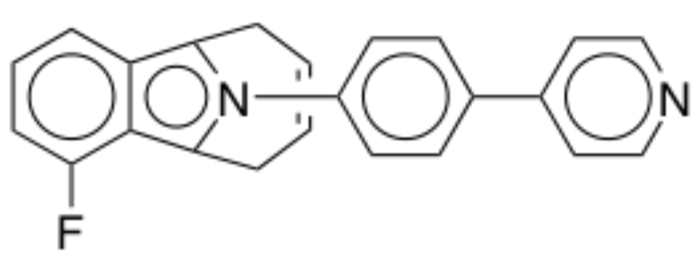
Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks Alec Radford, Luke Metz, Soumith Chintala

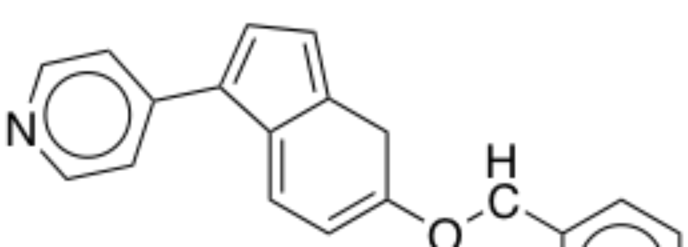


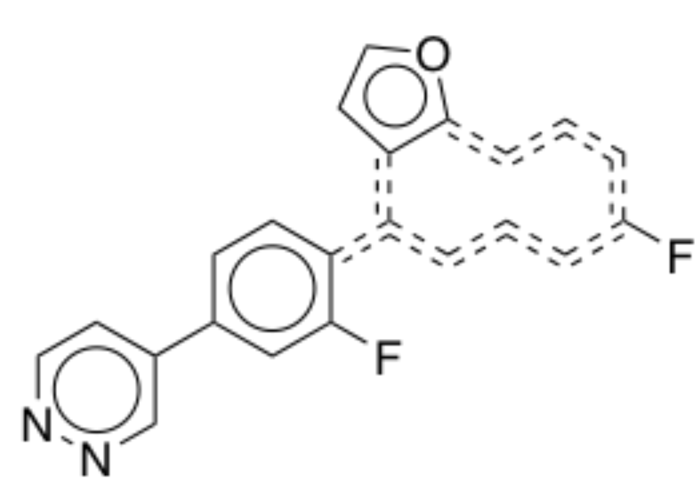


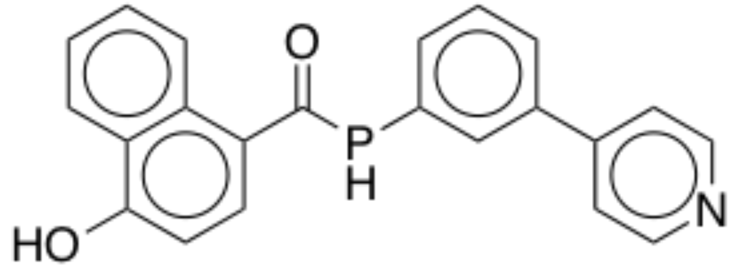


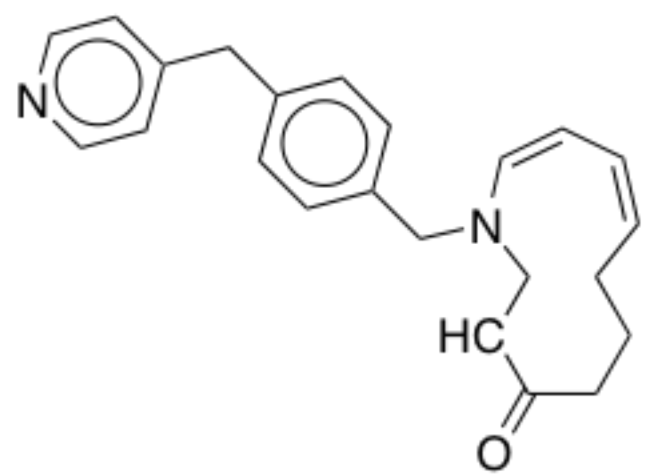


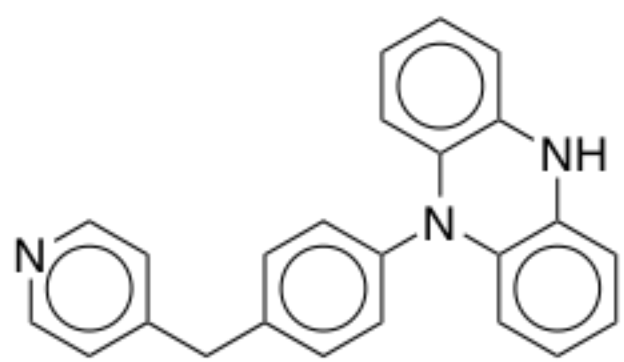


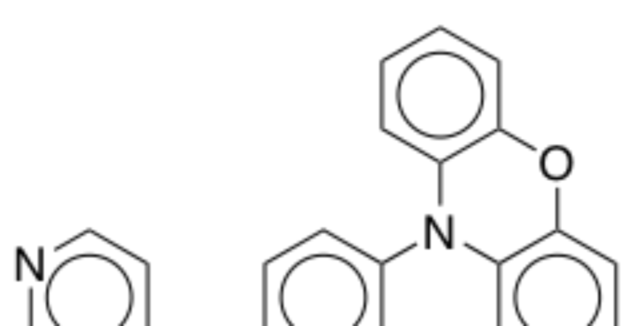


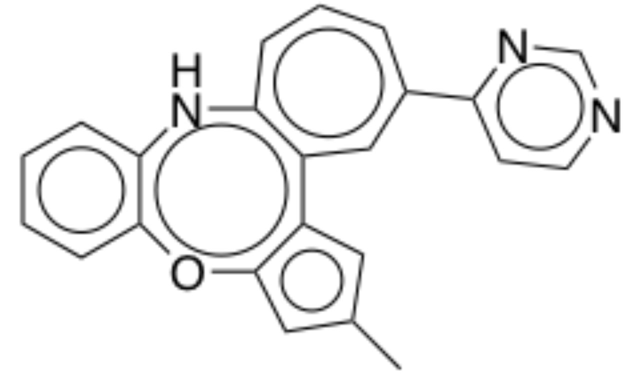


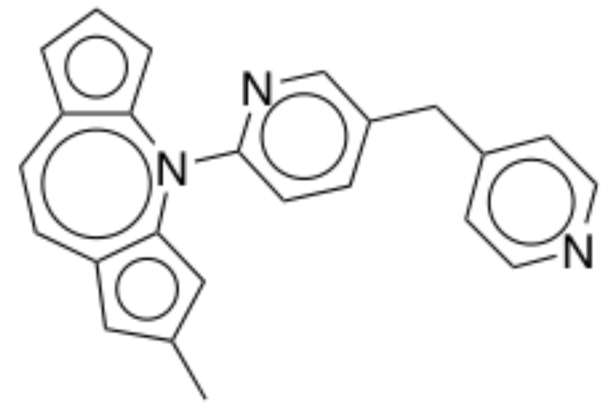


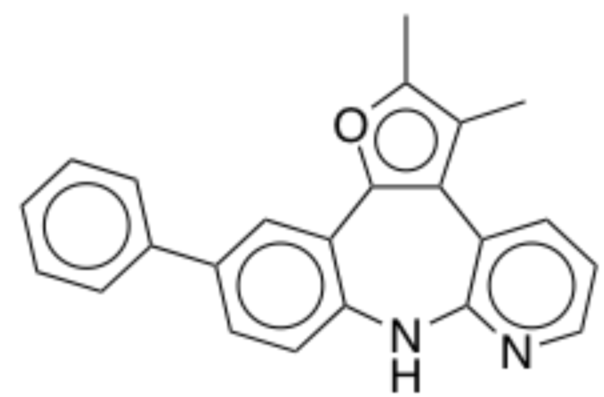


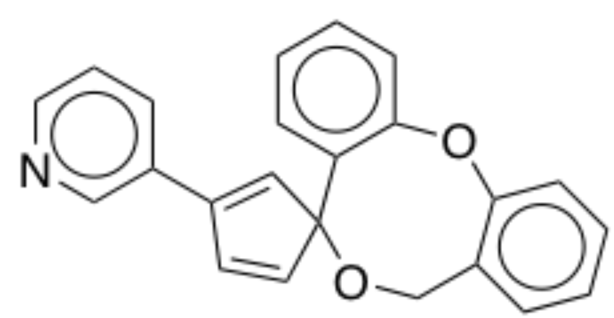


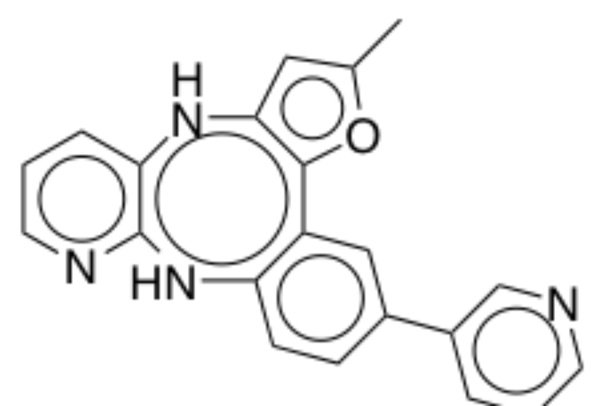


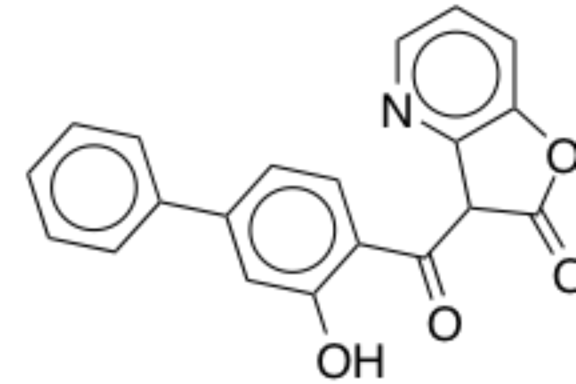


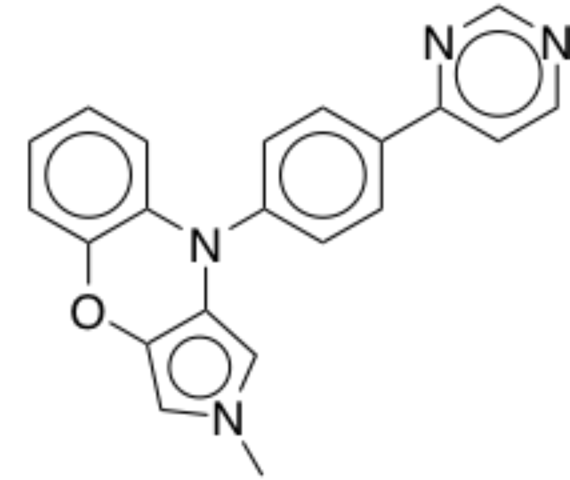


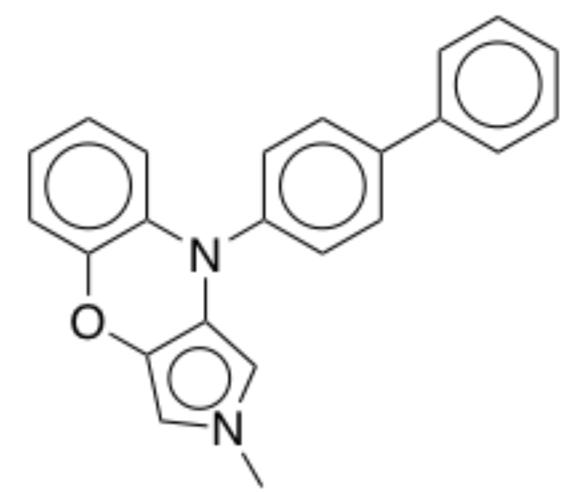


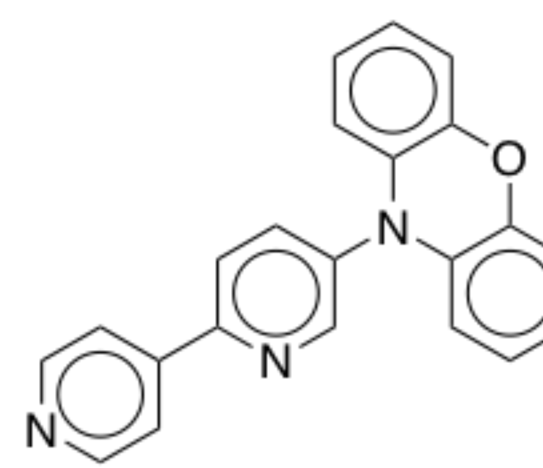


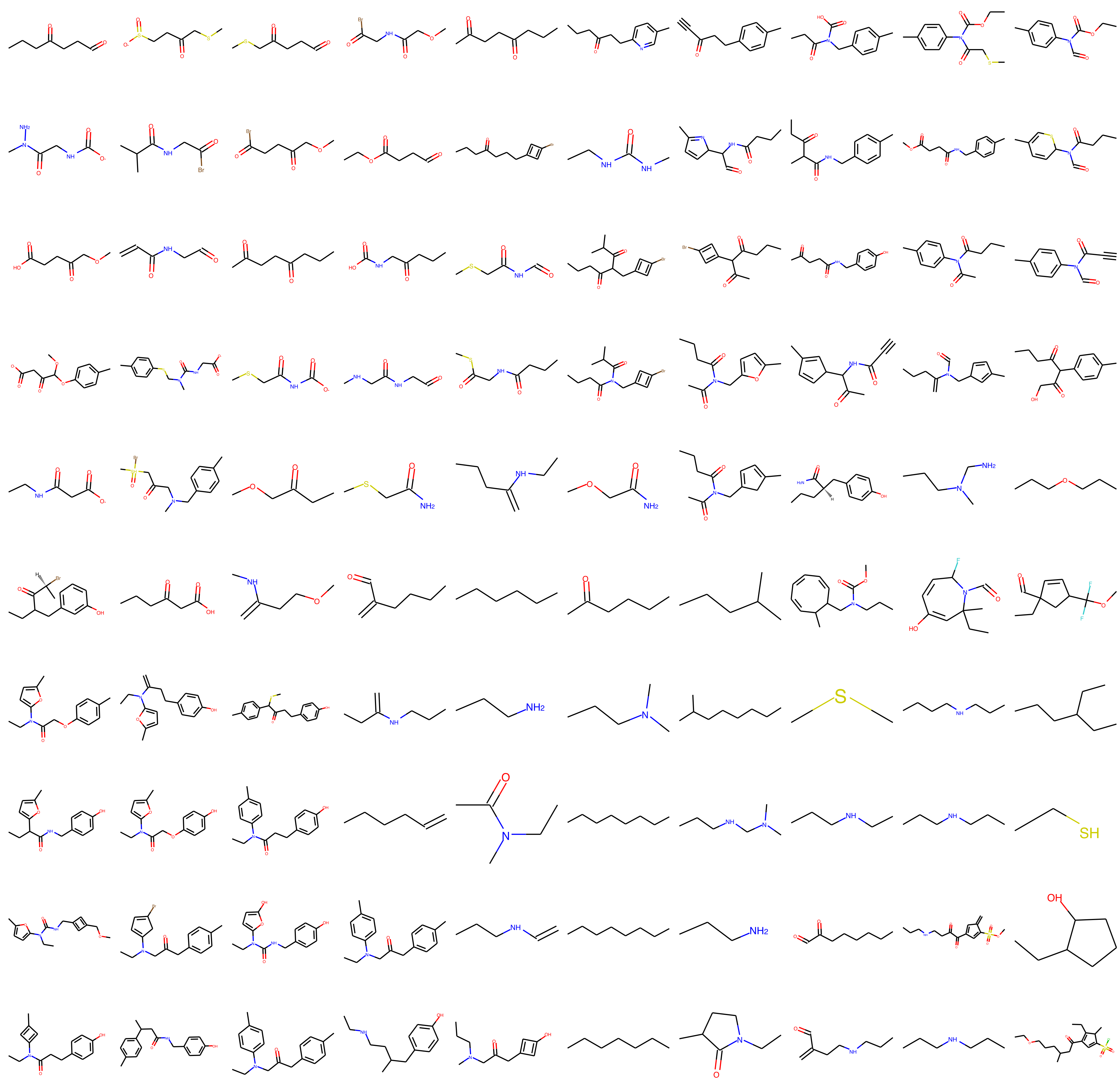












Course Themes

- Start with a simple model and add to it
  - Linear regression or PCA is a special case of almost everything
- A few ‘lego bricks’ are enough to build most models
  - Gaussians, Categorical variables, Linear transforms, Neural networks
  - The exact form of each distribution/function shouldn’t matter much
  - Your model should have a million parameters in it somewhere (the real world is messy!)
- Model checking is hard and important
  - Learning algorithms are especially hard to debug

Computation

- Later assignments will involve a bit of programming. Can use whatever language you want, but Python + Numpy is recommended.

- For fitting and inference in high-dimensional models, gradient-based methods are basically the only game in town

- Lots of methods conflate model and fitting algorithm, we will try to separate these

ML as a bag of tricks

Fast special cases:

- K-means
- Kernel Density Estimation
- SVMs
- Boosting
- Random Forests
- K-Nearest Neighbors

Extensible family:

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

Regularization as a bag of tricks

Fast special cases:

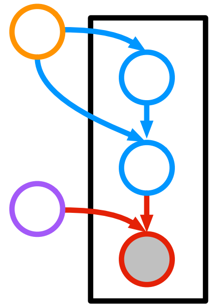
- Early stopping
- Ensembling
- L2 Regularization
- Gradient noise
- Dropout
- Expectation-Maximization

Extensible family:

- Stochastic variational inference

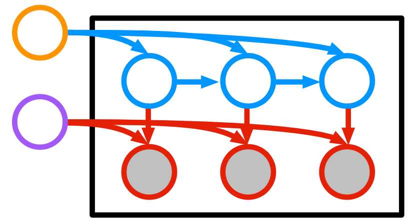
A language of models

- Hidden Markov Models, Mixture of Gaussians, Logistic Regression
- These are simply "sentences" - examples from a language of models.
- We will try to show larger family, and point out common special cases.



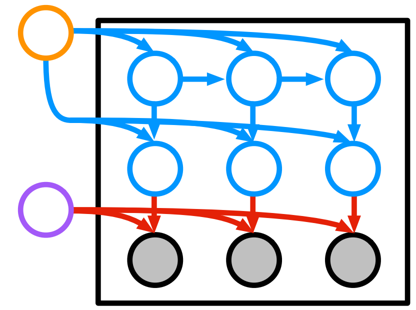
[1]

Gaussian mixture model



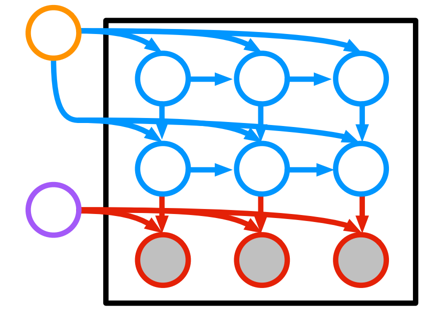
[2]

Linear dynamical system



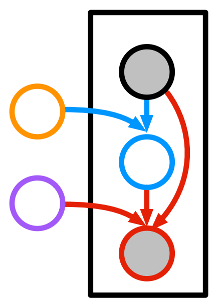
[3]

Hidden Markov model



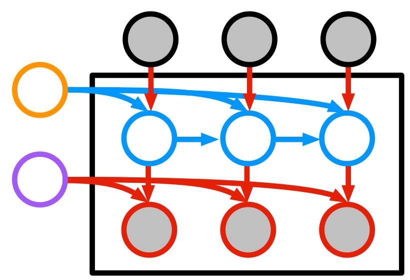
[4]

Switching LDS



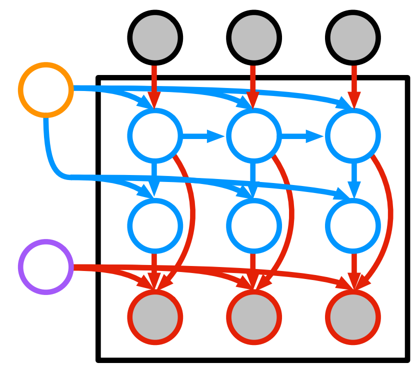
[5]

Mixture of Experts



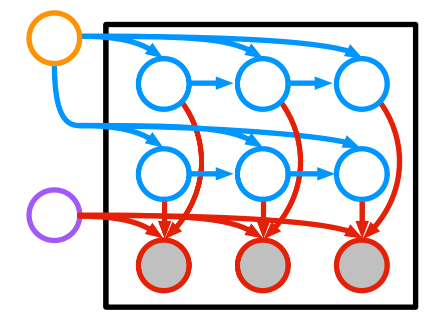
[2]

Driven LDS



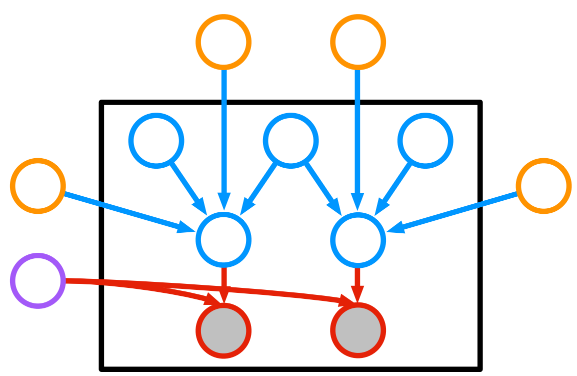
[6]

IO-HMM



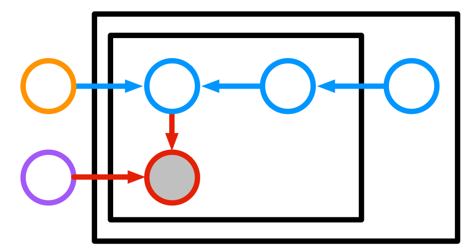
[7]

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

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

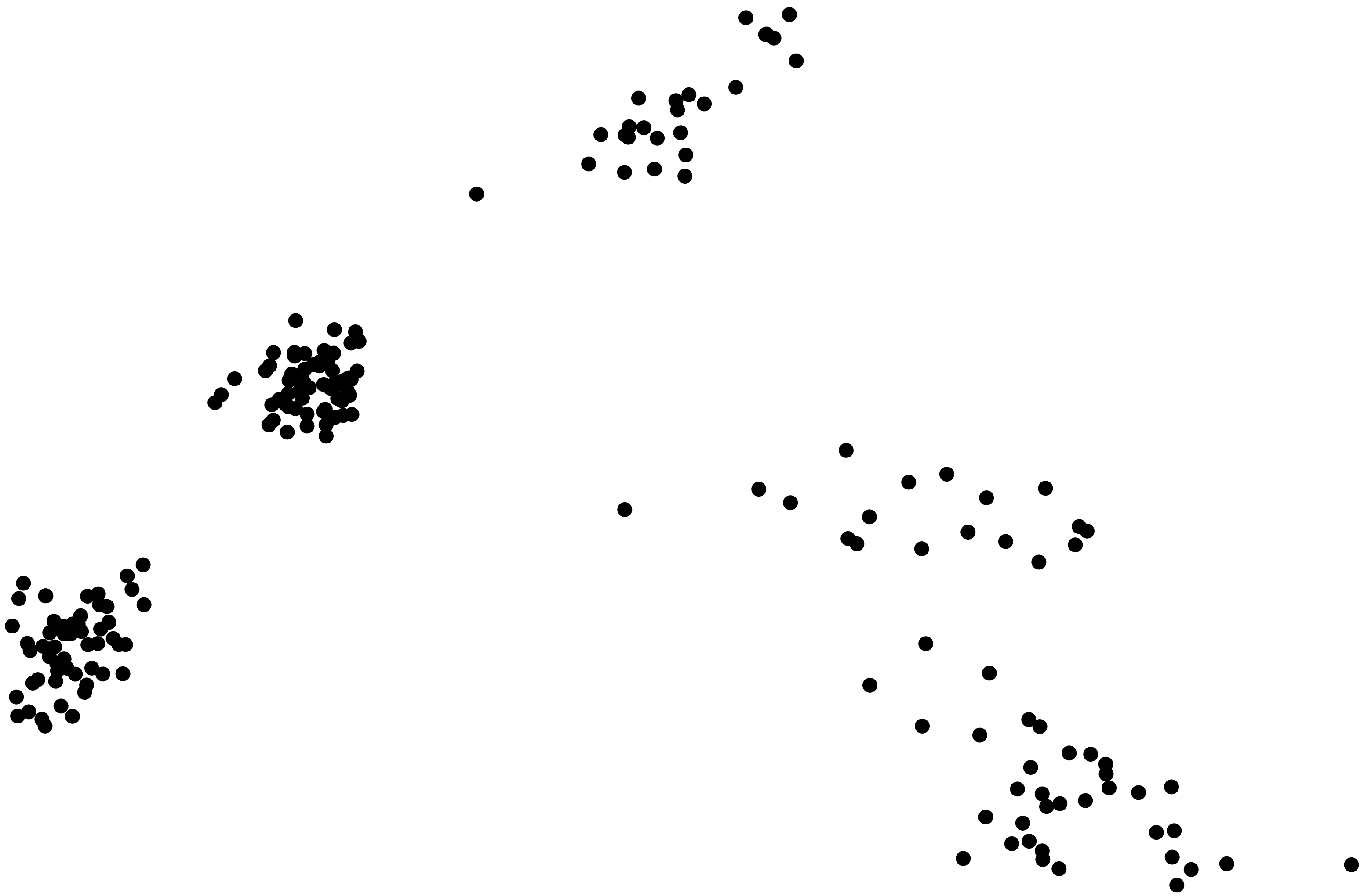
Extensible family:

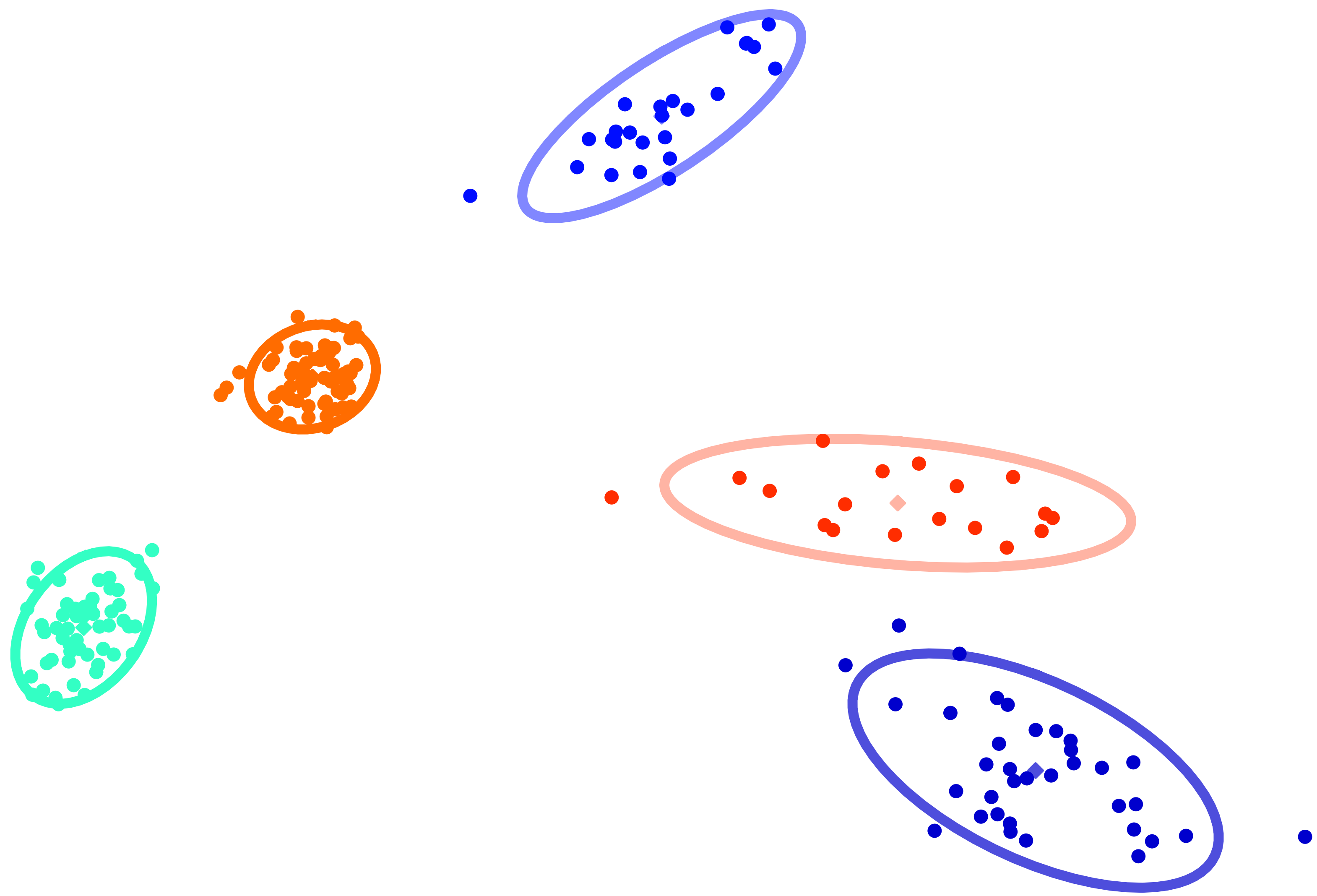
- Deep probabilistic latent-variable models + decision theory

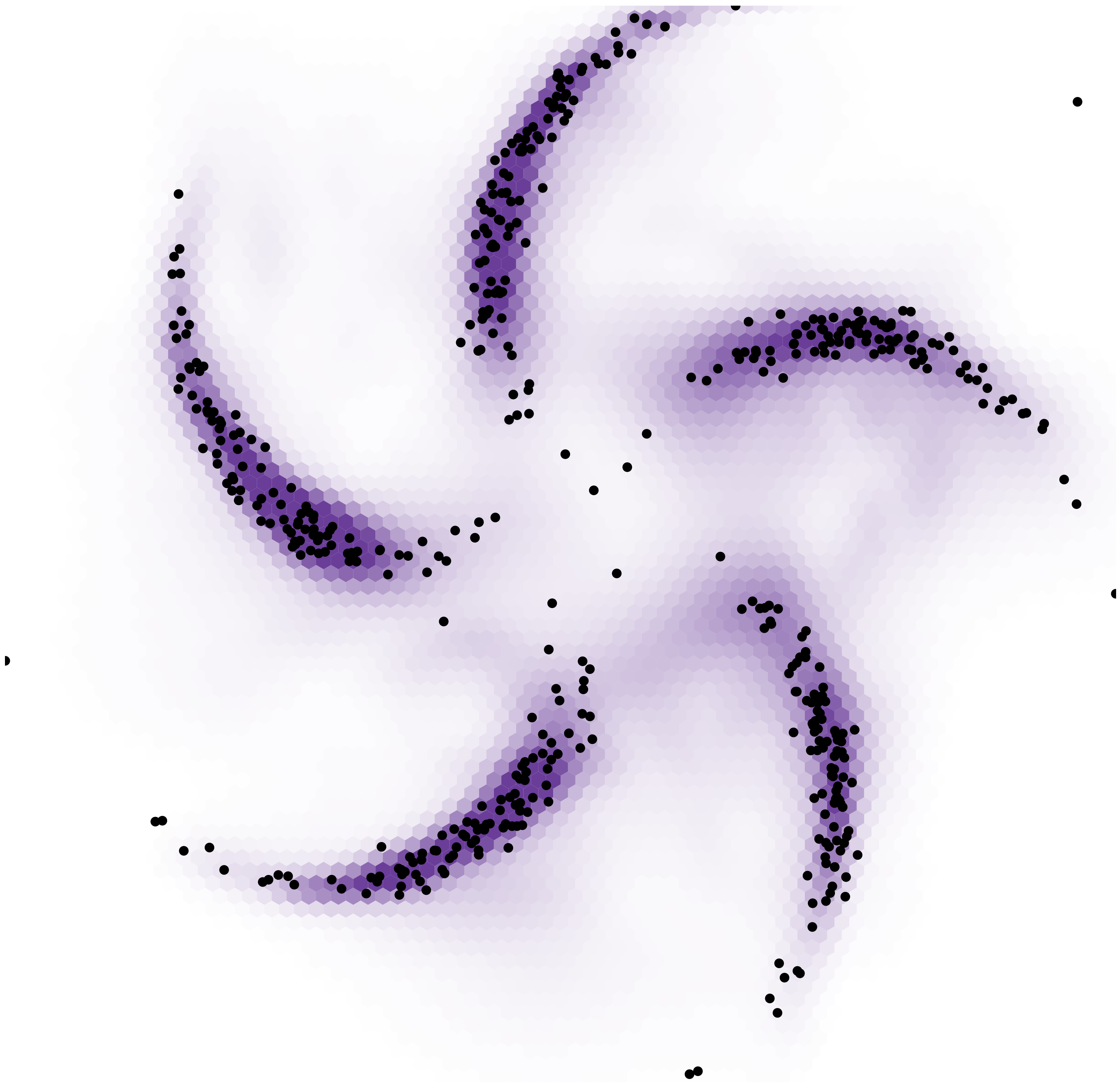
Advantages of probabilistic latent-variable models

- **Data-efficient learning** - automatic regularization, can take advantage of more information
- **Compose-able models** - e.g. incorporate data corruption model. Different from composing feedforward computations
- **Handle missing + corrupted data** (without the standard hack of just guessing the missing values using averages).
- **Predictive uncertainty** - necessary 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

- Cons:
    - intractable integral over latent variables
- Examples: medical diagnosis, image modeling





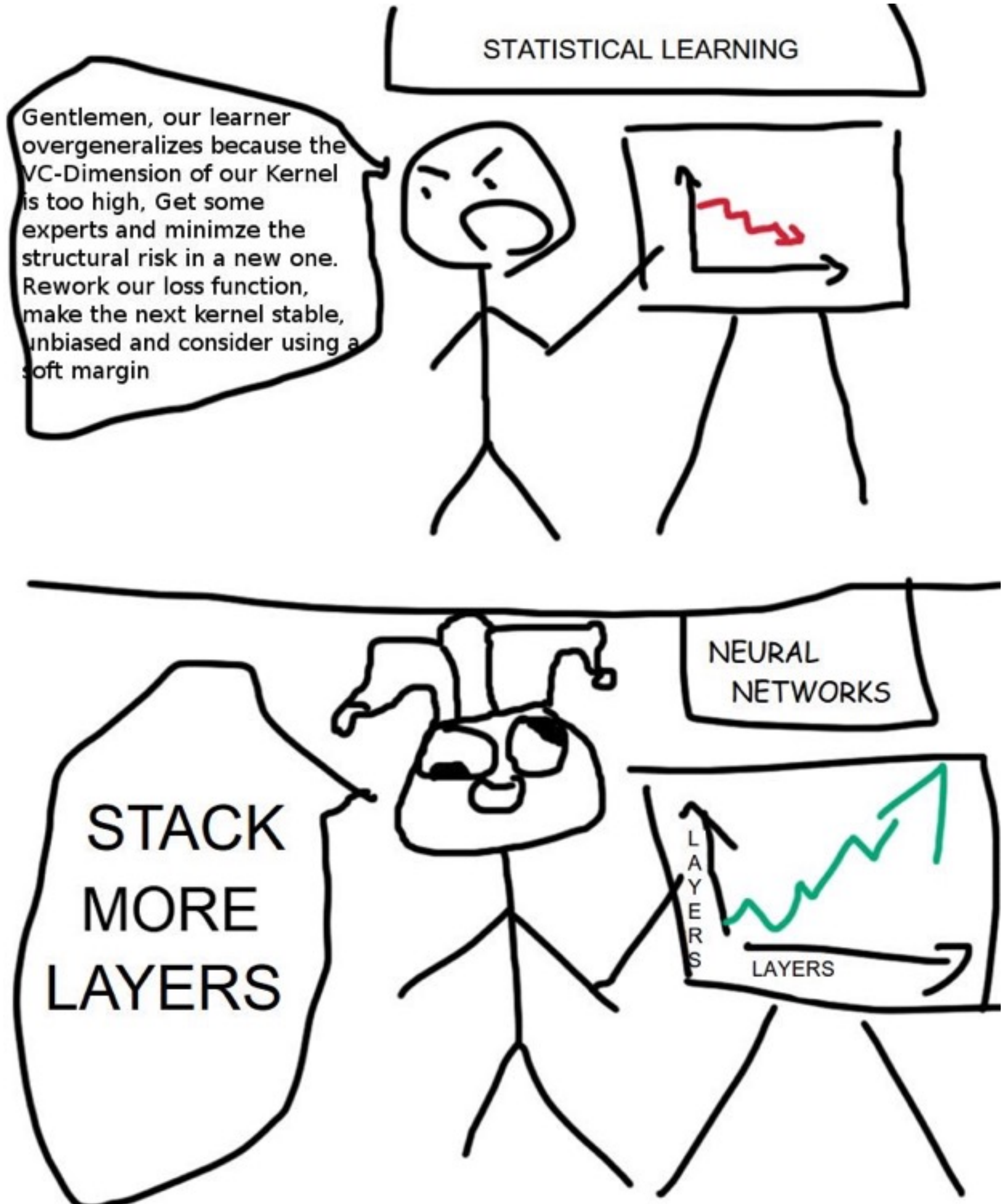


Probabilistic graphical models

+ structured representations
+ priors and uncertainty
+ data and computational efficiency
– rigid assumptions may not fit
– feature engineering
– top-down inference

Deep learning

– neural net “goo”
– difficult parameterization
– can require lots of data
+ flexible
+ feature learning
+ recognition networks


STATISTICAL LEARNING
Gentlemen, our learner overgeneralizes because the VC-Dimension of our Kernel is too high, Get some experts and minimze the structural risk in a new one. Rework our loss function, make the next kernel stable, unbiased and consider using a soft margin
NEURAL NETWORKS
STACK MORE LAYERS
LAYERS
LAYERS

The unreasonable easiness of deep learning

- Recipe: define an objective function (i.e. probability of data given params)
- Optimize params to maximize objective
- Gradients are computed automatically, you just define model by some computation
- Show demo here

Differentiable models

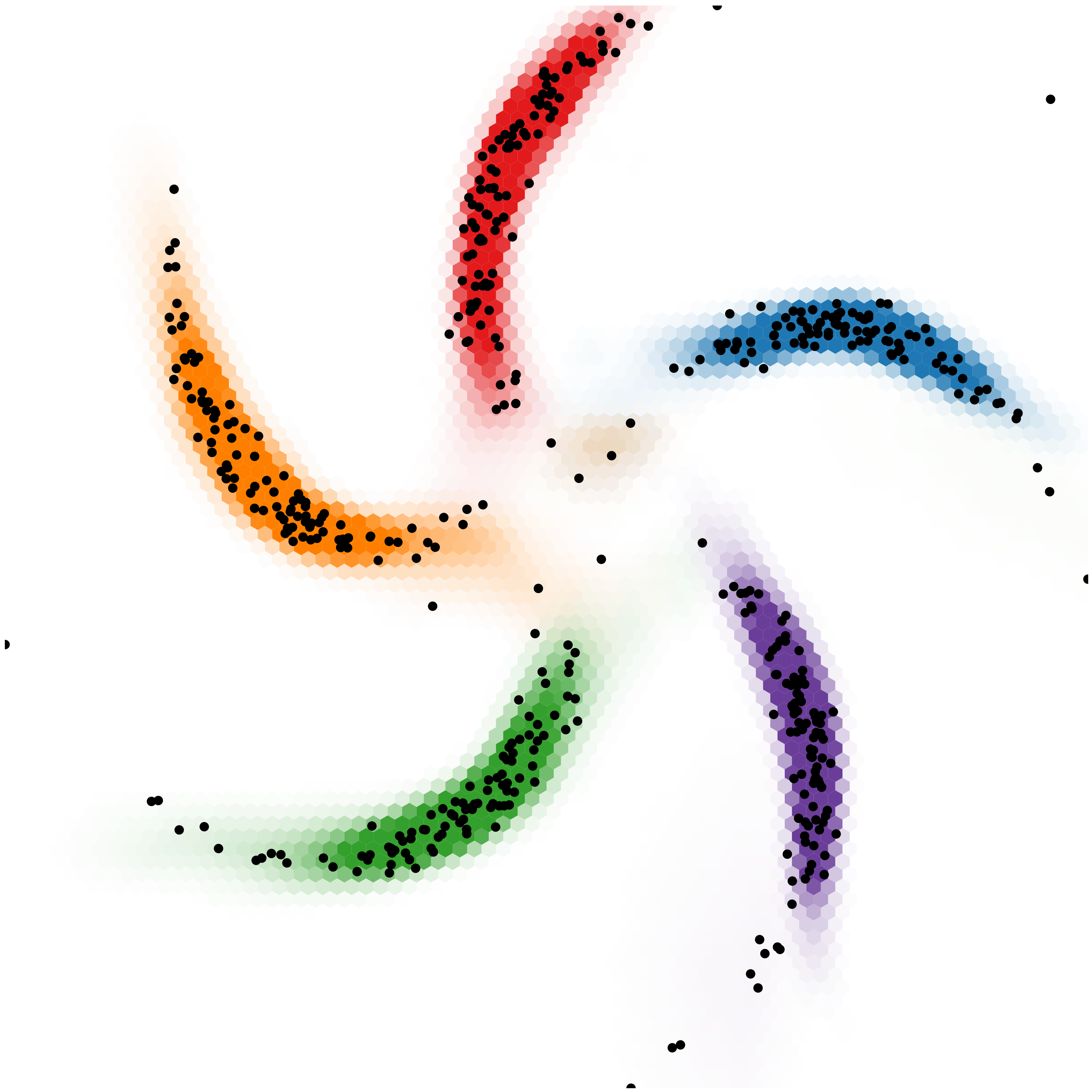
- Model distributions implicitly by a variable pushed through a deep net:

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

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

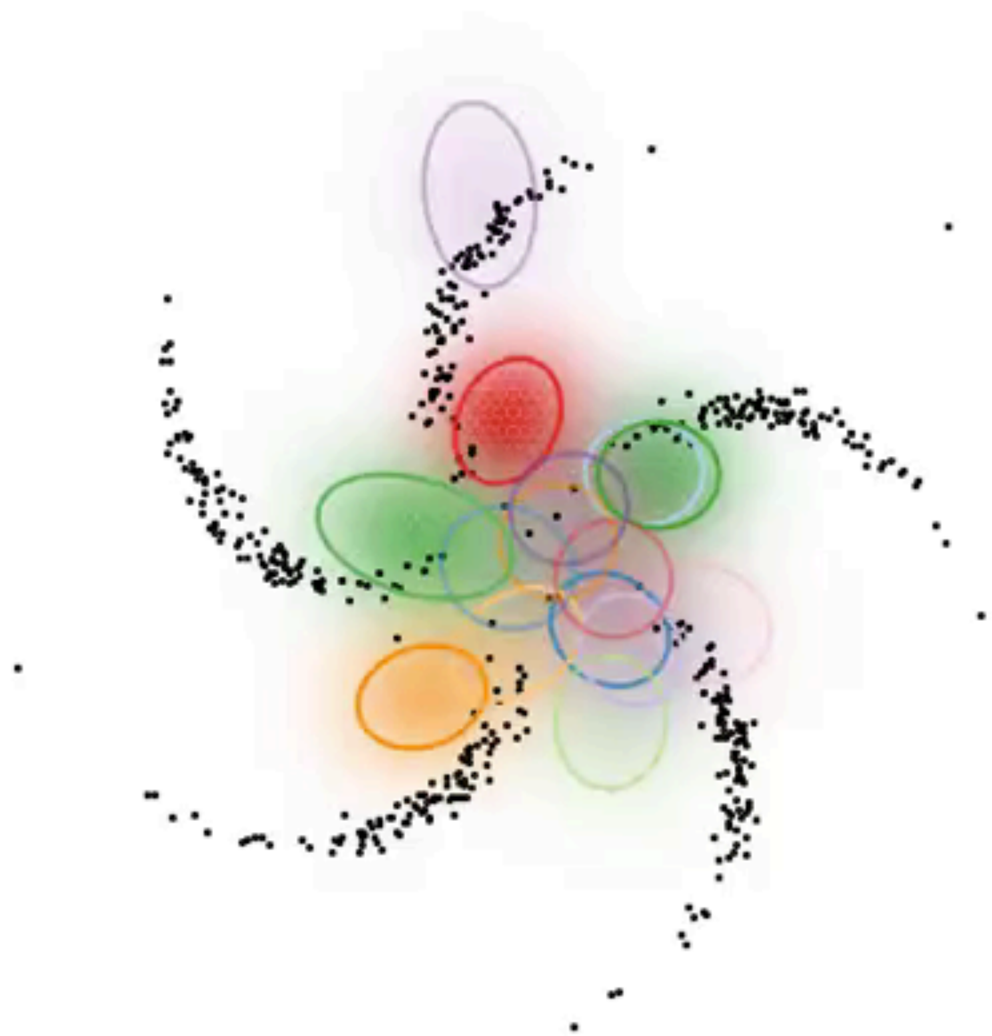
$$p(y|x) = \mathcal{N}(y|\mu = f_\theta(x), \Sigma = g_\theta(x))$$

- Optimize all parameters using stochastic gradient descent

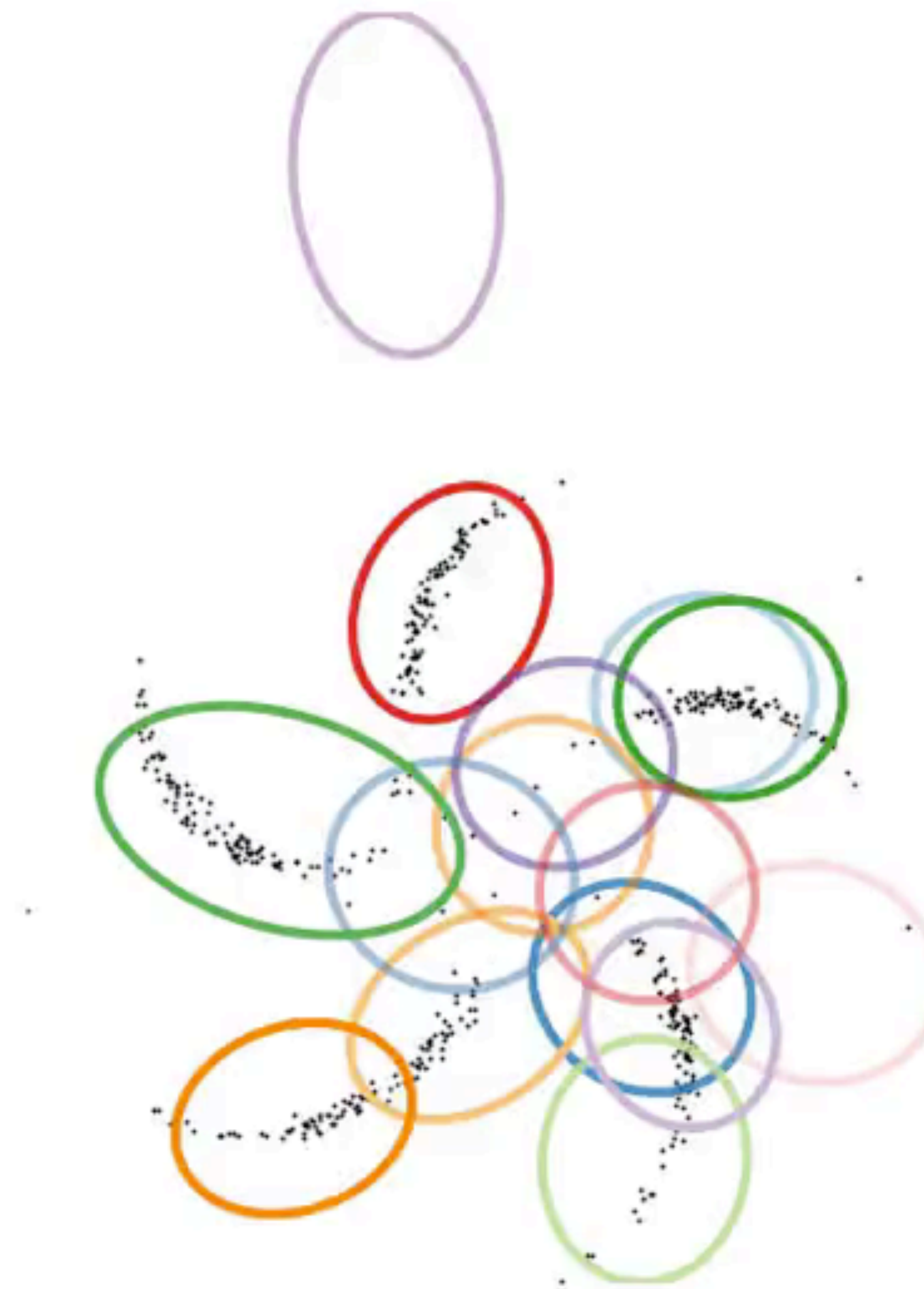


Modeling idea: graphical models on latent variables,
neural network models for observations

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

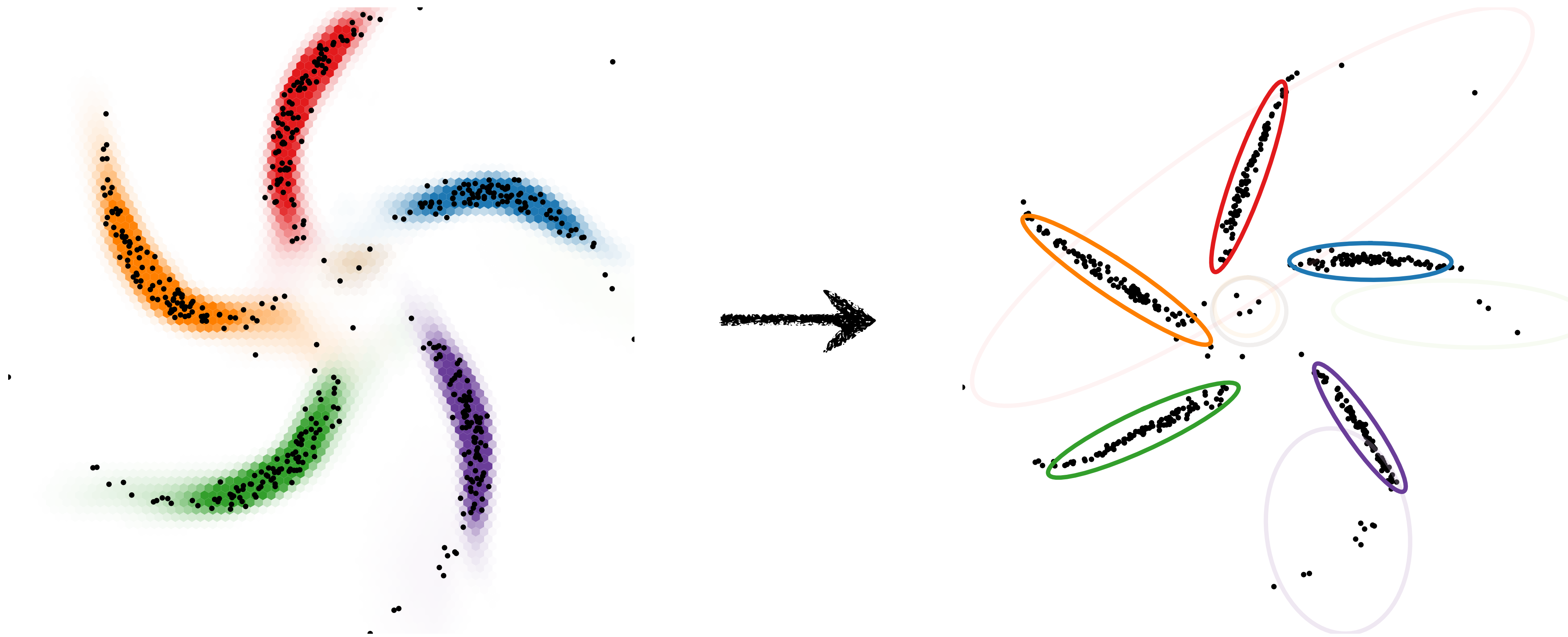


data space

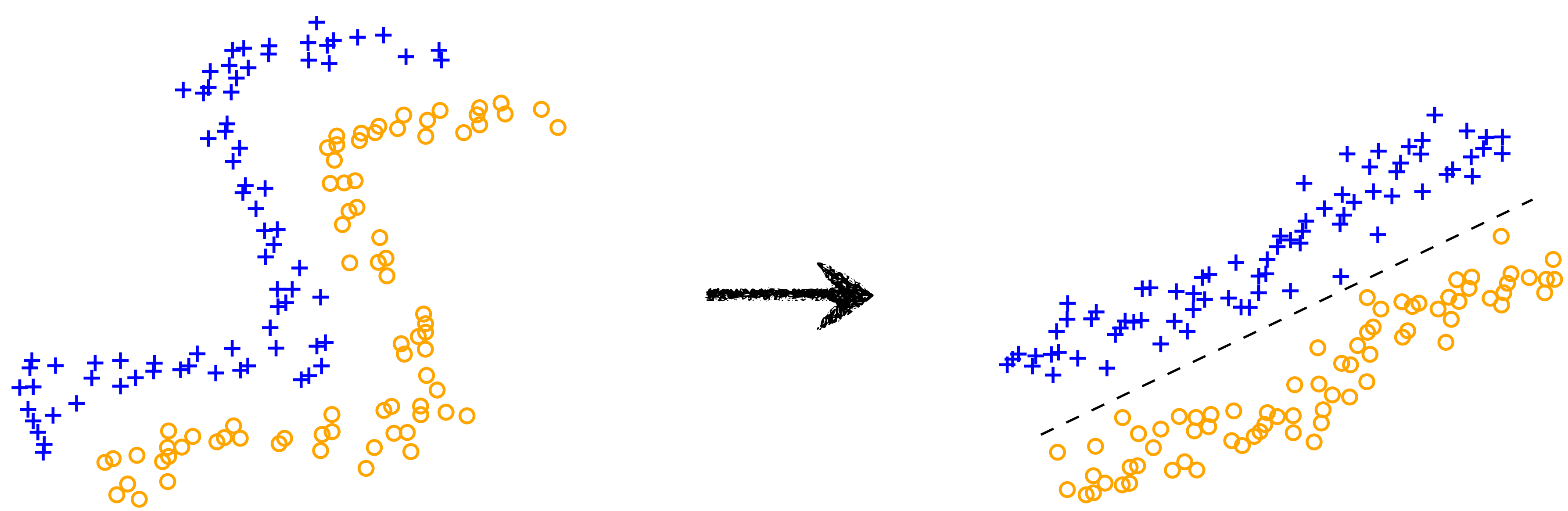


latent space

unsupervised
learning



supervised
learning



Courtesy of Matthew Johnson

Learning outcomes

- Know standard algorithms (bag of tricks), when to use them, and their limitations. For basic applications and baselines.

- Know main elements of language of deep probabilistic models (distributions, expectations, latent variables, neural networks) and how to combine them. For custom applications + research

- Know standard computational tools (Monte Carlo, Stochastic optimization, regularization, automatic differentiation). For fitting models

Tentative list of topics

- Linear methods for regression + classification, Bayesian linear regression
- Probabilistic Generative and Discriminative models, Regularization methods
- Stochastic Optimization (practically important)
- Neural Networks
- Model Comparison and marginal likelihood (conceptually important)
- Stochastic Variational Inference
- Time series and recurrent models
- Mixture Models, Graphical Models and Bayesian Networks
- Kernel Methods, Gaussian processes, Support Vector Machines

Quiz

Machine-learning-centric History of Probabilistic Models

- **1940s - 1960s** Motivating probability and Bayesian inference
- **1980s - 2000s** Bayesian machine learning with MCMC
- **1990s - 2000s** Graphical models with exact inference
- **1990s** - **present** Bayesian Nonparametrics with MCMC (Indian Buffet process, Chinese restaurant process)
- **1990s - 2000s** Bayesian ML with mean-field variational inference
- **2000s - present** Probabilistic Programming
- **2000s - 2013** Deep undirected graphical models (RBMs, pretraining)
- **2010s** - **present** Stan - Bayesian Data Analysis with HMC
- **2000s** - **2013** Autoencoders, denoising autoencoders
- **2000s** - **present** Invertible density estimation
- **2013** - **present** Stochastic variational inference, variational autoencoders
- **2014** - **present** Generative adversarial nets, Real NVP, Pixelnet
- **2016 - present** Lego-style deep generative models (attend, infer, repeat)