# CSC412/2506 Probabilistic Learning and Reasoning

Introduction

Jesse Bettencourt

# Today

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

- **Thursday**: No tutorial this week!

# Course Website

- www.cs.toronto.edu/~jessebett/CSC412
- Contains all course information, slides, etc.

# Evaluation

- Assignment 1: due Feb ~8 worth 15%
- Assignment 2: due March ~15 worth 15%
- Assignment 3: due Apr ~5 worth 20%
- 1-hour Midterm: Feb 14 worth 20%
- 3-hour Final: April ? 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
- CSC321: Neural networks - about 30% overlap

# Textbooks + Resources

- No required textbook
- Kevin Murphy (2012), *Machine Learning: A Probabilistic Perspective*.
- David MacKay (2003) *Information Theory, Inference, and Learning Algorithms*

# 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.

- **Active learning and RL:** Also get to choose actions that influence future information + reward. Can just use basic decision theory.

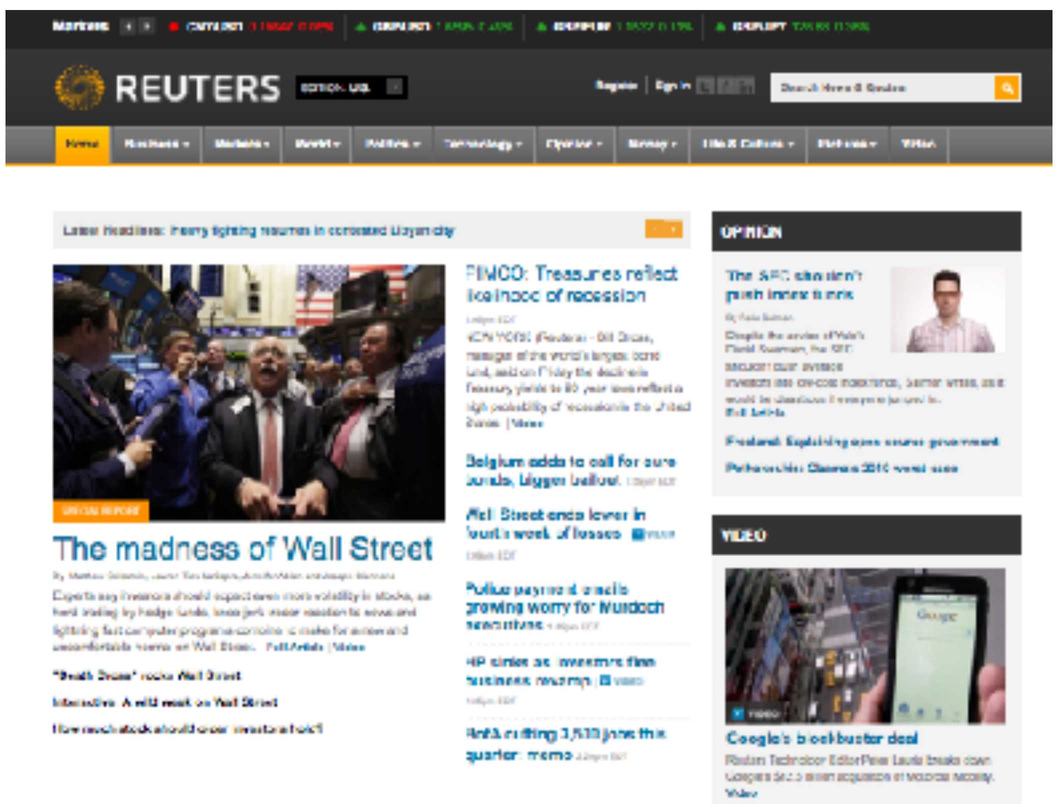
- **All just special cases** of estimating distributions from data: $p(y|x)$, $p(x)$, $p(x, y)$.

# Finding Structure in Data

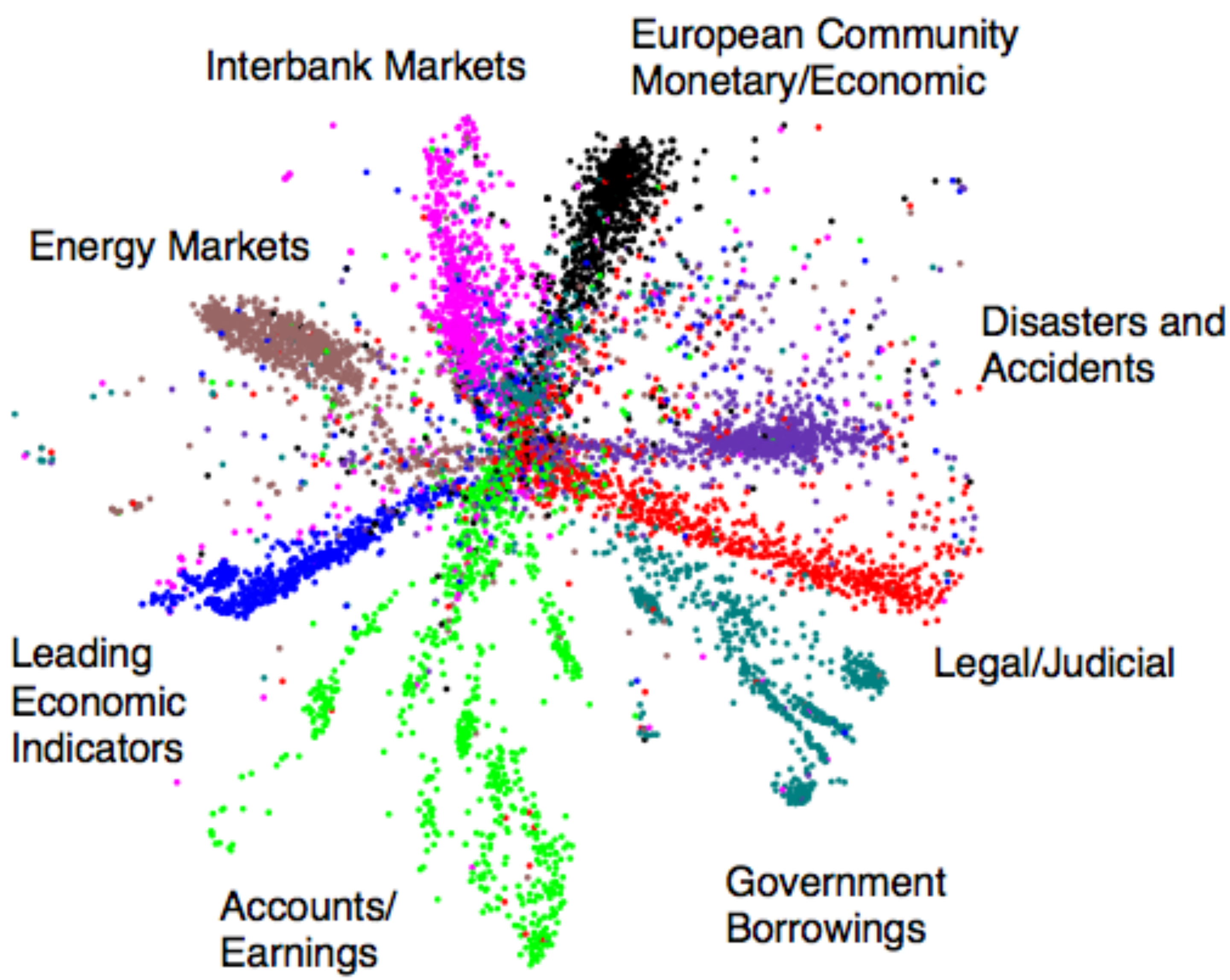
$$P(\mathbf{x}) = \frac{1}{\mathcal{Z}}\sum_{\mathbf{h}} \exp\left[\mathbf{x}^\top \mathbf{W}\mathbf{h}\right]$$

Vector of word counts on a webpage

Latent variables: hidden topics




804,414 newswire stories



# Matrix Factorization

Collaborative Filtering/
Matrix Factorization/

movielens
helping you find the right movies

amazon

| | | | | | |
|---|---|---|---|---|---|
| | ★★☆ | ? | ? | ★★☆ | ★★☆ |
| | ? | ★★☆ | ★★★ | ? | ★★★ |
| | ★★★ | ? | ★★☆ | ★★★ | ? |

## 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


movielens
helping you find the right movies

amazon


Learned ``genre”

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

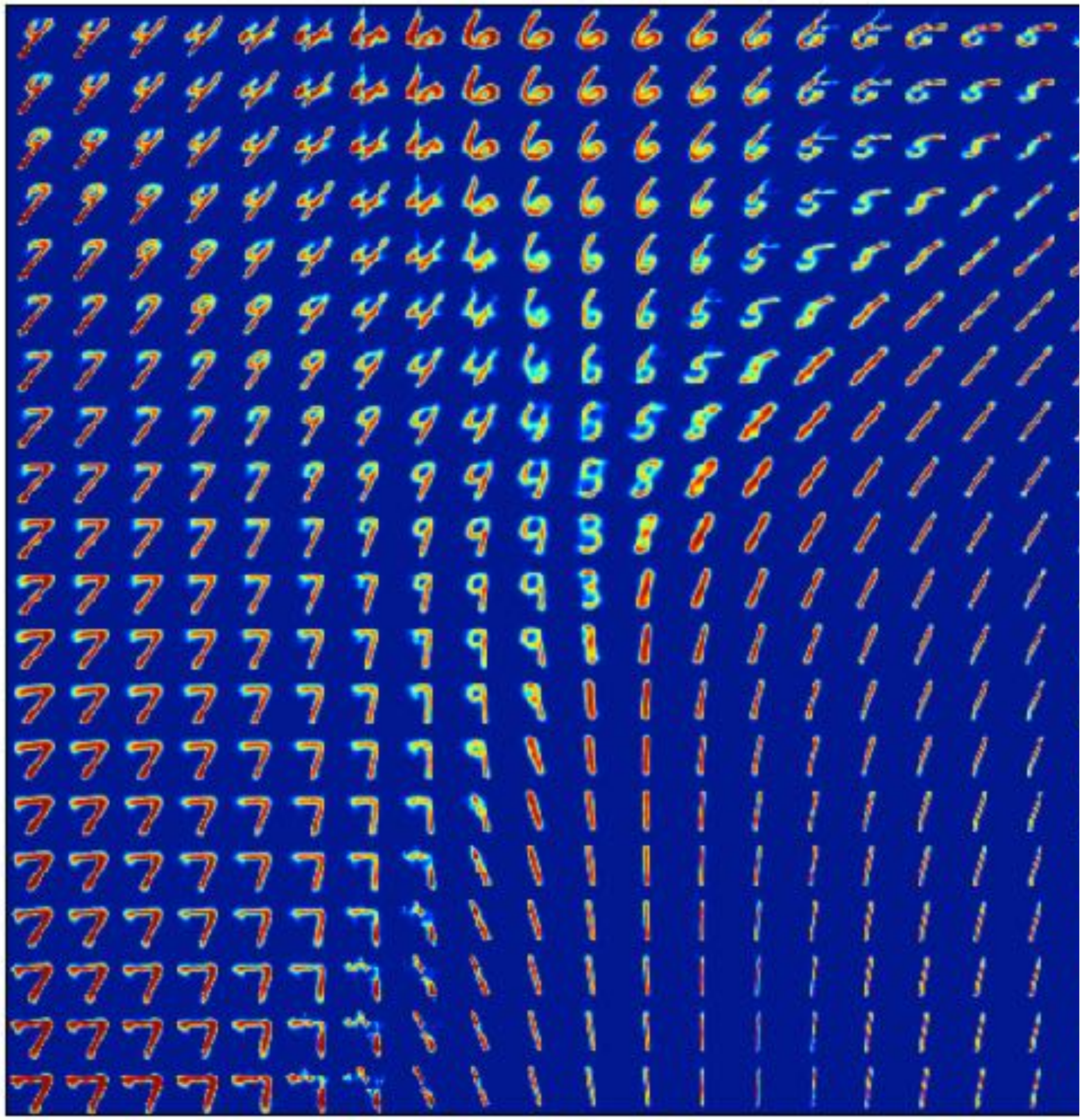
Fahrenheit 9/11
Bowling for Columbine
The People vs. Larry Flynt
Canadian Bacon
La Dolce Vita

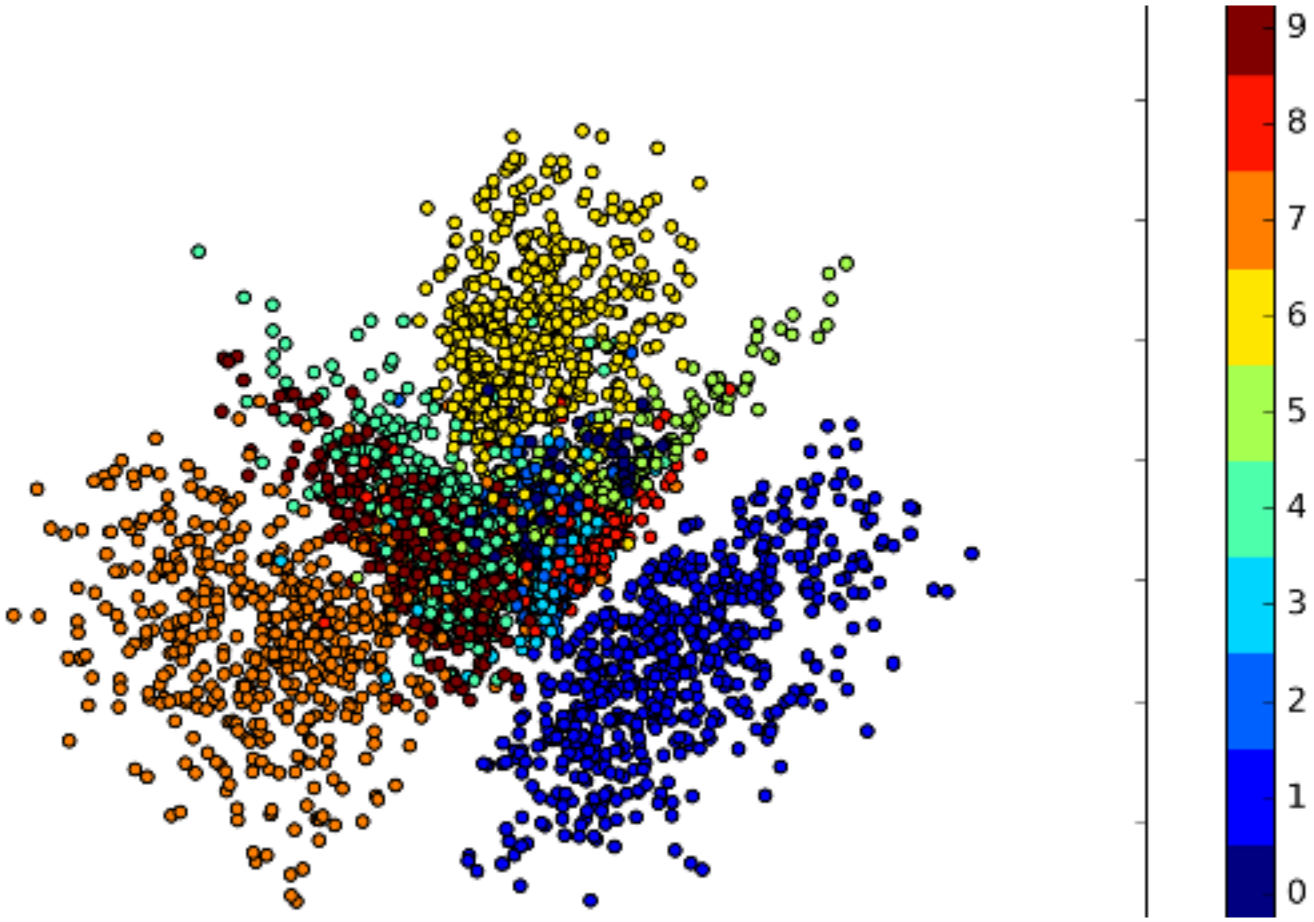
Independence Day
The Day After Tomorrow
Con Air
Men in Black II
Men in Black

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

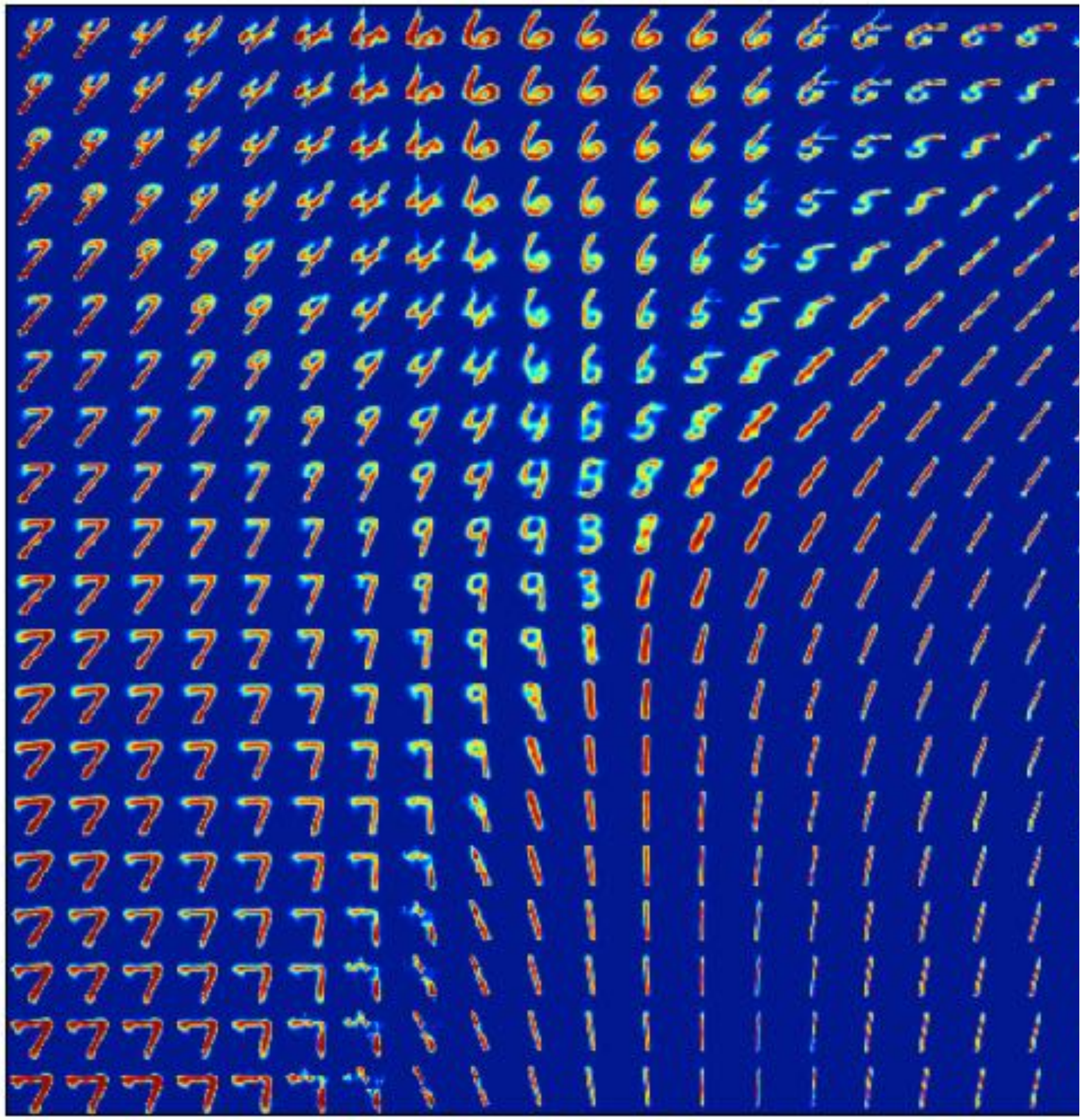
- Part of the wining solution in the Netflix contest (1 million dollar prize).

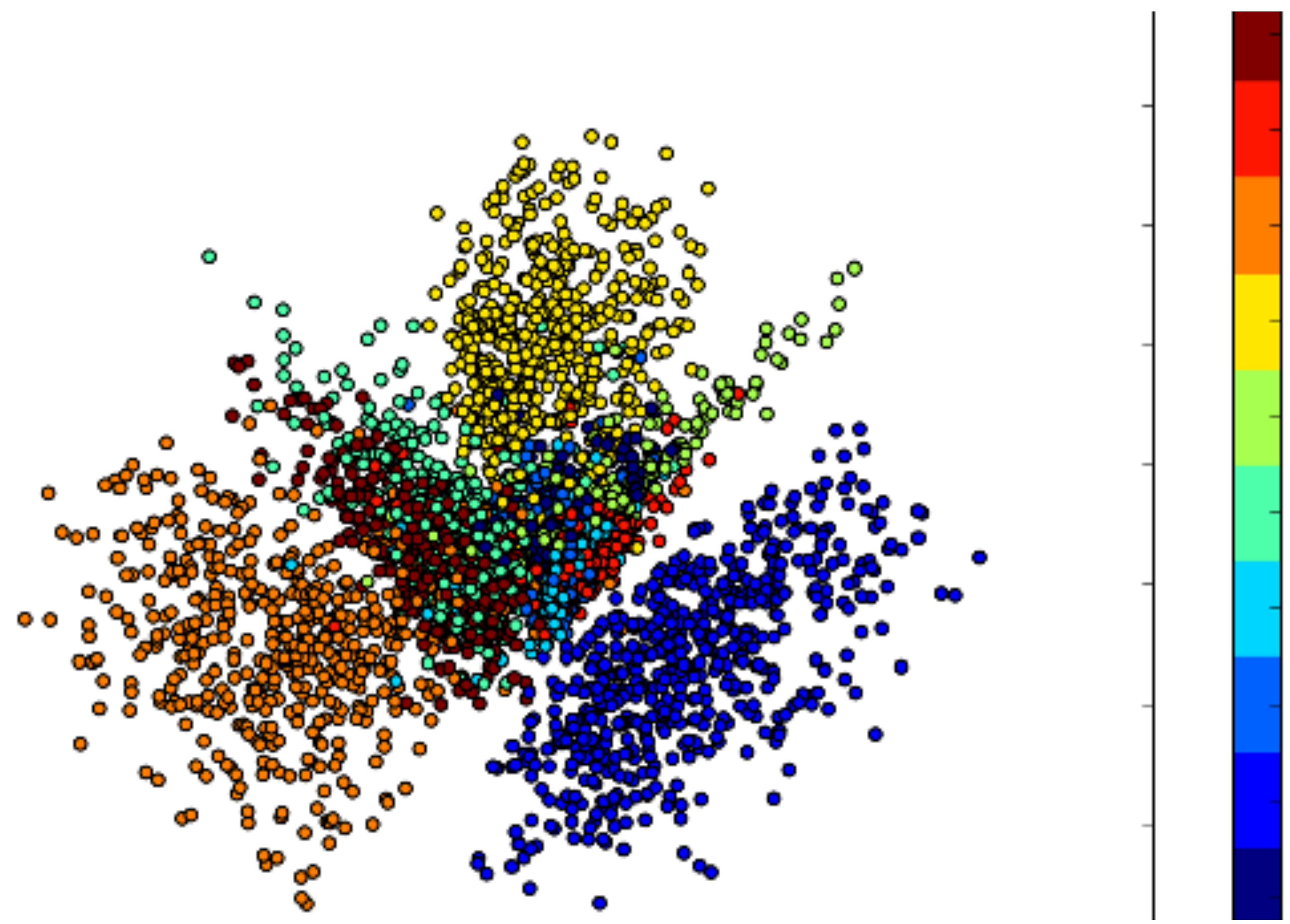
# Latent: Lower Dimensional Abstract Representation



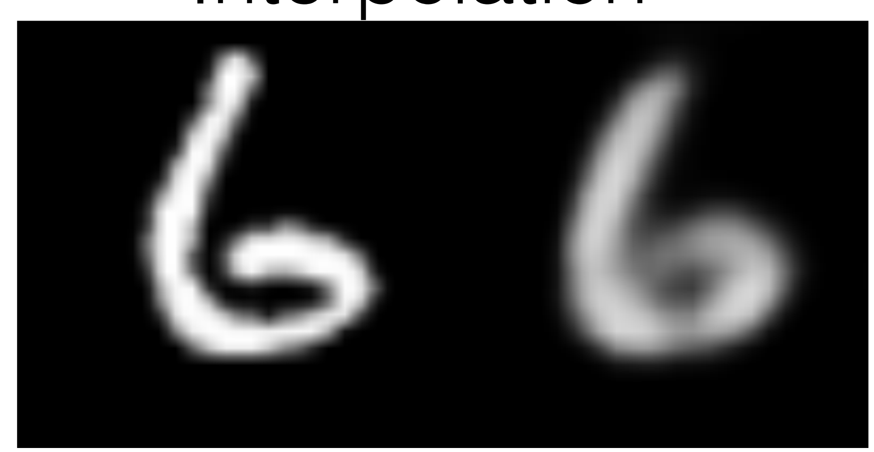


# Latent: Lower Dimensional Abstract Representation






Interpolation



data space latent space

*From Julien Despois' Latent space visualization — Deep Learning bits #2

# Multiple Kinds of Data in One Model

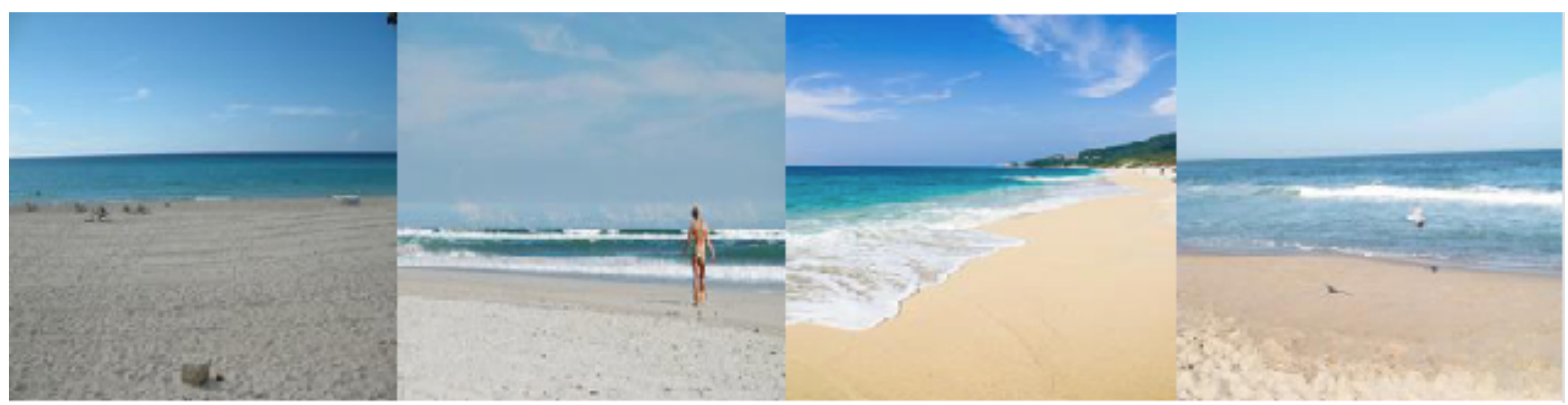
mosque, tower, building, cathedral, dome, castle

ski, skiing, skiers, skiiers, snowmobile

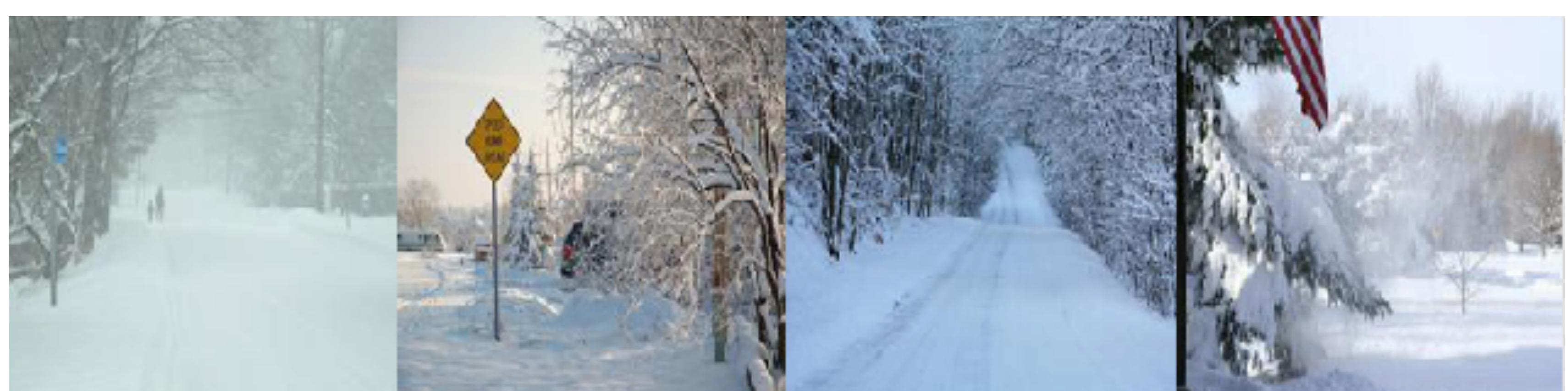
kitchen, stove, oven, refrigerator, microwave

bowl, cup, soup, cups, coffee

beach



snow



# Caption Generation

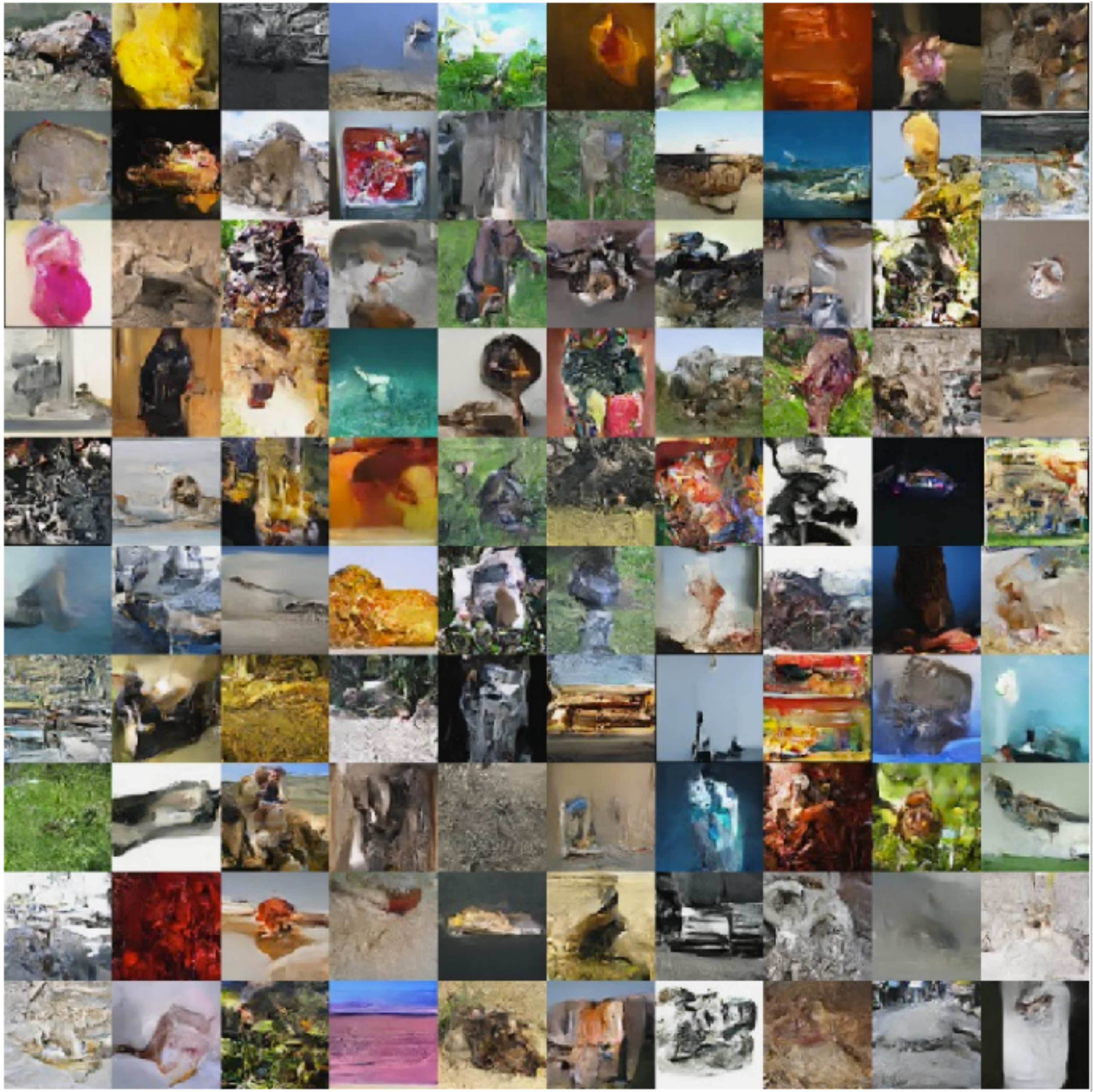
a car is parked in
the middle of nowhere .

a wooden table and chairs
arranged in a room .

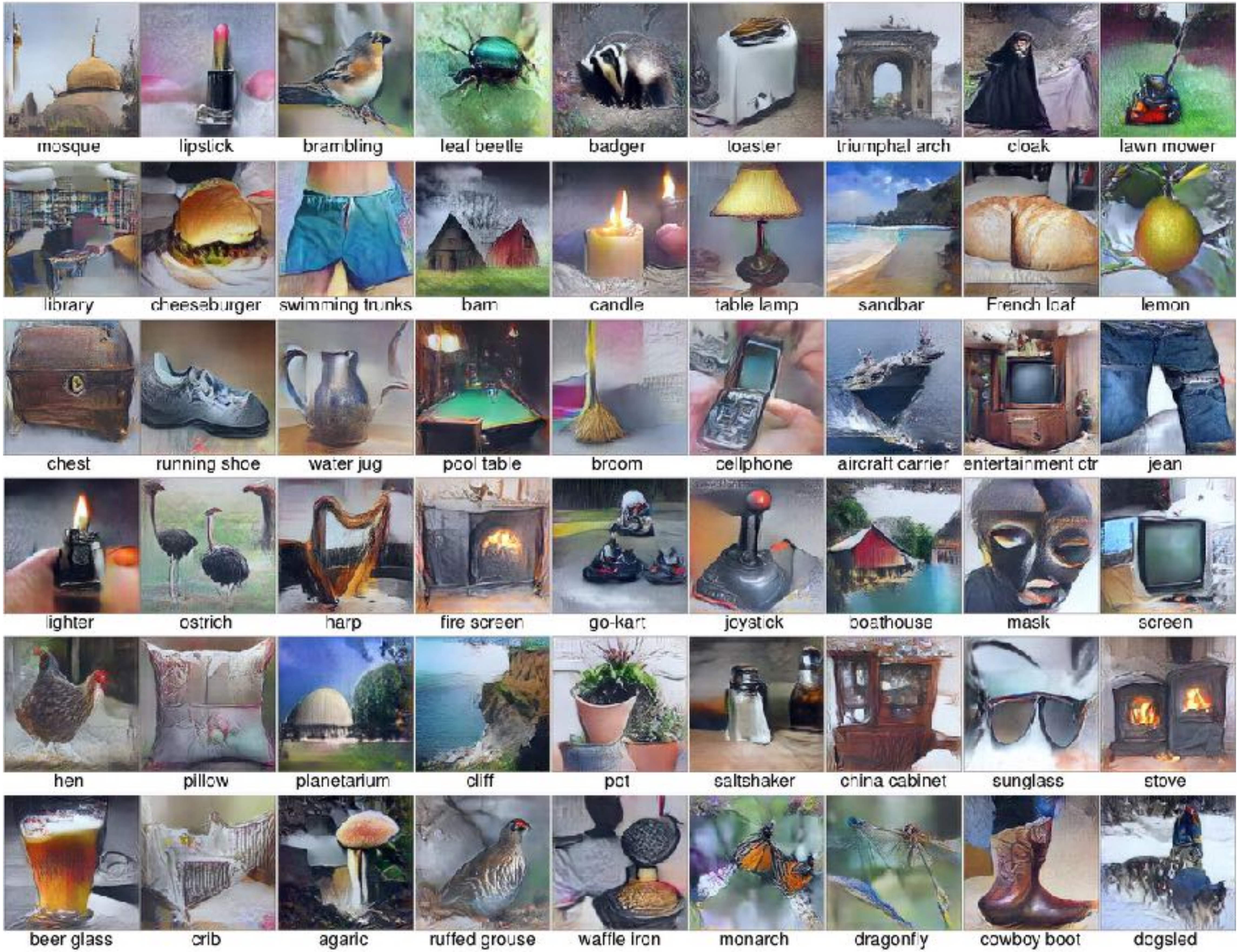
there is a cat sitting on a shelf .

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

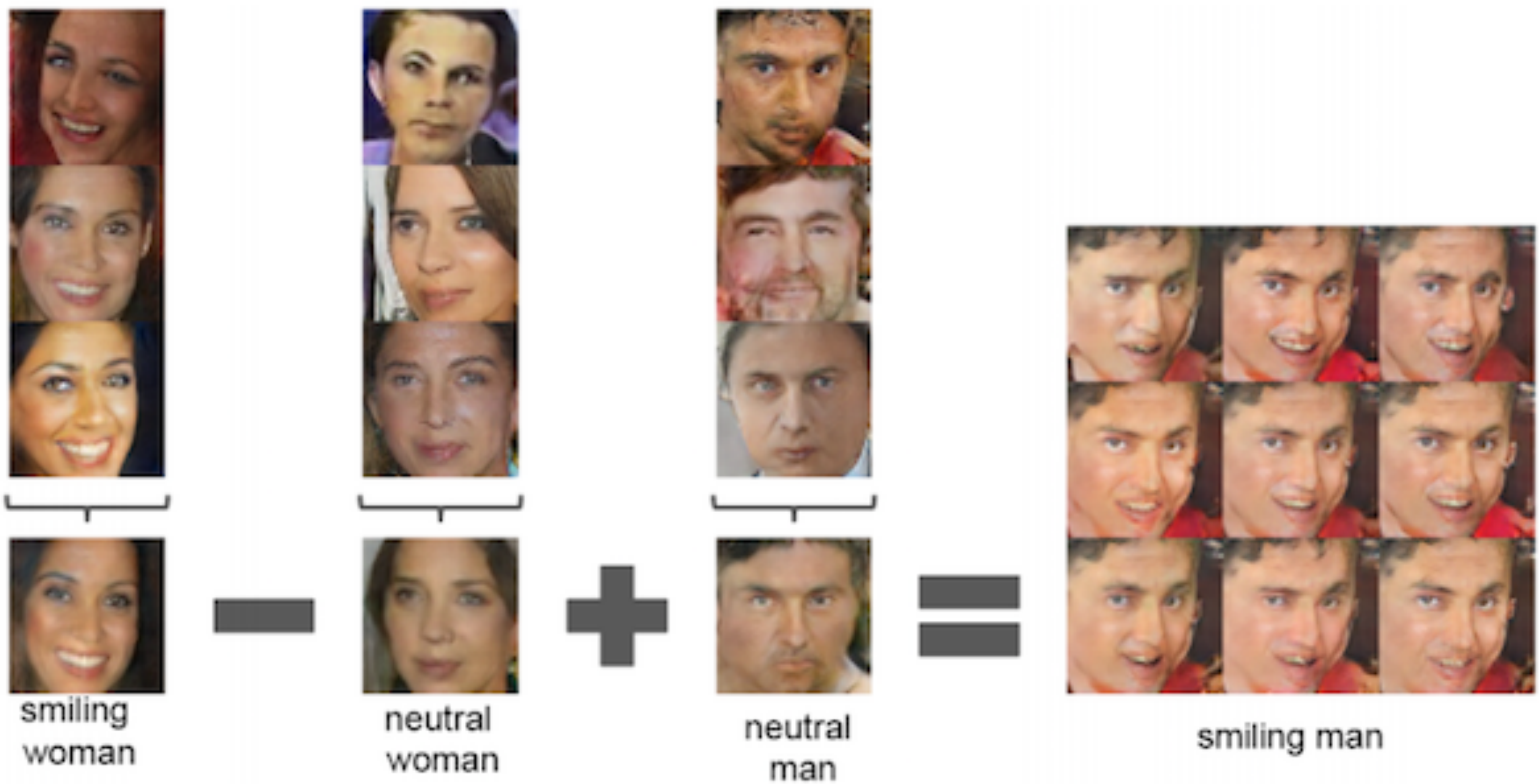



Nguyen A, Dosovitskiy A, Yosinski J, Brox T, Clune J (2016). *Synthesizing the preferred inputs for neurons in neural networks via deep generator networks.* Advances in Neural Information Processing Systems 29

A Style-Based Generator Architecture for Generative Adversarial Networks, 2018
Tero Karras, Samuli Laine, Timo Aila

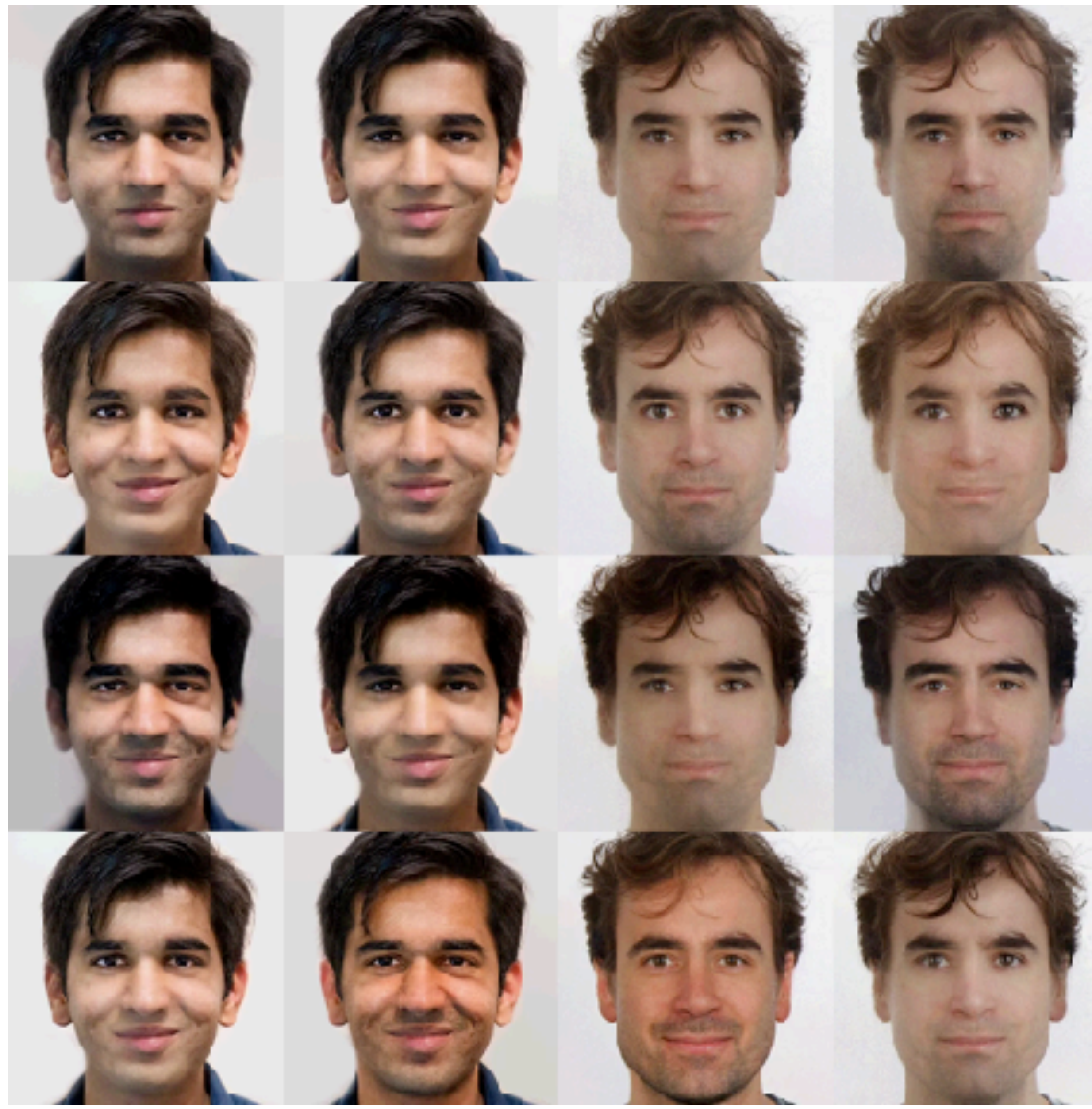

Pixel Recurrent Neural Networks, 2016
Aaron van den Oord, Nal Kalchbrenner, Koray Kavukcuoglu

# Arithmetic on Abstract Features




Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2015
Alec Radford, Luke Metz, Soumith Chintala

# Arithmetic on Abstract Features



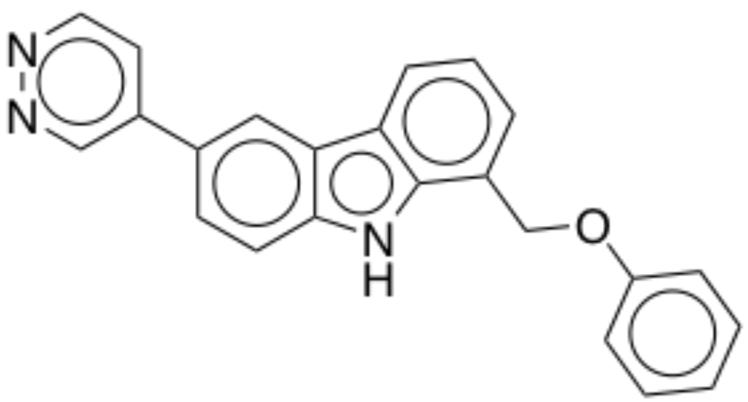
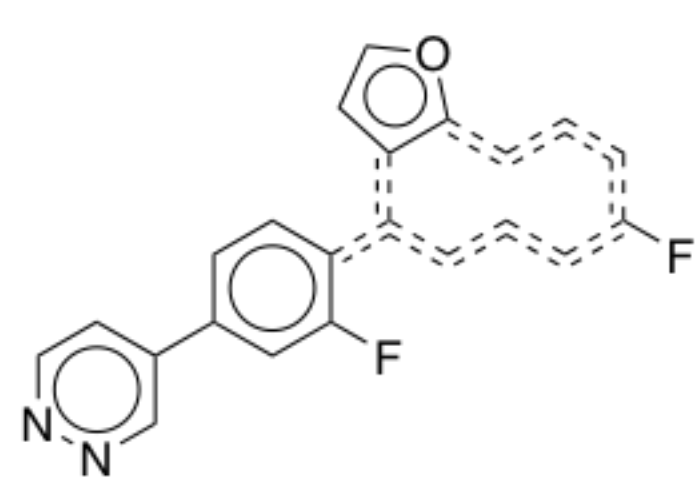
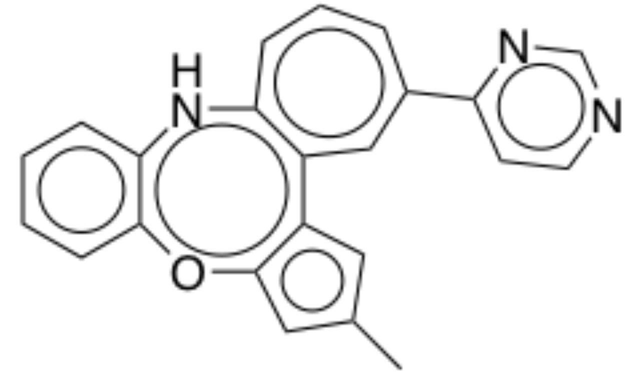
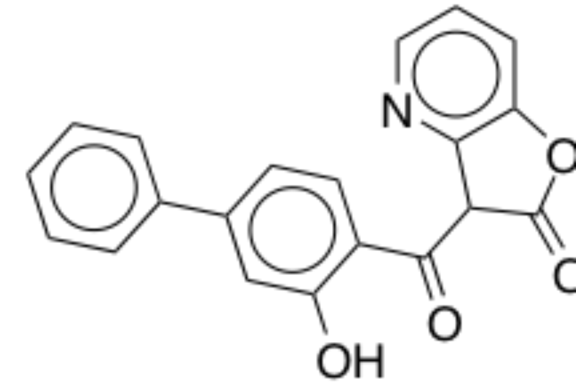
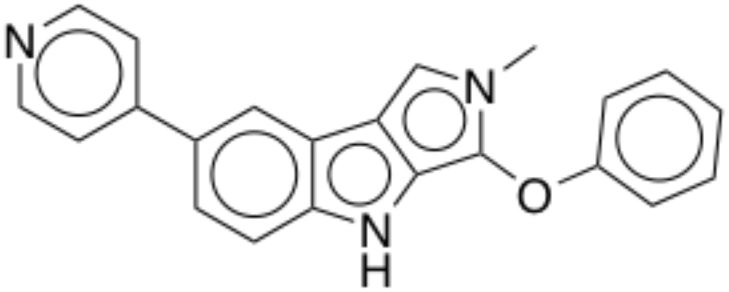
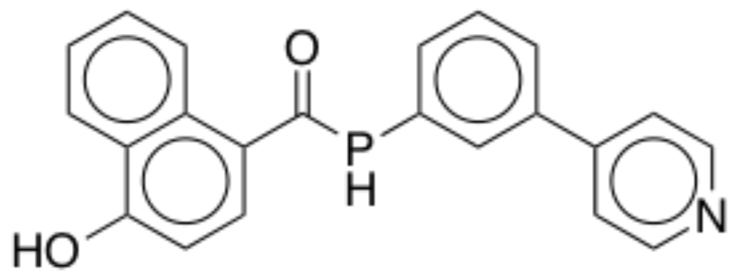
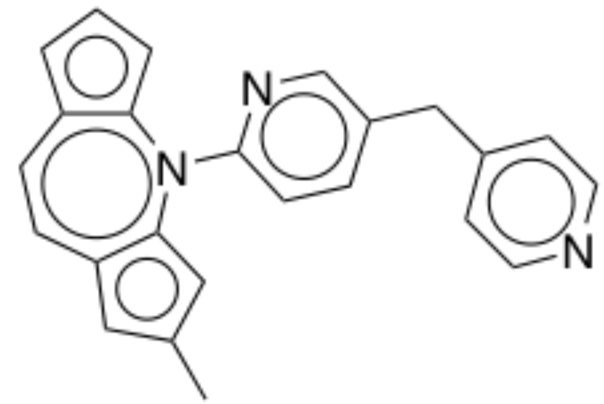
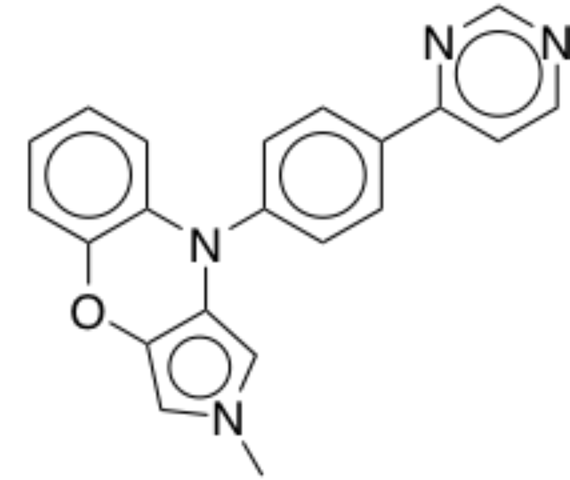
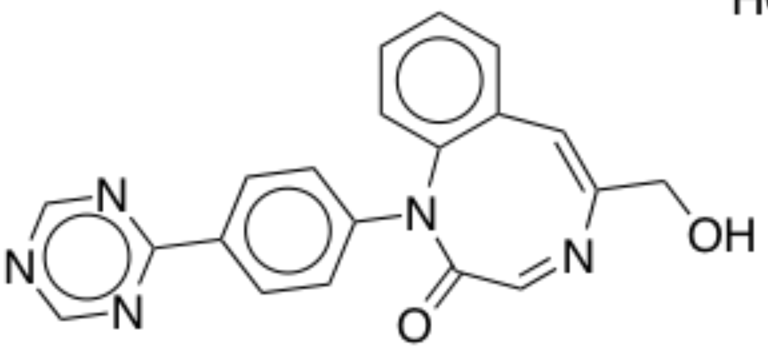
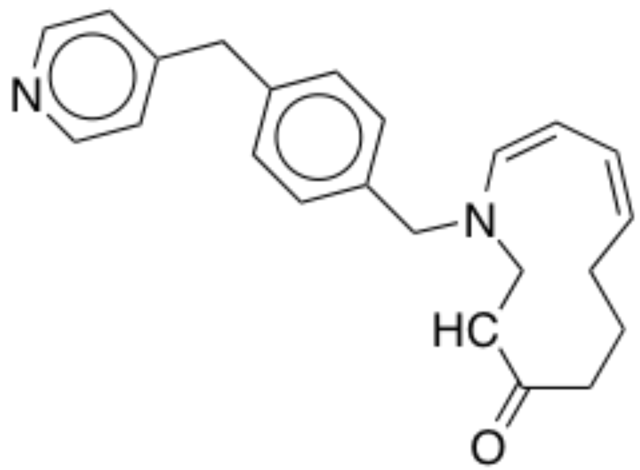
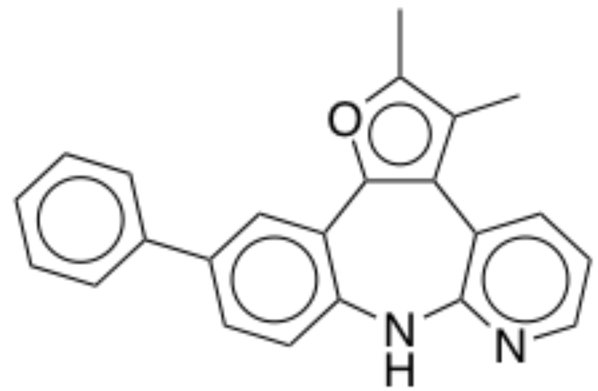
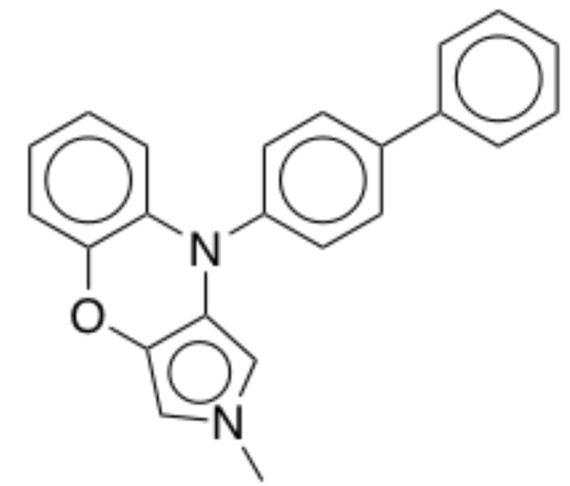
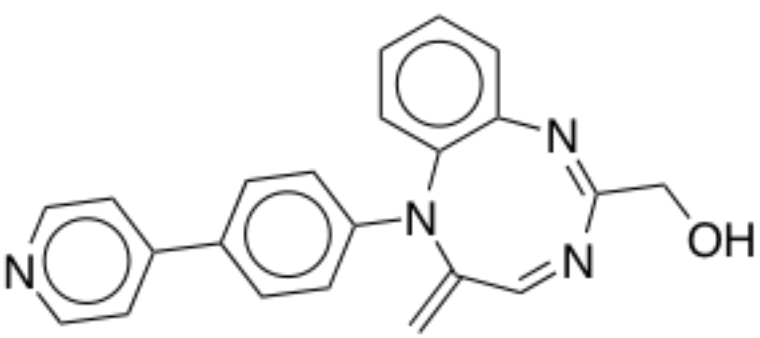
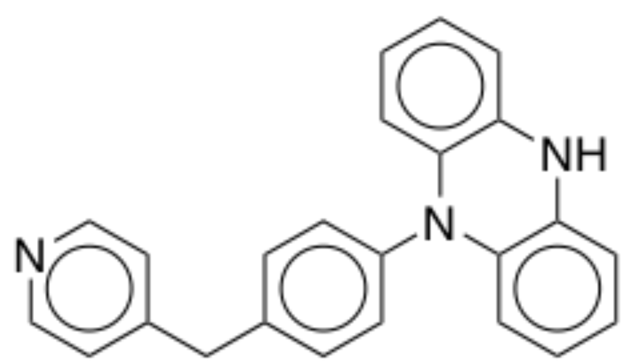
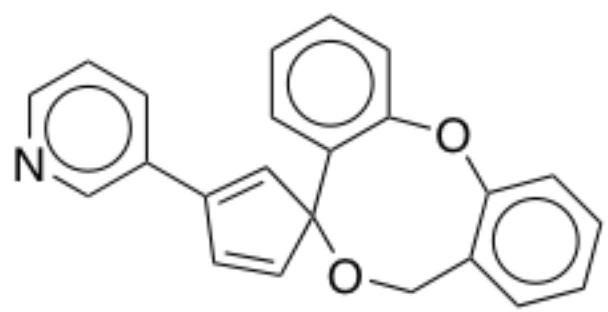
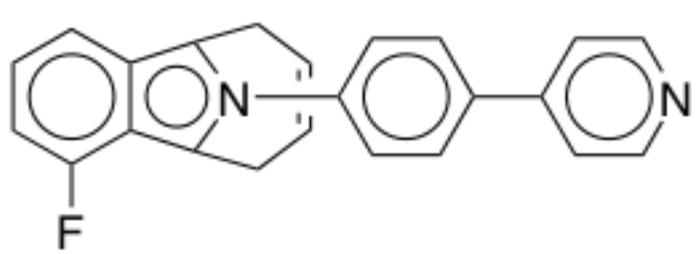
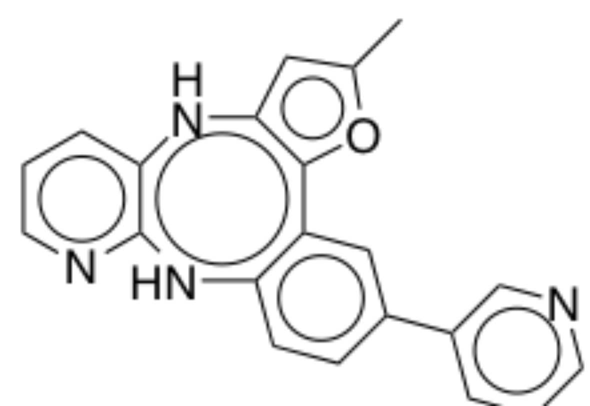
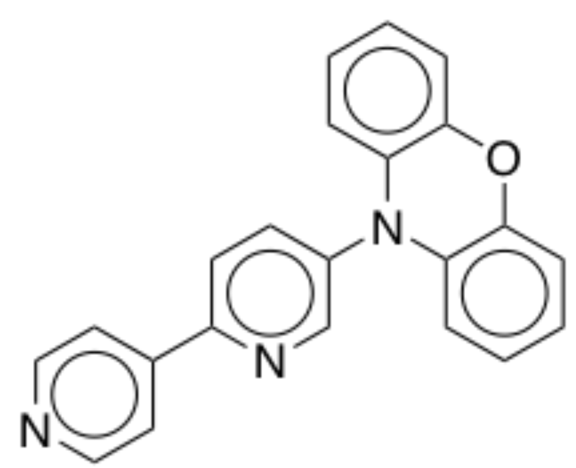
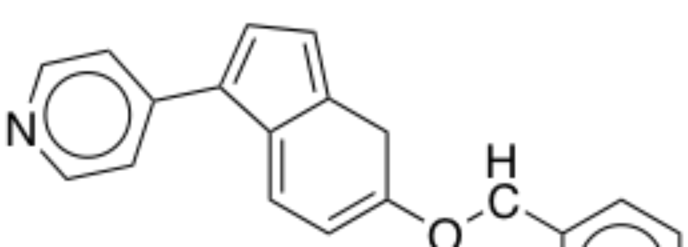
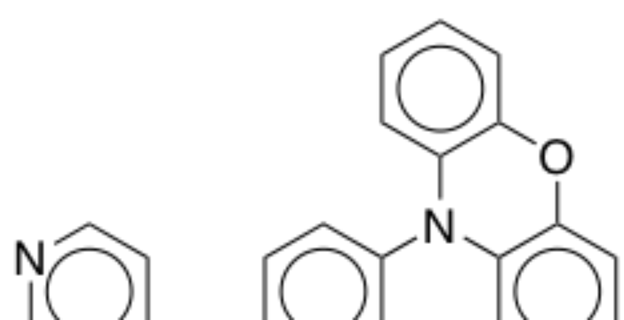
Glow: Generative Flow with Invertible 1x1 Convolutions, 2018
Diederik P. Kingma, Prafulla Dhariwal

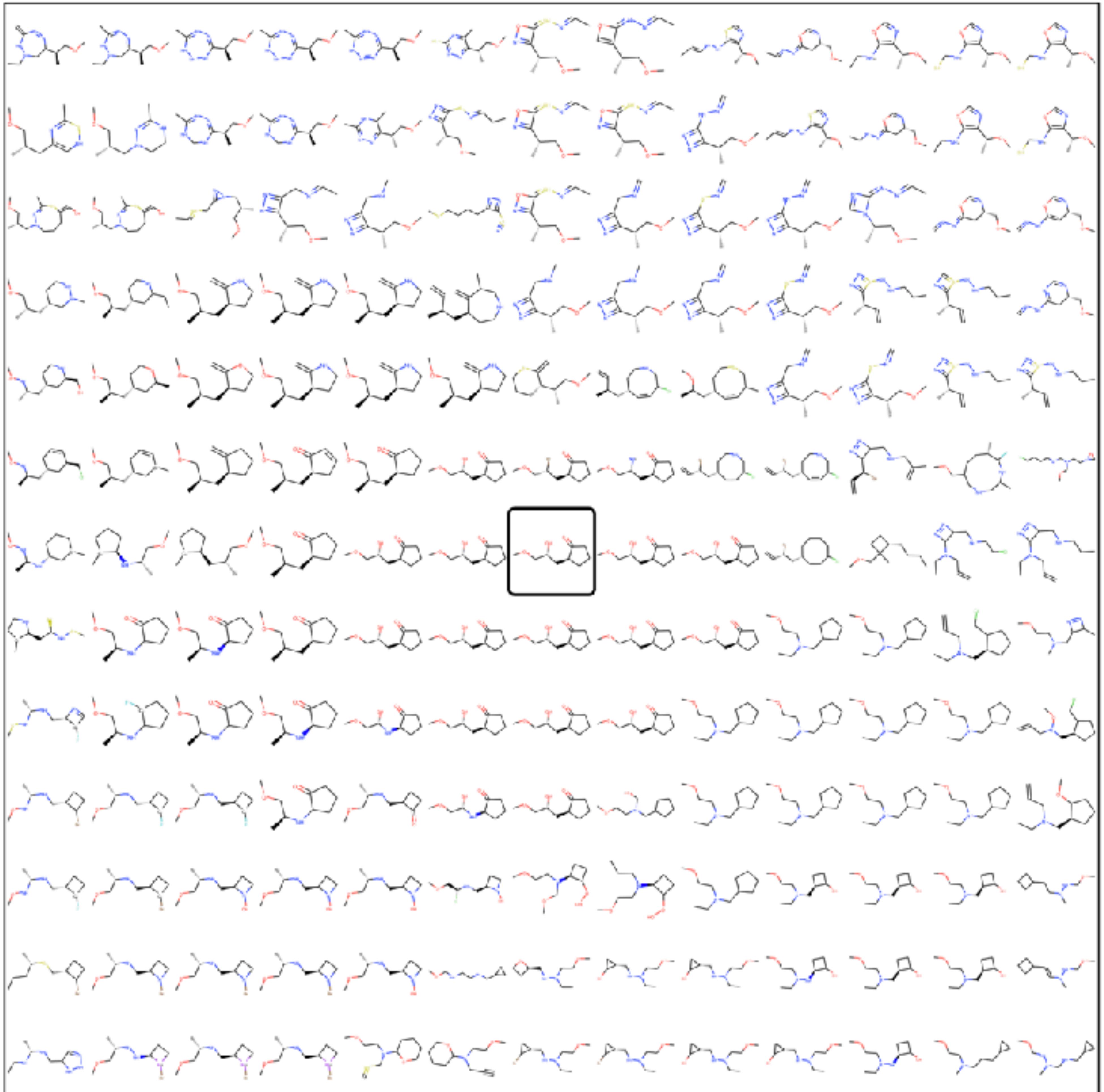
# Represent “Style” and “Content” Separately


A Neural Algorithm of Artistic Style, 2015
Leon A. Gatys, Alexander S. Ecker, Matthias Bethge

# Represent “Style” and “Content” Separately

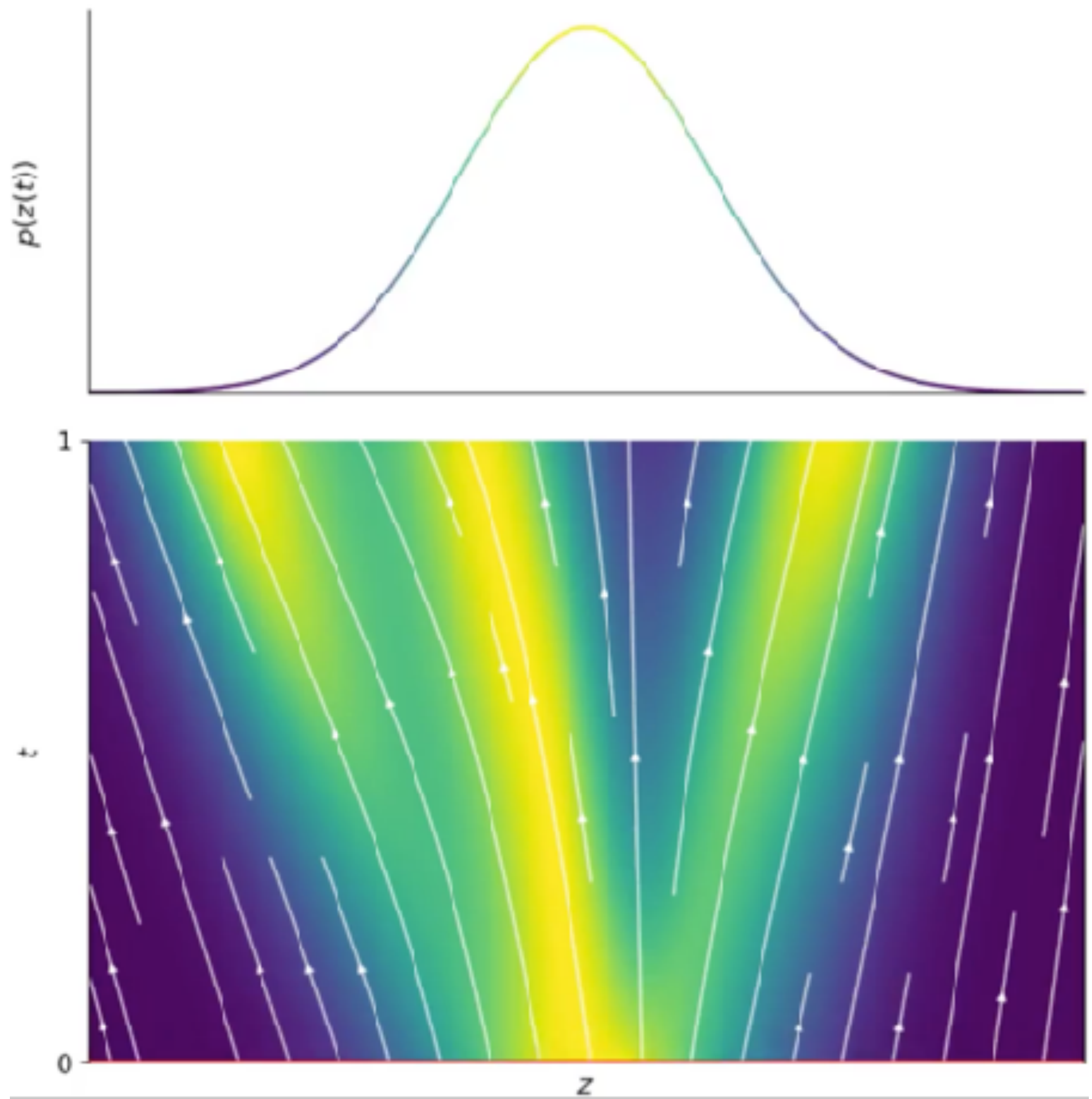

A Style-Based Generator Architecture for Generative Adversarial Networks, 2018
Tero Karras, Samuli Laine, Timo Aila



Grammar Variational Autoencoder (2017). Kusner, Paige, Hernández-Lobato

# Continuous Normalizing Flows

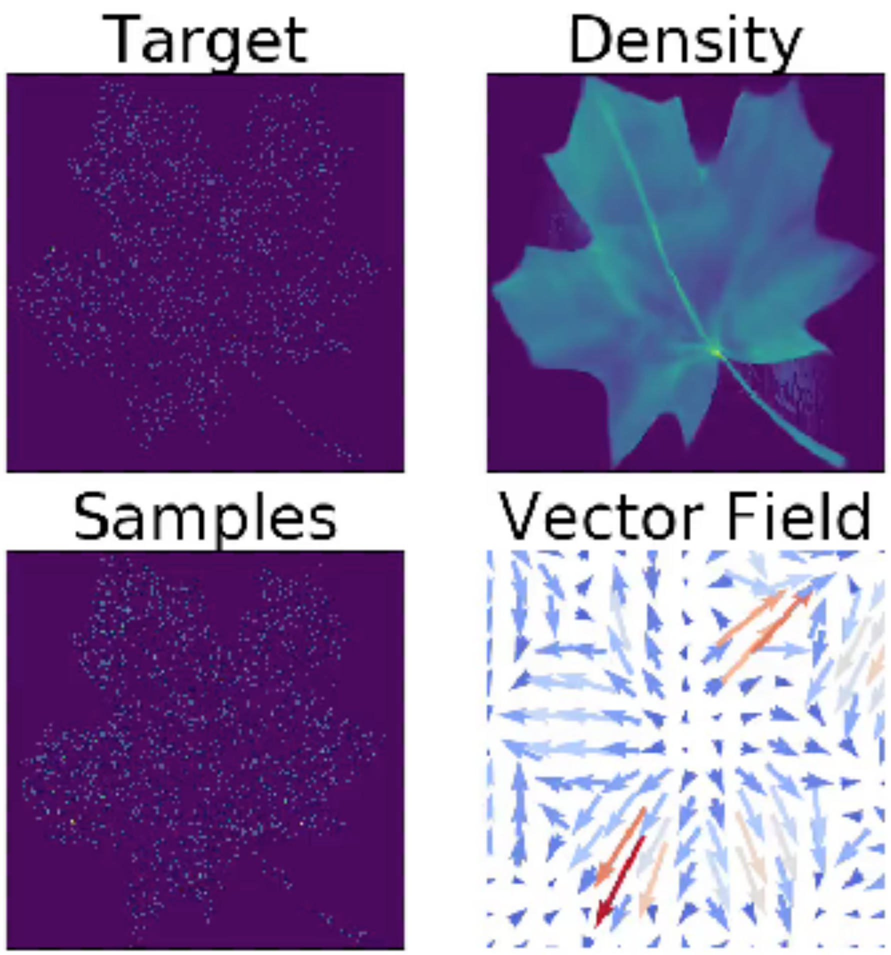



Continuously transform simple distribution into complex target

Neural Ordinary Differential Equations, 2018. Ricky T. Q. Chen*, Yulia Rubanova*, **Jesse Bettencourt***, David Duvenaud

FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models, 2018. Will Grathwohl*, Ricky T. Q. Chen*, **Jesse Bettencourt**, Ilya Sutskever, David Duvenaud

# Continuous Normalizing Flows

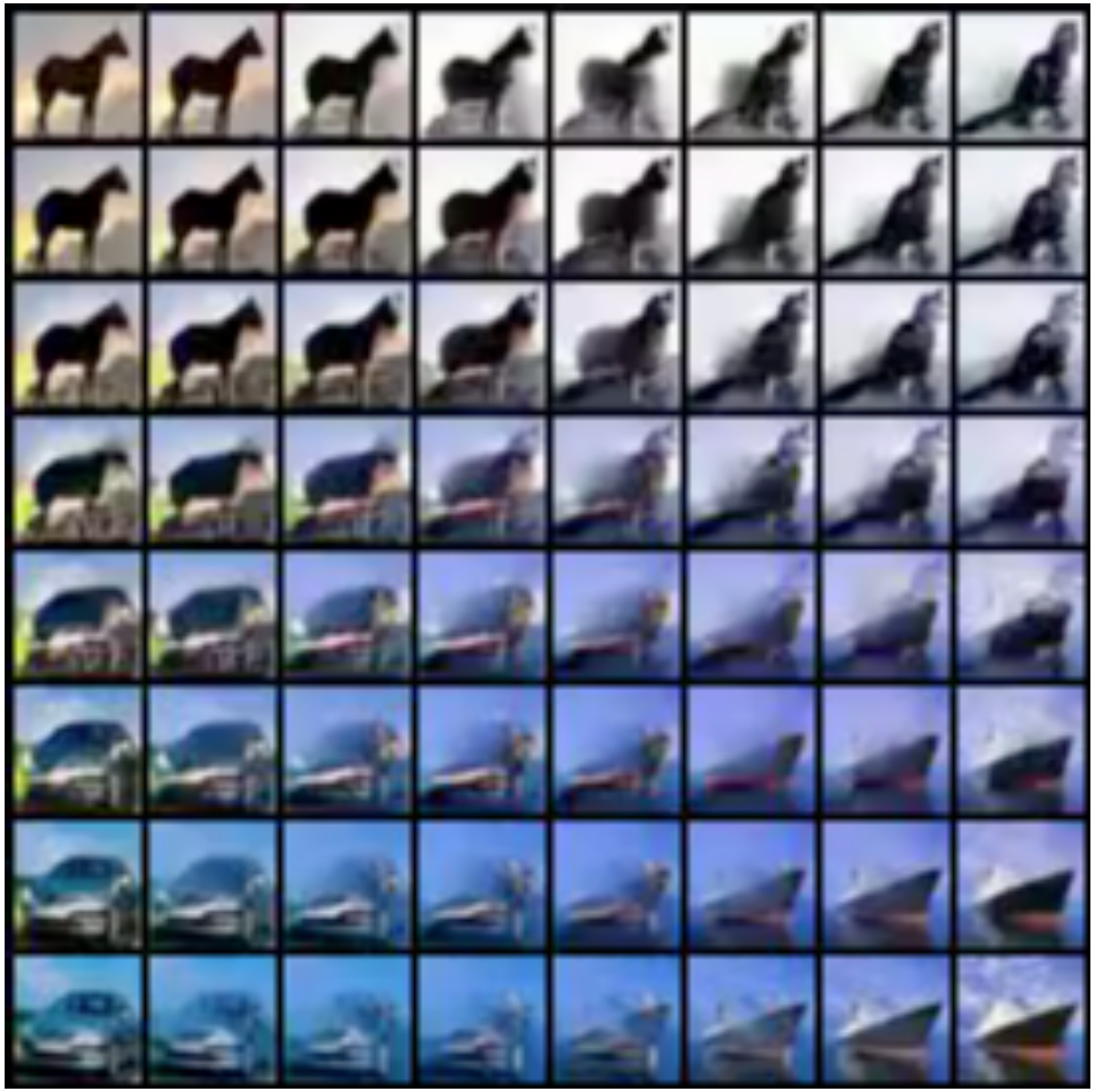



Continuously transform simple distribution into complex target

Neural Ordinary Differential Equations, 2018. Ricky T. Q. Chen*, Yulia Rubanova*, **Jesse Bettencourt***, David Duvenaud

FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models, 2018. Will Grathwohl*, Ricky T. Q. Chen*, **Jesse Bettencourt**, Ilya Sutskever, David Duvenaud

# Continuous Normalizing Flows



Continuously transform simple distribution into complex target

Neural Ordinary Differential Equations, 2018. Ricky T. Q. Chen*, Yulia Rubanova*, **Jesse Bettencourt***, David Duvenaud

FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models, 2018. Will Grathwohl*, Ricky T. Q. Chen*, **Jesse Bettencourt**, Ilya Sutskever, David Duvenaud

# 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: | Extensible family: |
| --- | --- |
| • K-means | • Mixture of Gaussians |
| • Kernel Density Estimation | • Latent variable models |
| • SVMs | • Gaussian processes |
| • Boosting | • Deep neural nets |
| • Random Forests | • Bayesian neural nets |
| • K-Nearest Neighbours | • ?? |

# Regularization as a bag of tricks

Fast special cases:

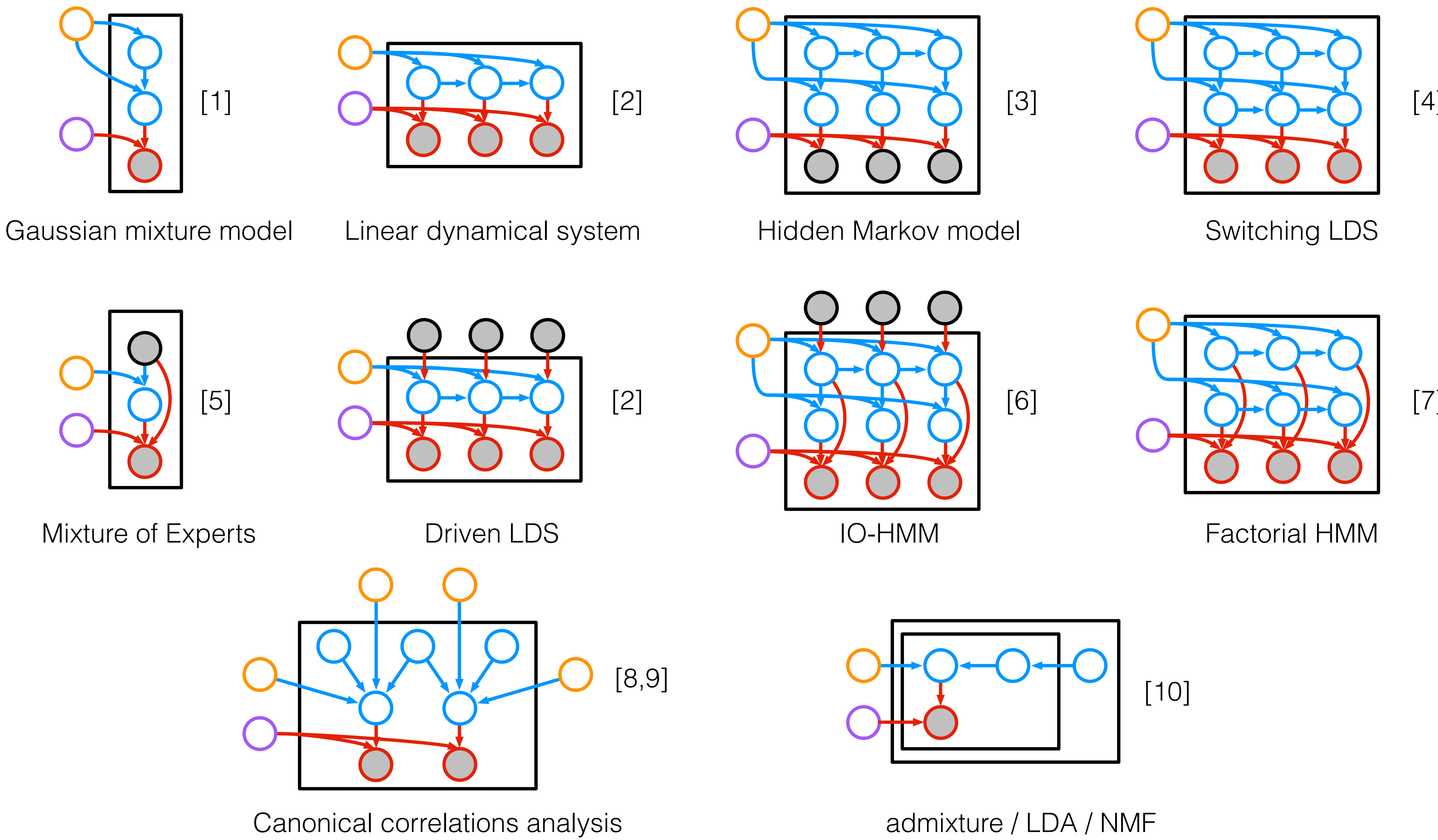
- Early stopping
- Ensembling
- L2 Regularization
- Gradient noise
- Dropout
- Expectation-Maximization

Extensible family:

- Stochastic variational inference

# A language of models

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





[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

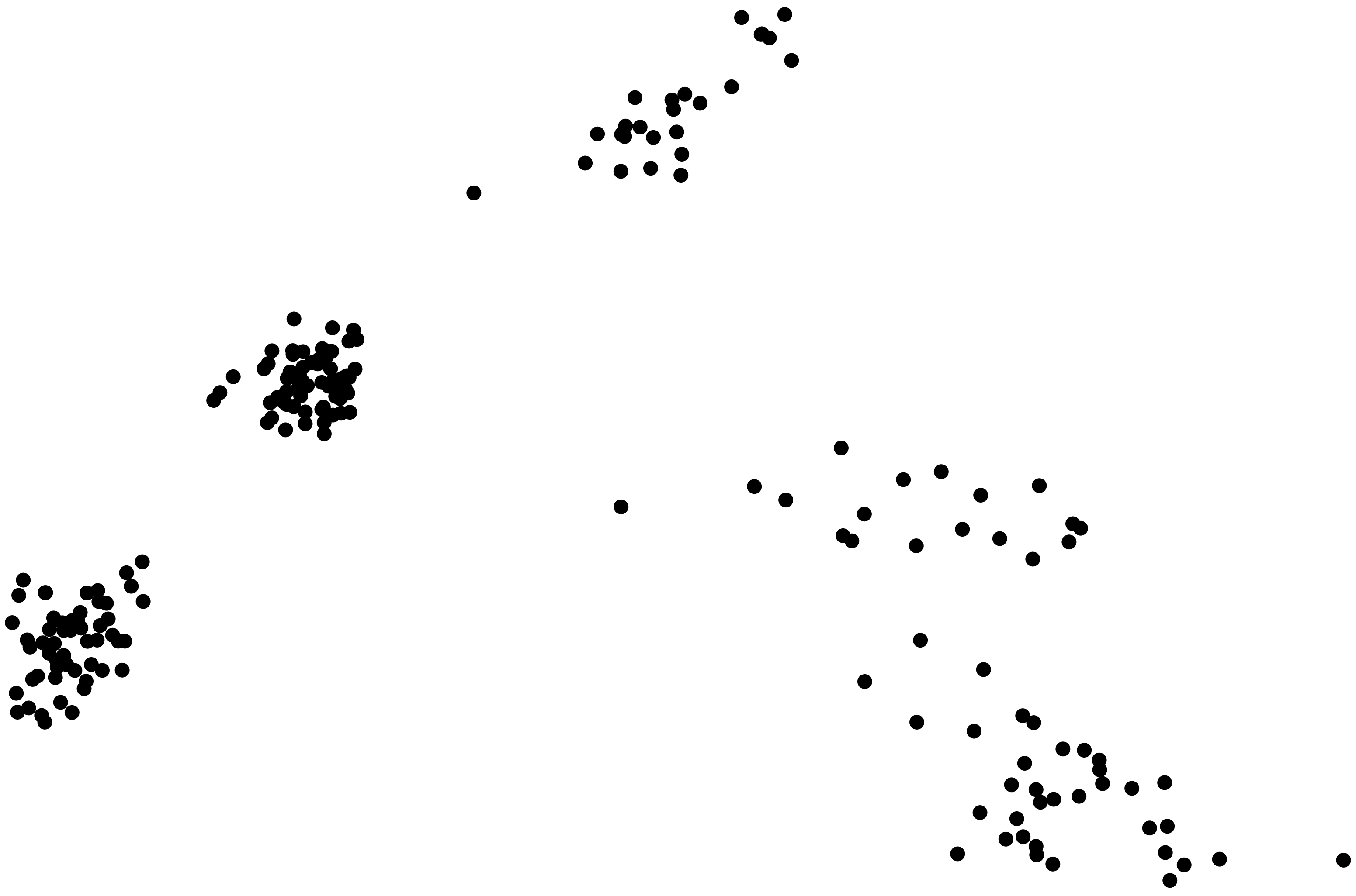
Extensible family:

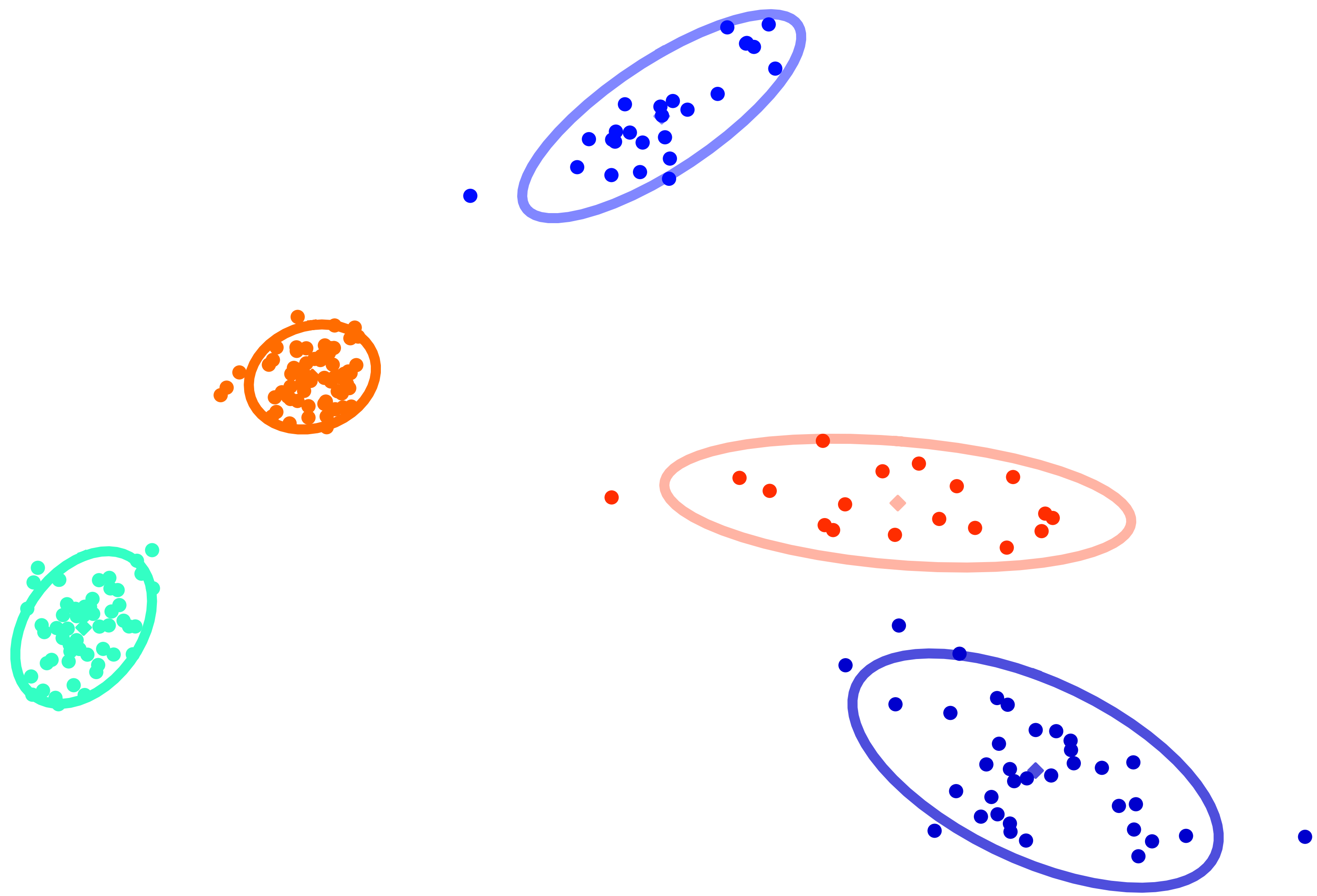
- Deep probabilistic latent-variable models + decision theory

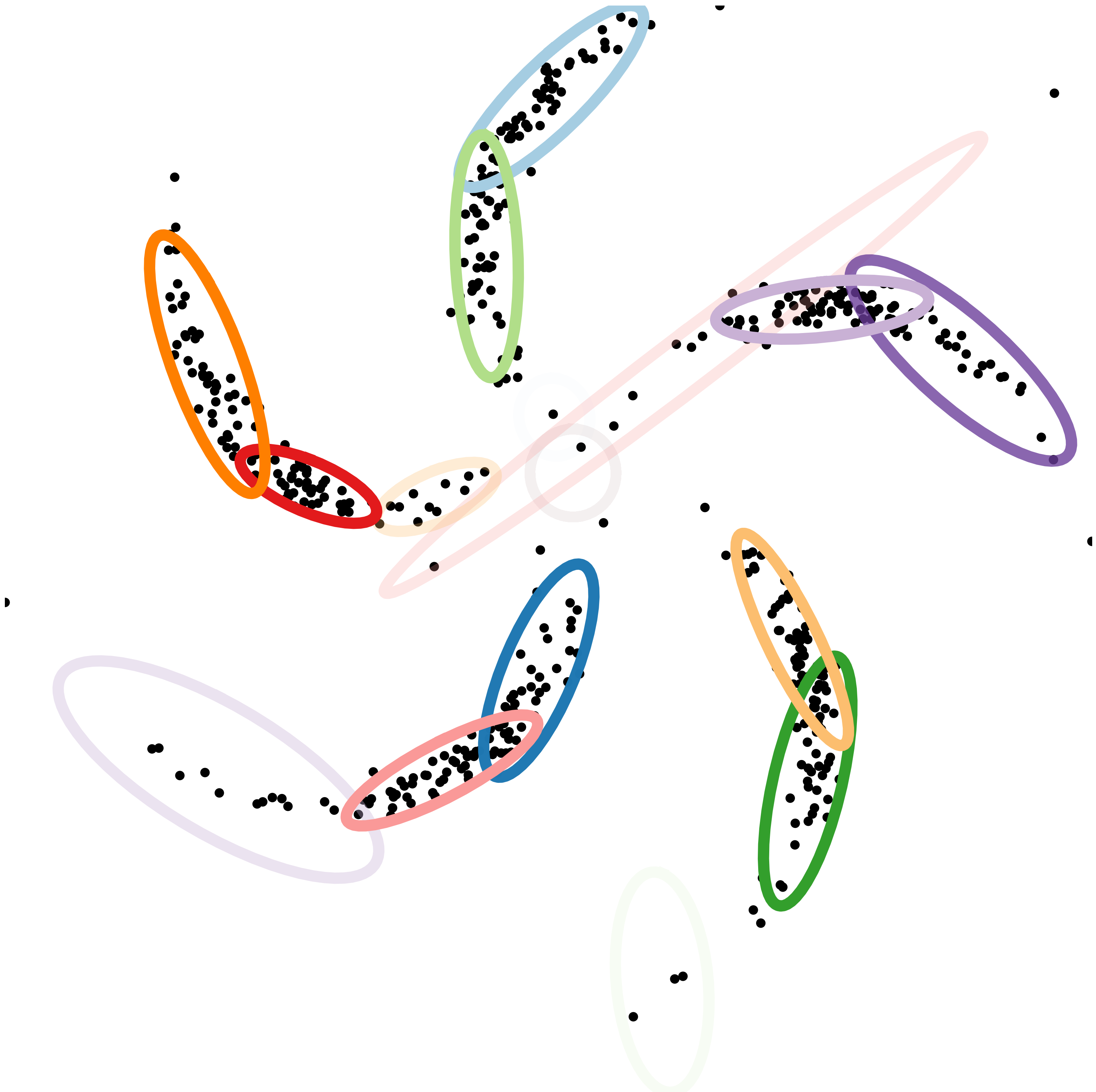
# Advantages of probabilistic latent-variable models

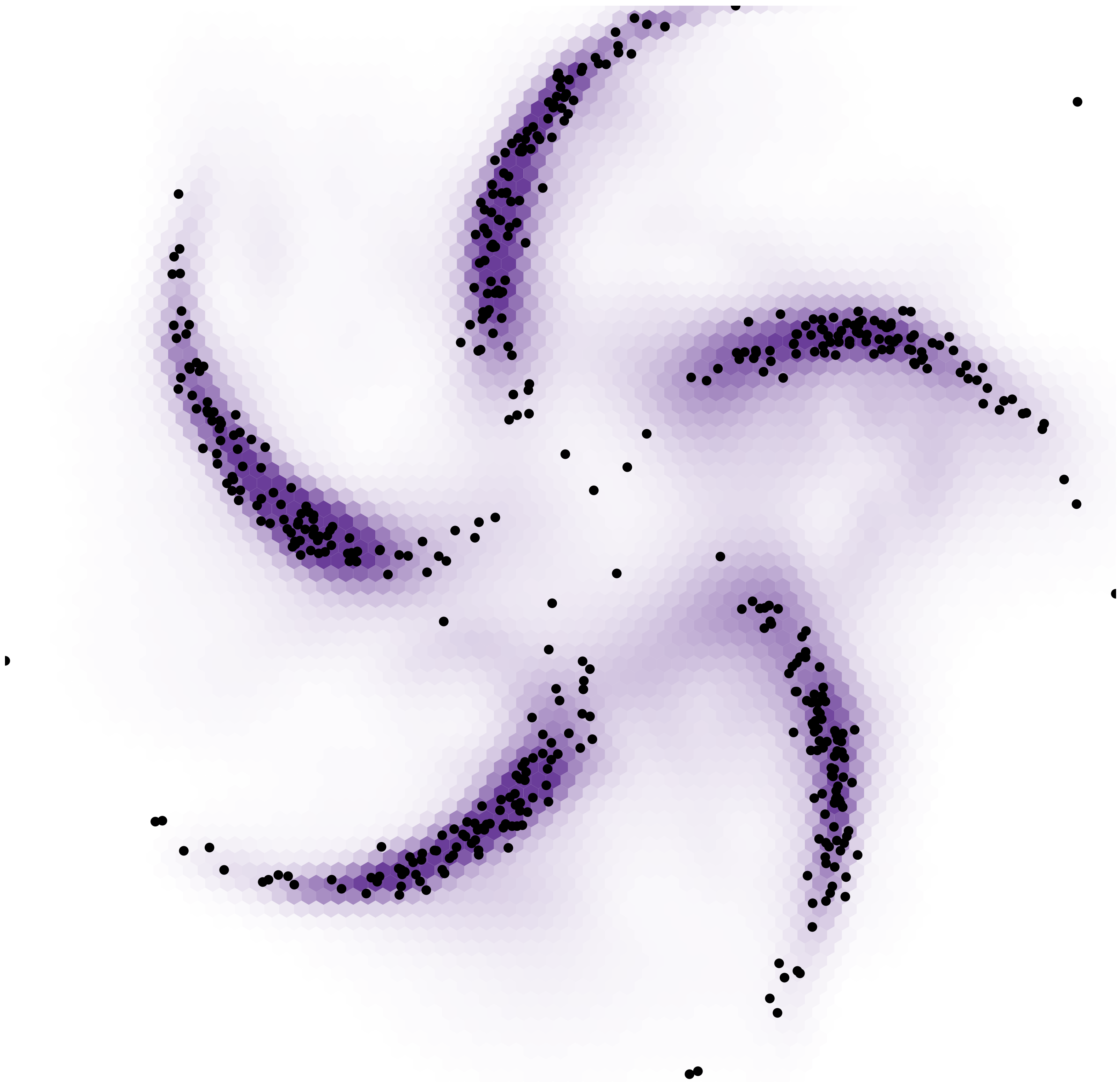
- **Data-efficient Learning** - automatic regularization, can take advantage of more information
- **Compose-able Models** - e.g. incorporate data corruption model. Different from composing feedforward computations
- **Handle Missing + Corrupted Data** (without the standard hack of just guessing the 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

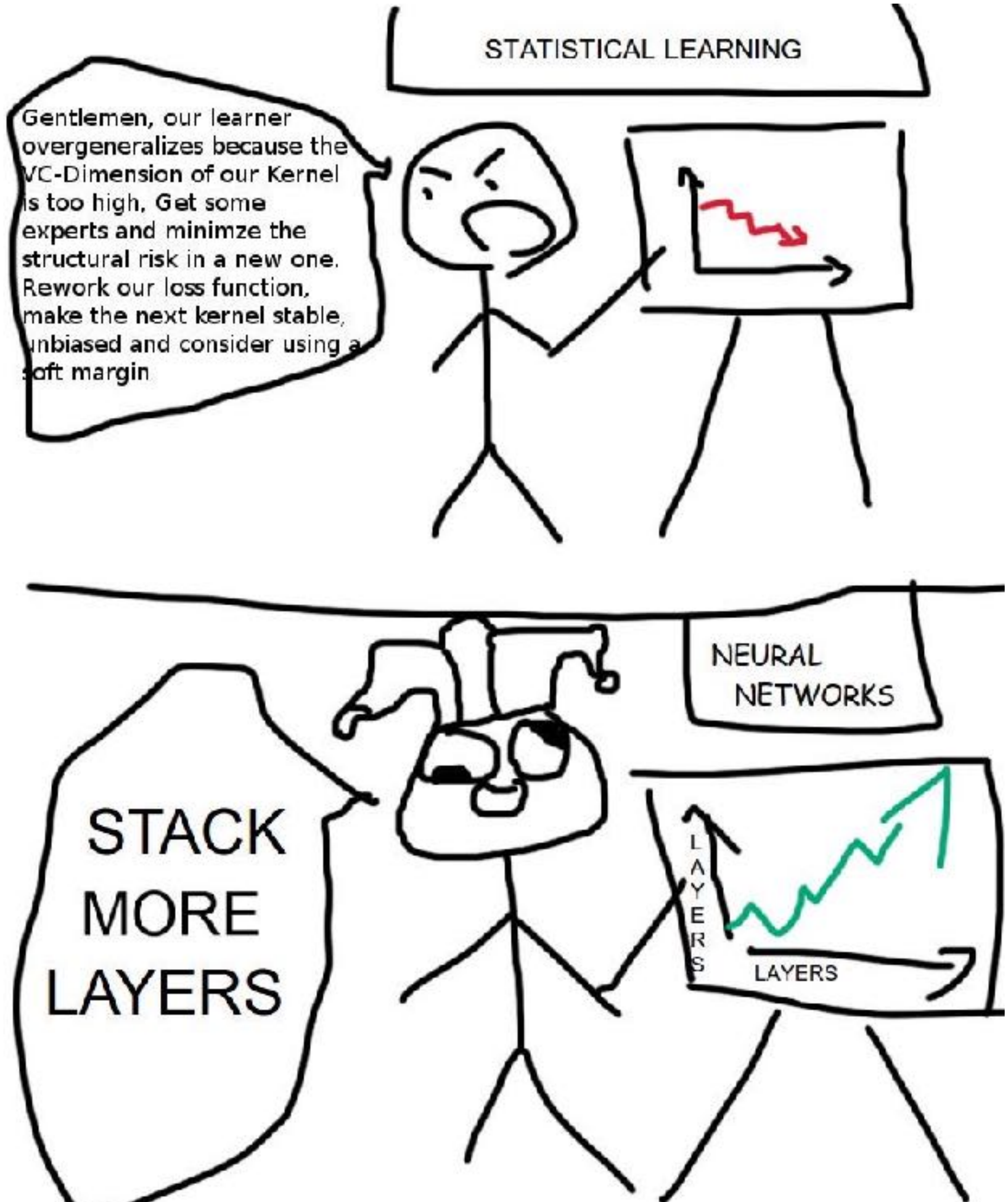








| Probabilistic graphical models | Deep learning |
| --- | --- |
| + structured representations | – neural net “goo” |
| + priors and uncertainty | – difficult parameterization |
| + data and computational efficiency | – can require lots of data |
| – rigid assumptions may not fit | + flexible |
| –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


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

# Differentiable models

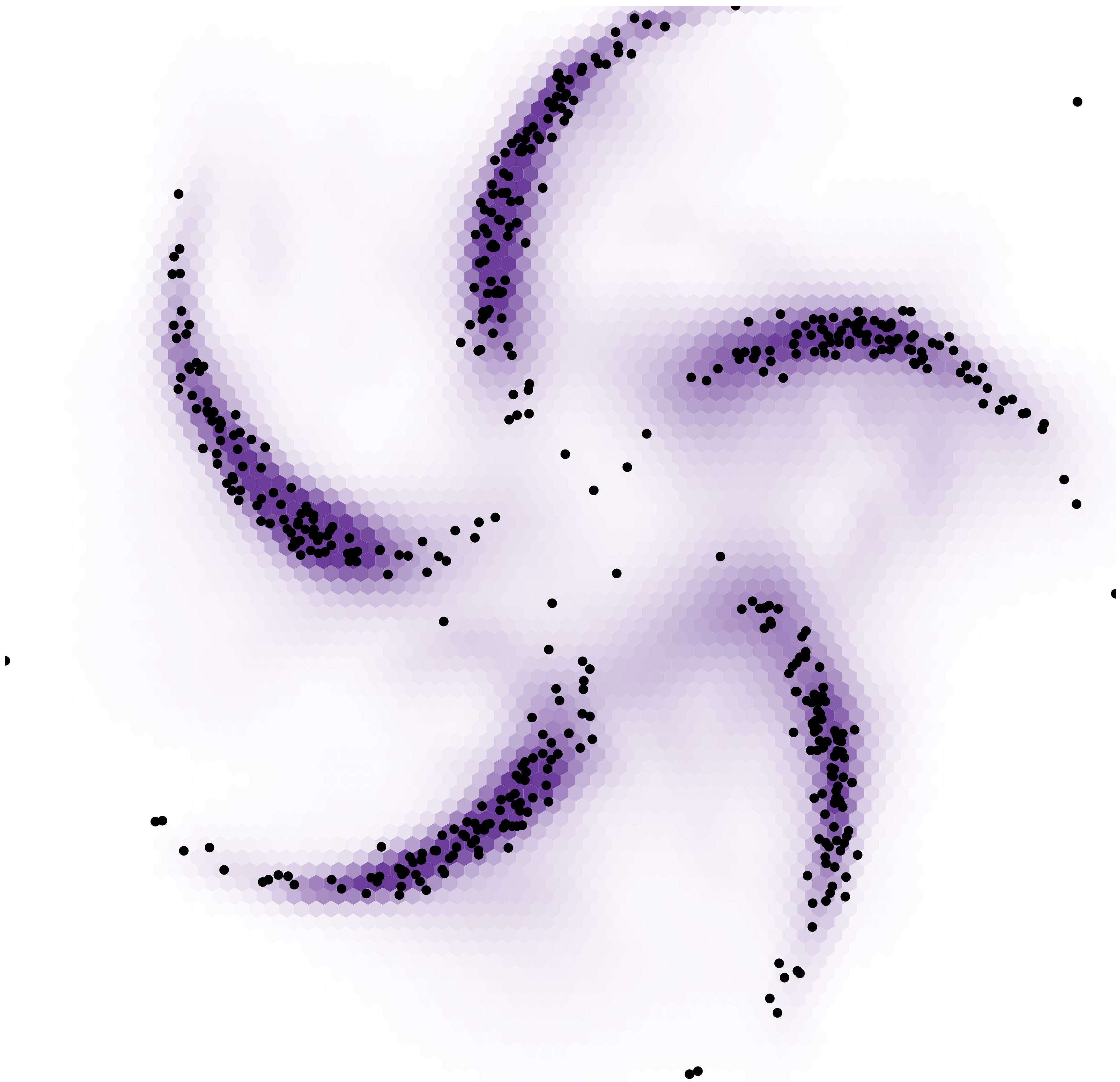
- Model distributions implicitly by a variable pushed through a deep net:

$$y = f_\theta(x) \qquad x \sim \mathcal{N}(0, \mathbf{I})$$

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

$$p(y|x) = \mathcal{N}(y|\mu = f_\theta(x), \Sigma = g_\theta(x)) \qquad x \sim \mathcal{N}(0, \mathbf{I})$$

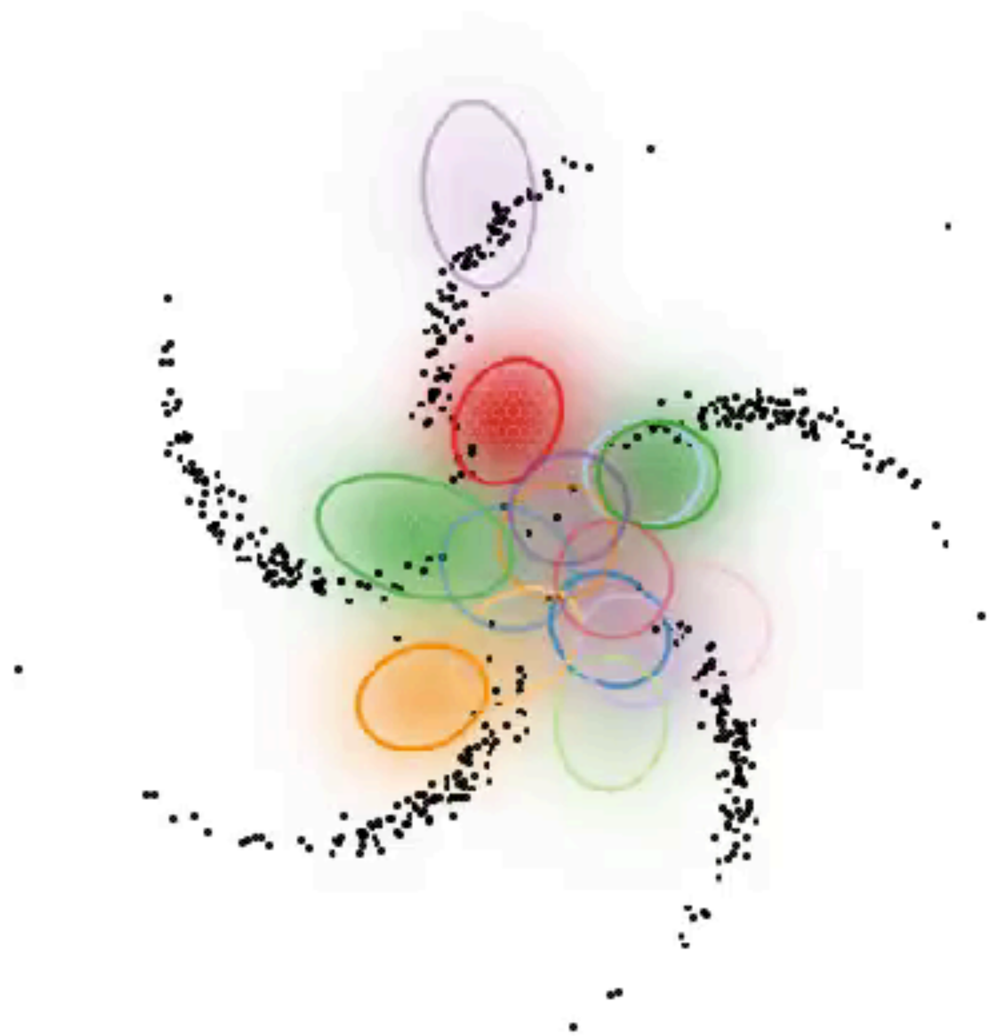
- Optimize all parameters using stochastic gradient descent



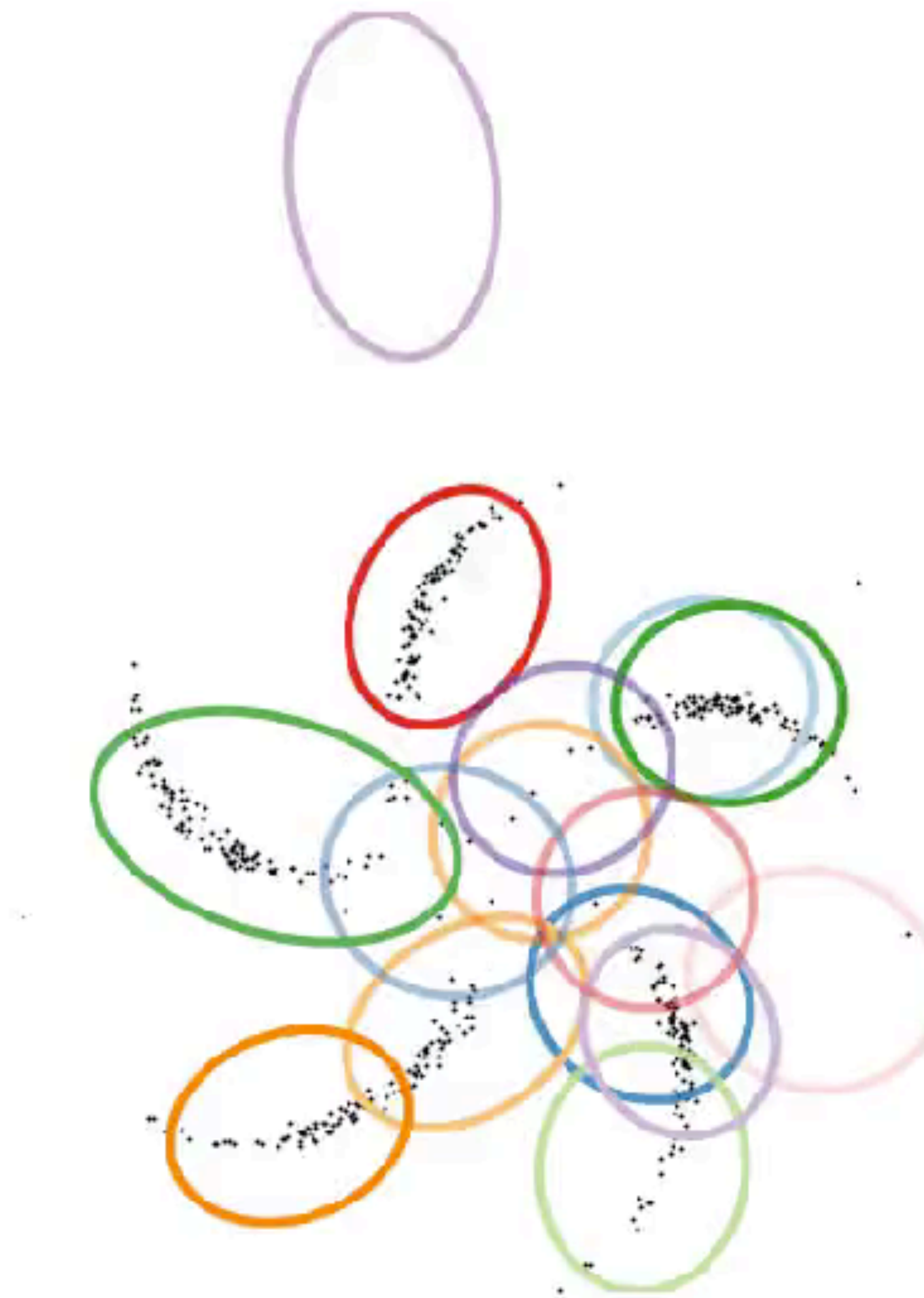
# Compose Probabilistic Graphical Models with Neural Networks

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

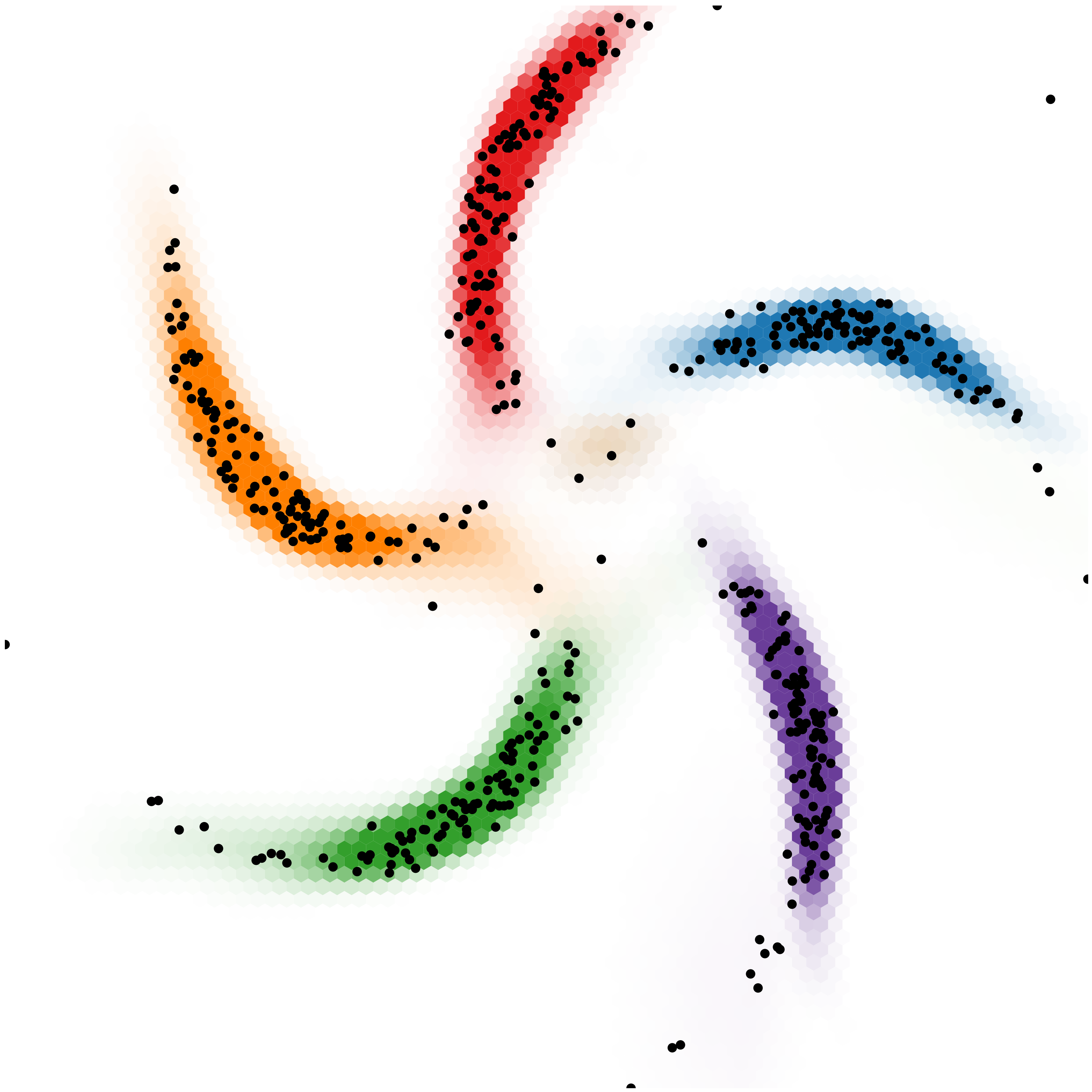
Composing graphical models with neural networks for structured representations and fast inference. Johnson, Duvenaud, Wiltschko, Datta, Adams, NIPS 2016



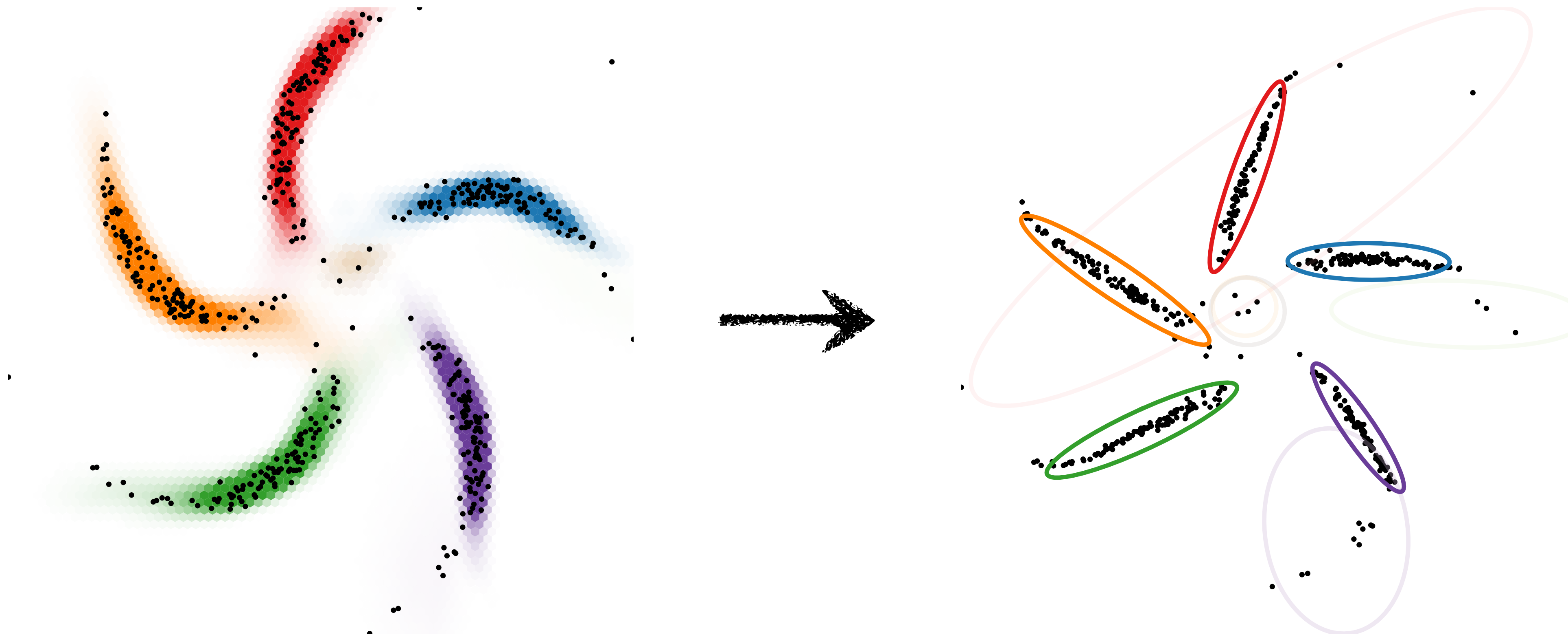
data space



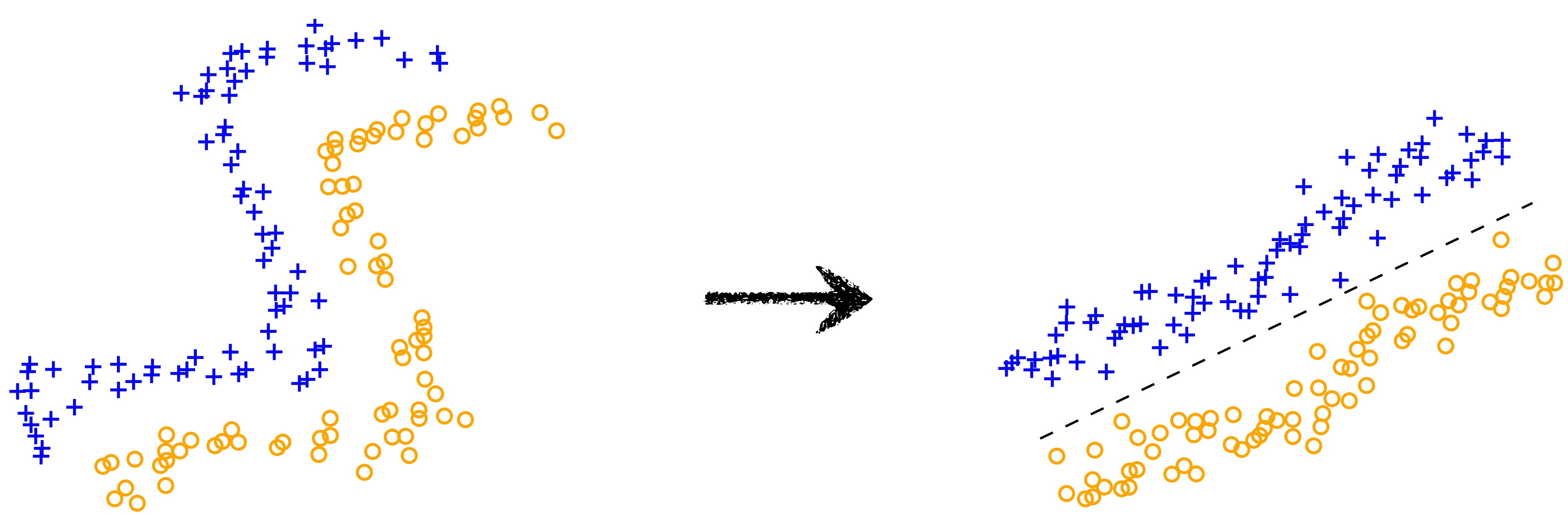
latent space



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 (bag of bricks: 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 and Neural Networks
- Graphical model notation and exact inference
- Mixture Models, Bayesian Networks
- Model Comparison and marginal likelihood
- Stochastic Variational Inference
- Time series and recurrent models
- Gaussian processes
- Variational Autoencoders
- Generative Adversarial Networks
- Normalizing Flows?

# 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)