GitHub: /dkahle/2019-MPI

Stochastic Exploration of Real Varieties and Applications

David Kahle

In collaboration with Jon Hauenstein and Jerry Ma

Baylor University
COLLEGE OF ARTS & SCIENCES
Statistical Science

Overview

1. Motivation

2. Variety distributions

3. Sampling and implementation

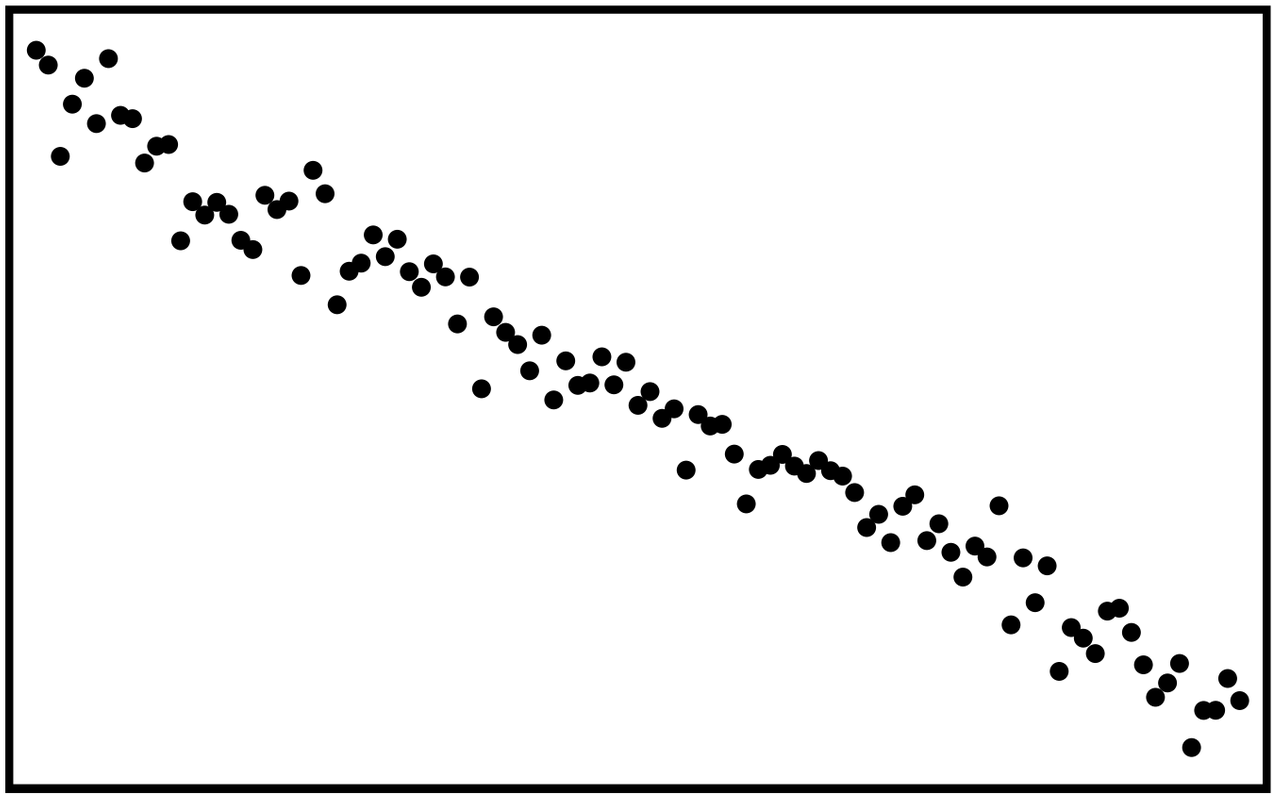
4. Examples

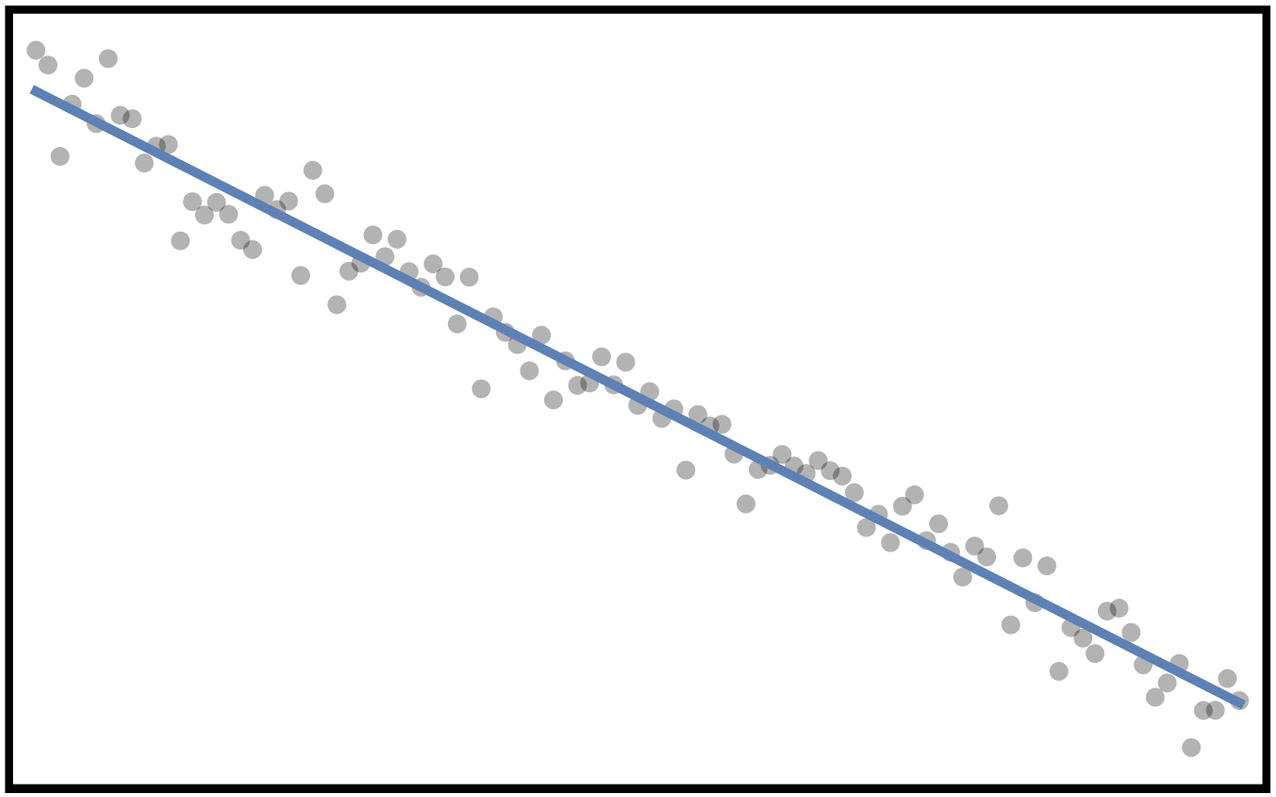
5. Other thoughts

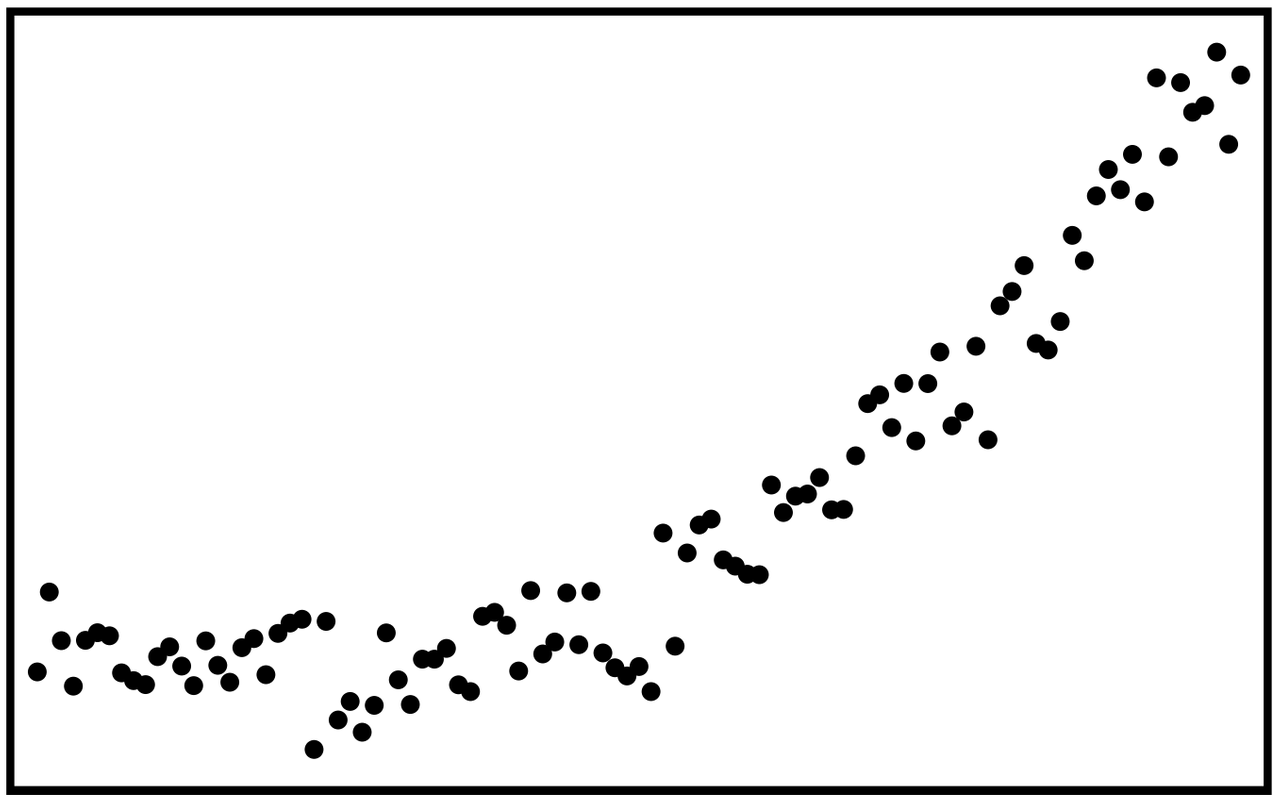
Motivation

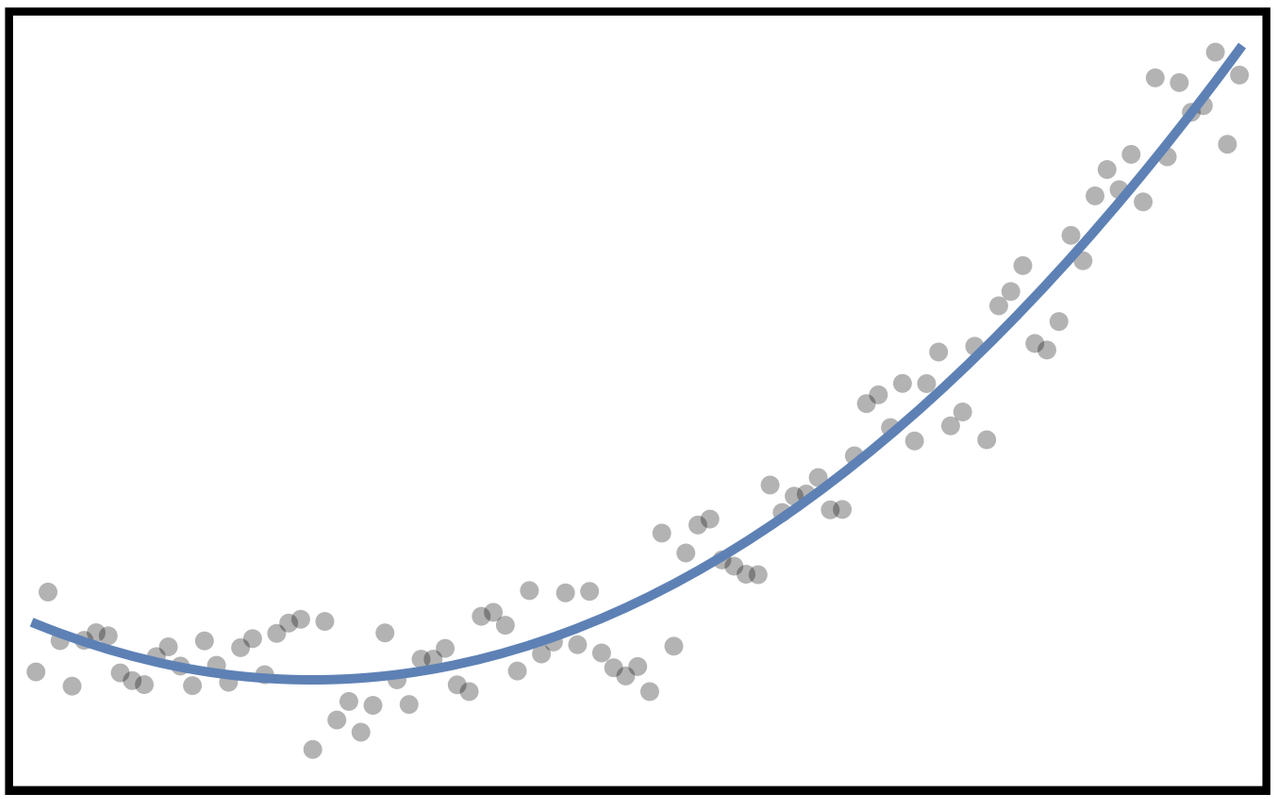
Motivation

Baylor University



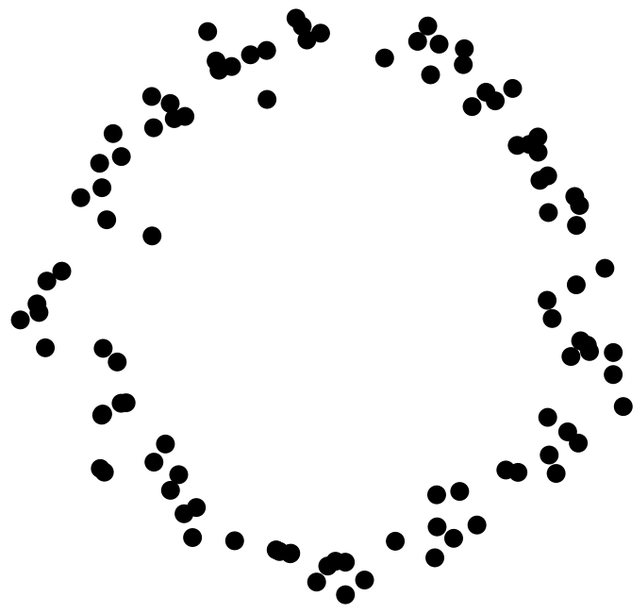


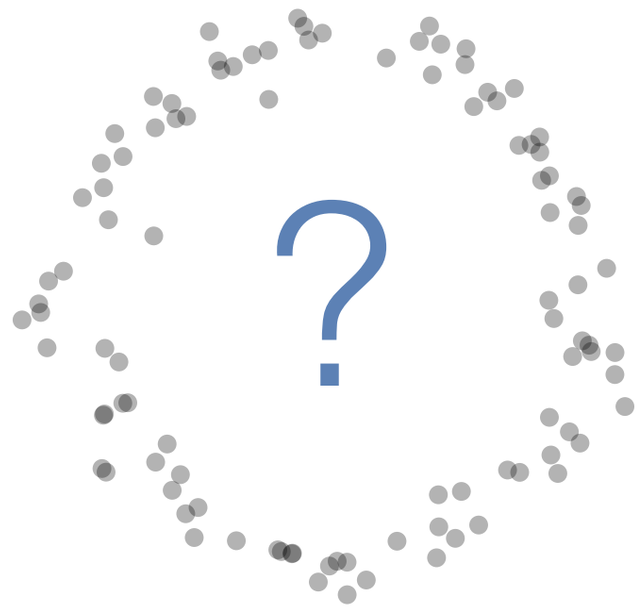


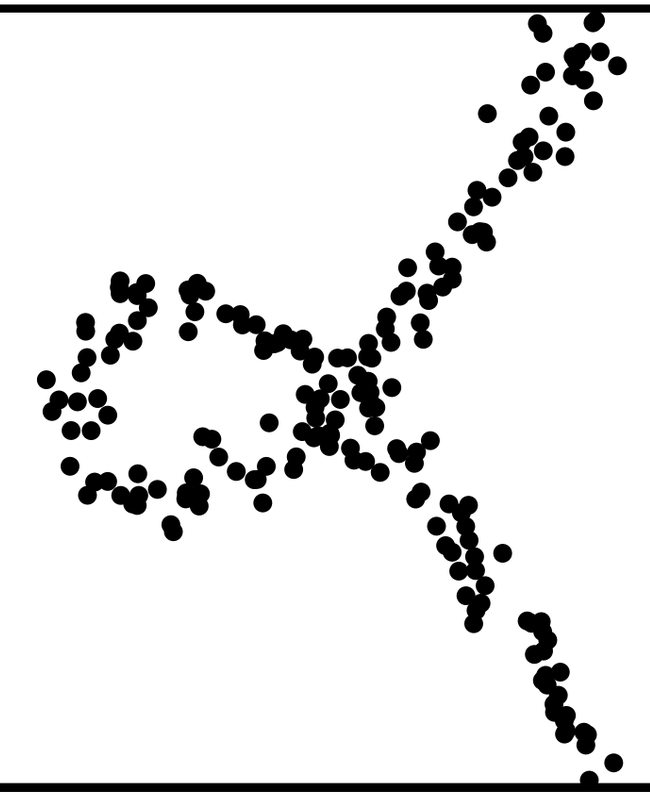


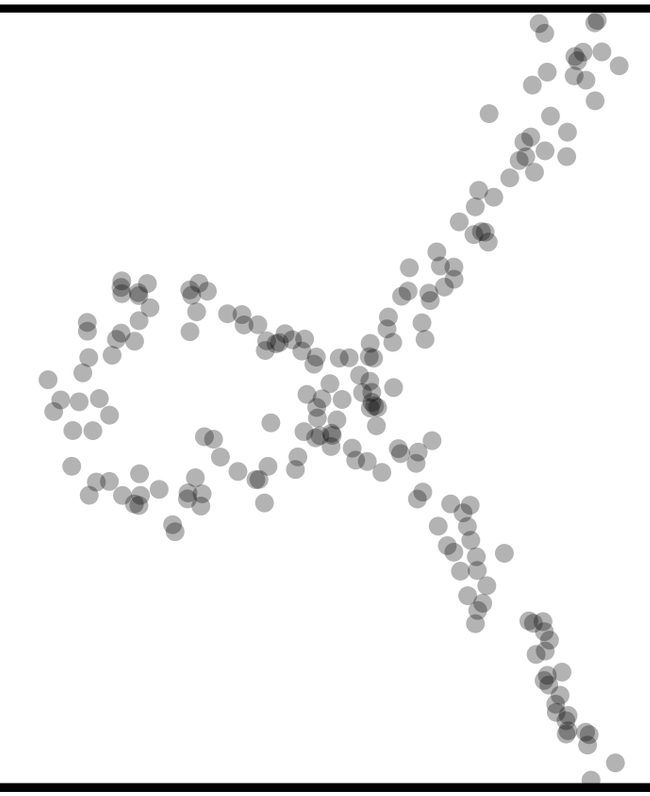
Motivation

Baylor University

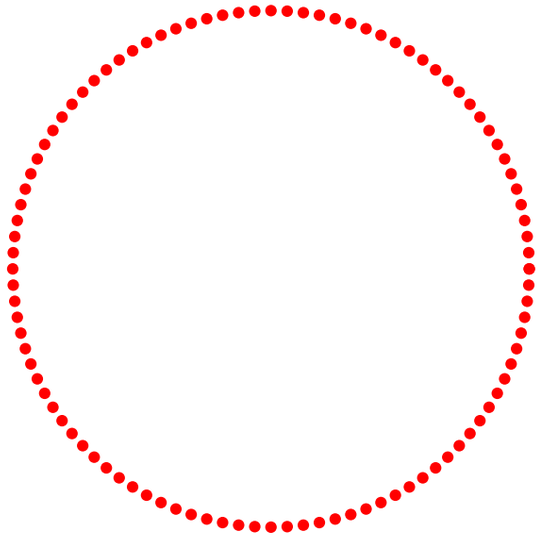


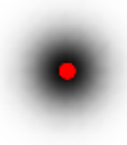




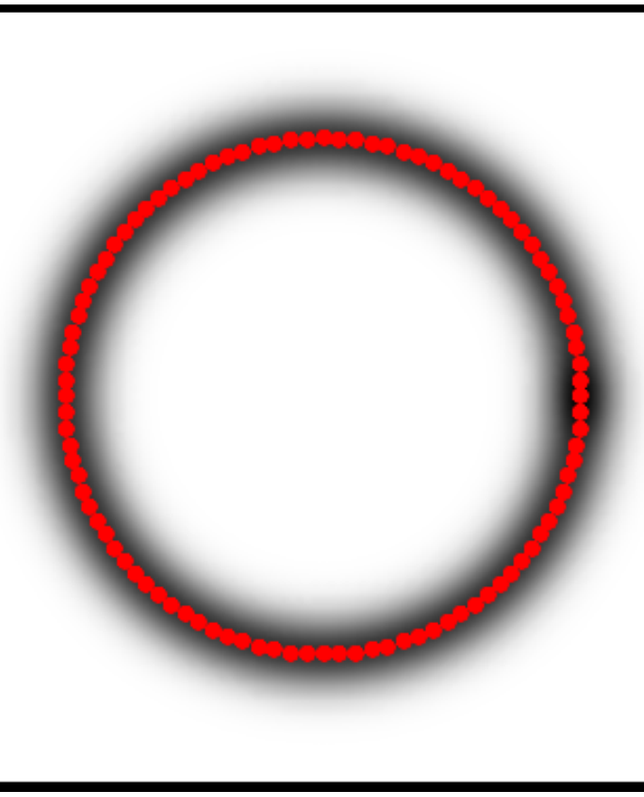


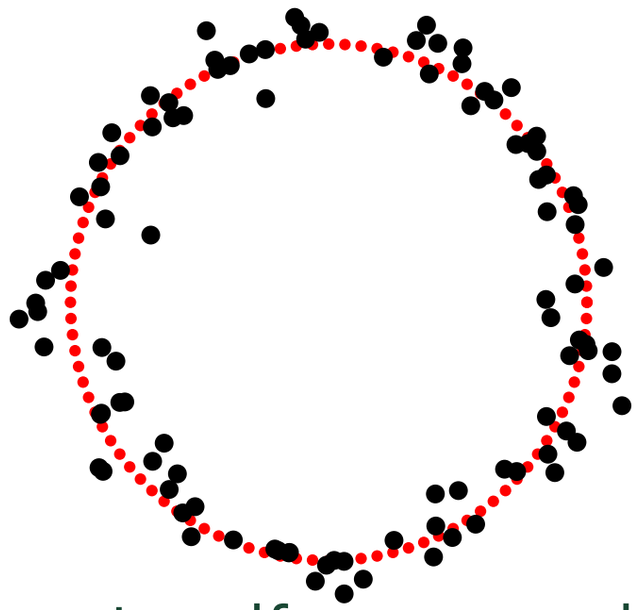
Motivation





Add bivariate normal noise





Careful: not uniform on circle!

Motivation

Problems for pattern recognition:

Limiting – only varieties with known parameterizations

No stochastic structure – distribution of estimators? etc.

General problem – how to sample near varieties

Applications: algebraic pattern recognition, TDA, solving nonlinear systems, optimization, gridding, …

Strategy for stochastically exploring real varieties

Create a distribution with mass near the variety of interest

Sample from the distribution

Magnetize the sampled points onto the variety with endgames

Variety distributions

Recasting the normal distribution

Baylor University

The normal density is

Partition function, normalizing constant dependent on parameters

$$p(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

μ is the mean; the center of the bell curve

σ is the standard deviation; governs dispersion about μ

Empirical rule –

68% of distribution within ±σ of μ

95% of distribution within ±2σ of μ

99.7% of distribution within ±3σ of μ

Recasting the normal distribution

The normal density is

$$p(x|\mu,\sigma) \propto \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

Probability mass concentrates near root of polynomial

$$g(x) = g(x|\mu) = x - \mu \in \mathbb{R}[x]$$

Same is true for arbitrary polynomials

exp{ $-g^2$ } is largest on the variety, where it has value 1

Decays exponentially as you move away from variety

Variety normal distribution provisional

A random vector $\boldsymbol{X}$ has the variety normal distribution if

$$p(\boldsymbol{x}|g,\sigma) \propto \exp\left\{-\frac{g(\boldsymbol{x}|\boldsymbol{\beta})^2}{2\sigma^2}\right\}$$

with $g(\boldsymbol{x}|\boldsymbol{\beta}) \in \mathbb{R}[\boldsymbol{x}]$

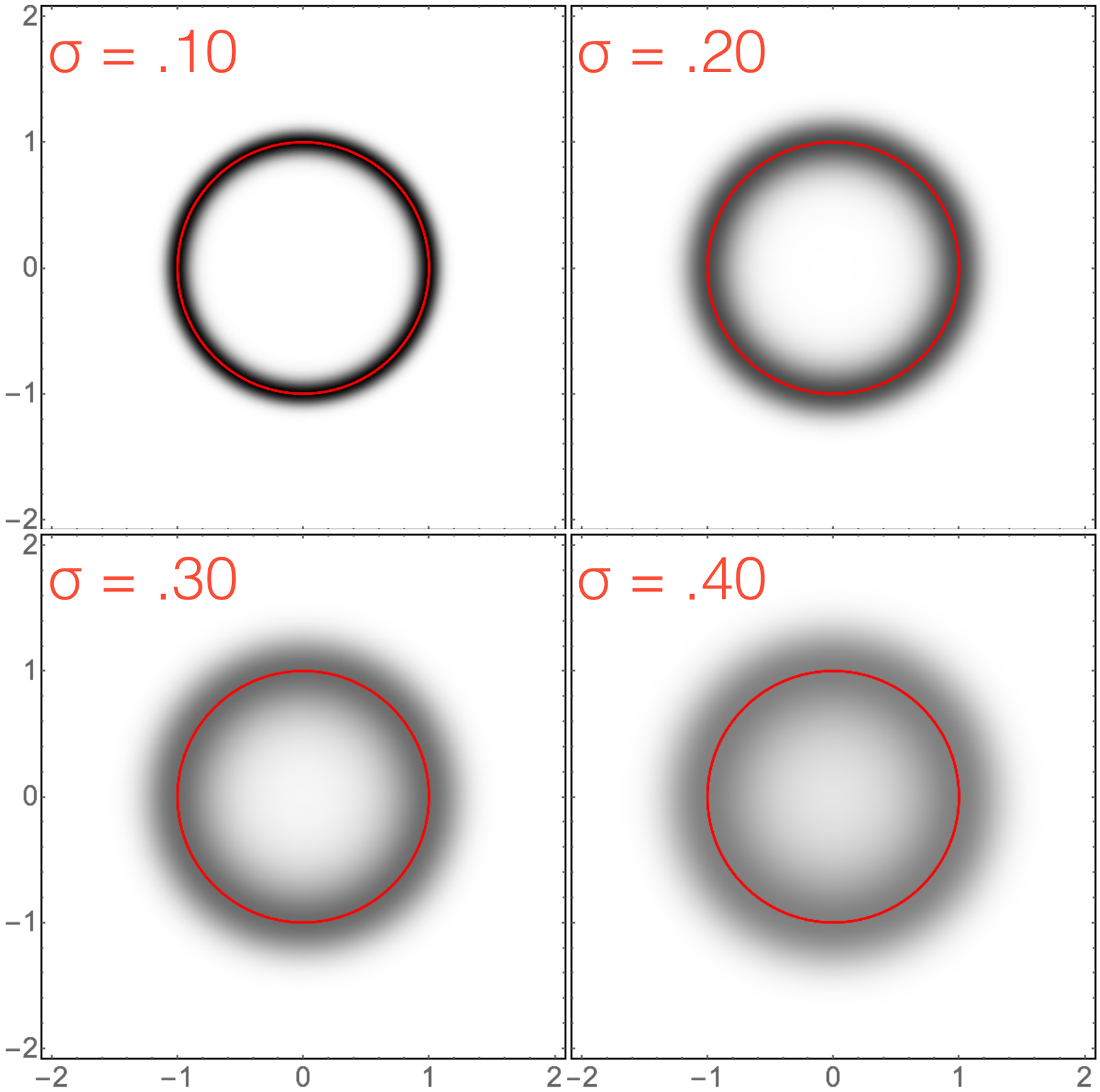
g is "given" : $\boldsymbol{\beta}$ is known and the polynomial form is specified

Example. $\boldsymbol{X} = (X\ Y)' \sim \mathcal{N}_2\left(x^2+y^2-1, \sigma\right)$

$$p(x,y|g,\sigma) \propto \exp\left\{-\frac{(x^2+y^2-1)^2}{2\sigma^2}\right\}$$

Variety normal distribution provisional

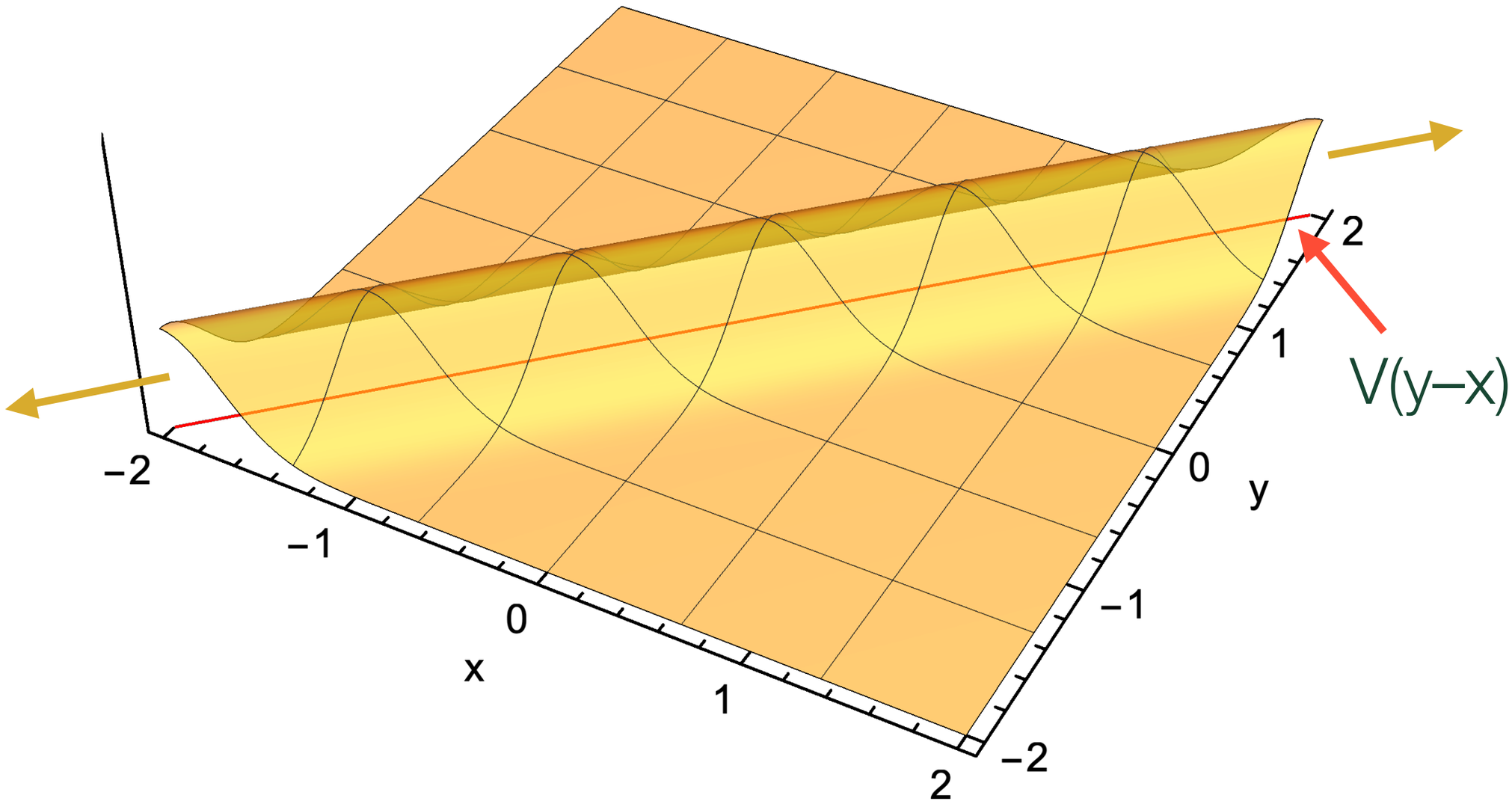
Baylor University



1. Non-compact varieties

If the variety is unbounded, then it obviously can't be normalized

Example: g(x, y) = y – x



Variety normal distribution provisional

1. Non-compact varieties

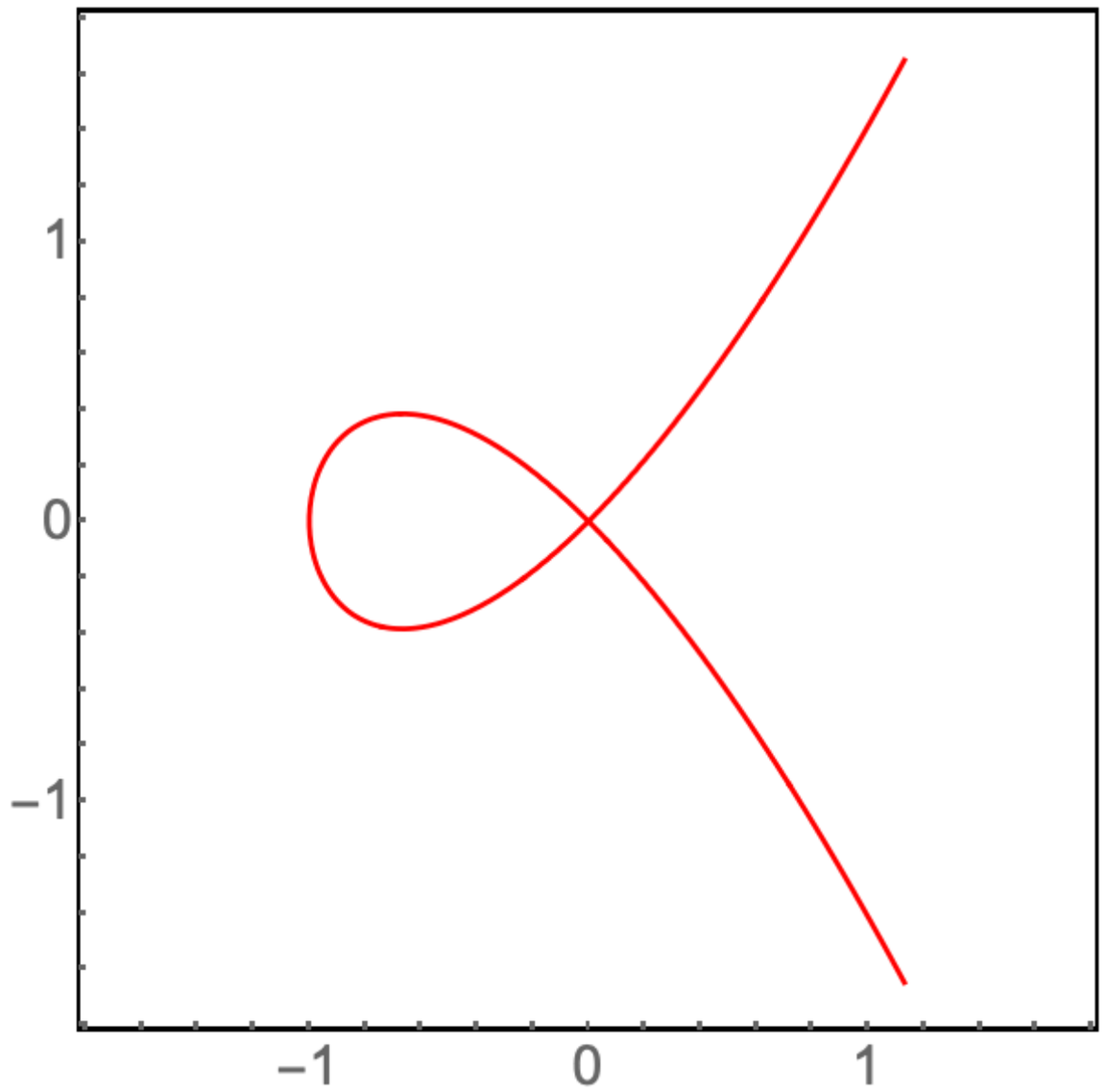
Solution: Truncate or taper

2. σ does not gauge variability globally

2. σ does not gauge variability globally

Probability mass does not decay evenly across variety

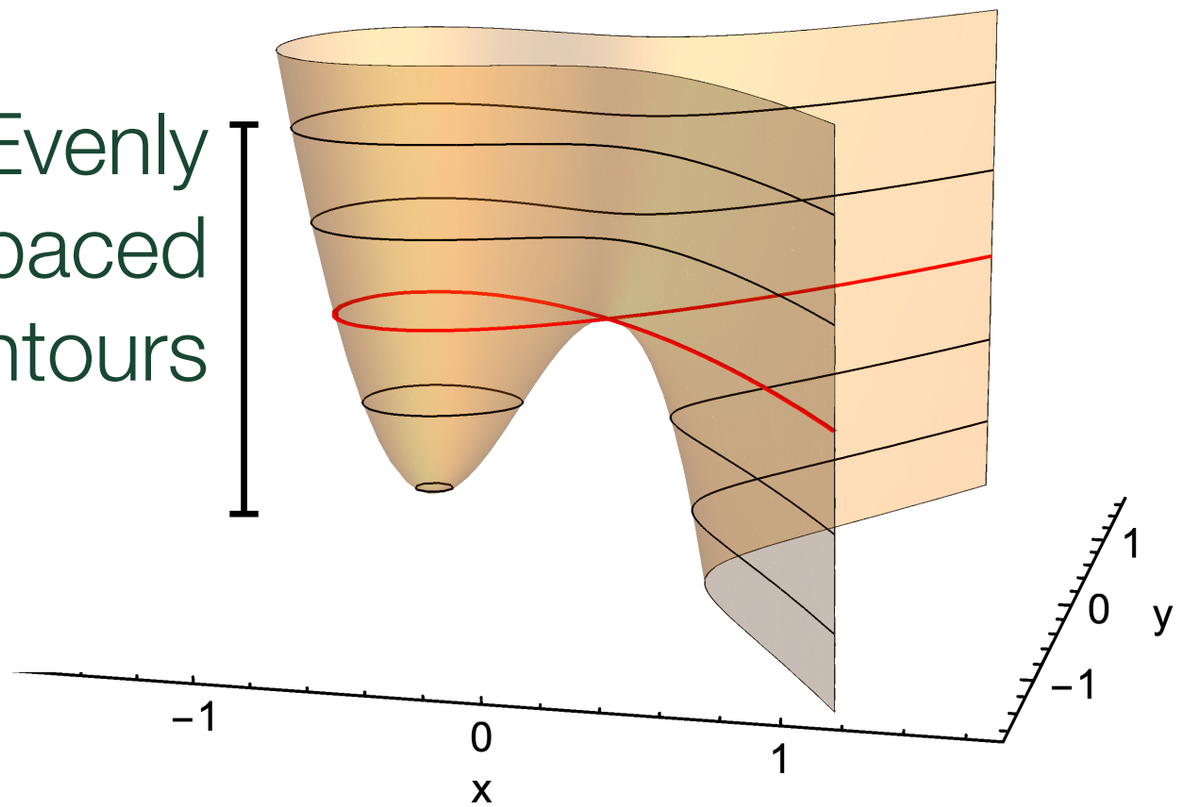
Example: Alpha curve, $V(y^2 - (x^3 + x^2))$

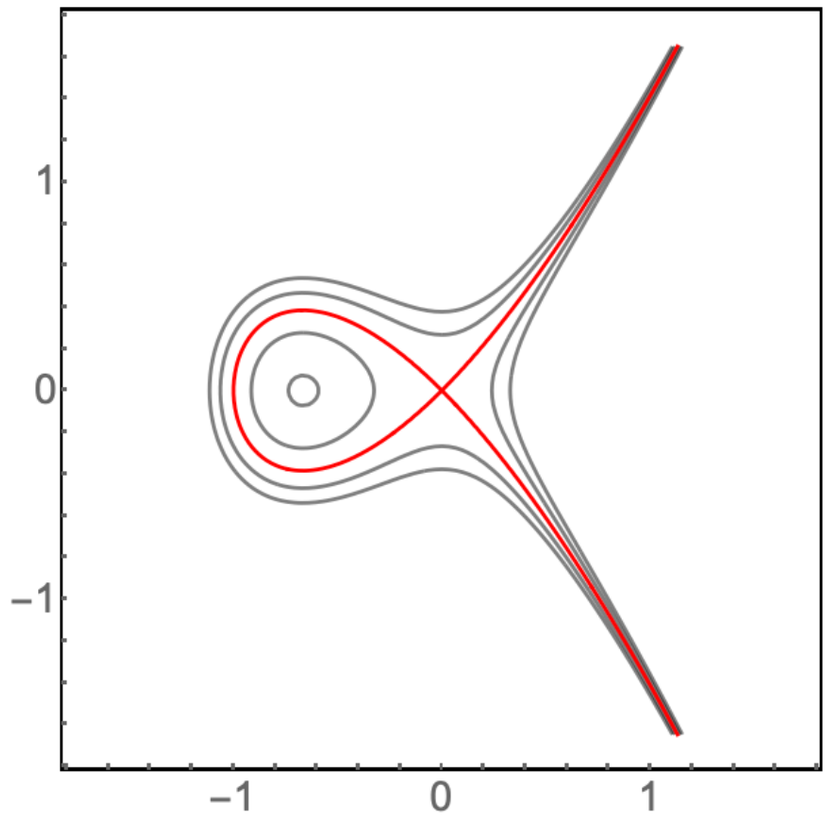


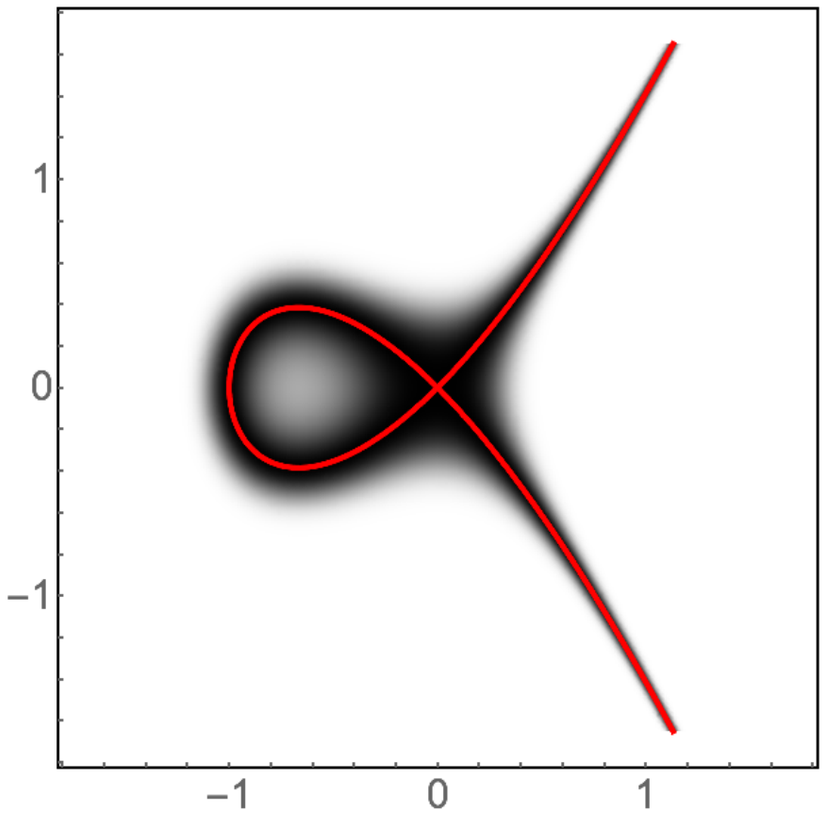
Variety normal distribution provisional

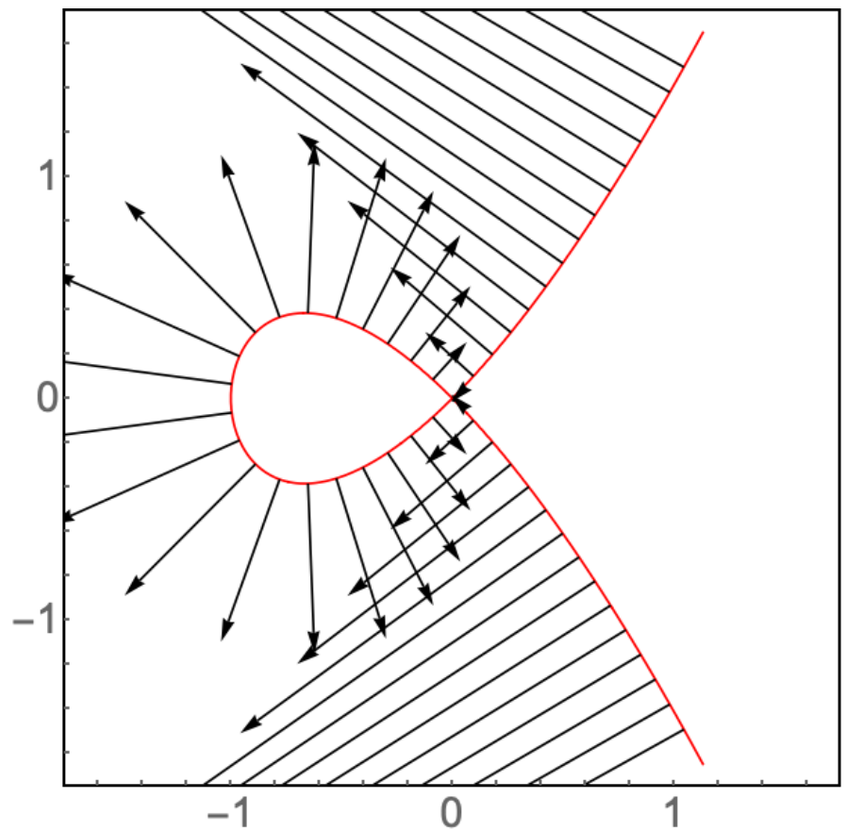
Baylor University

Evenly spaced contours







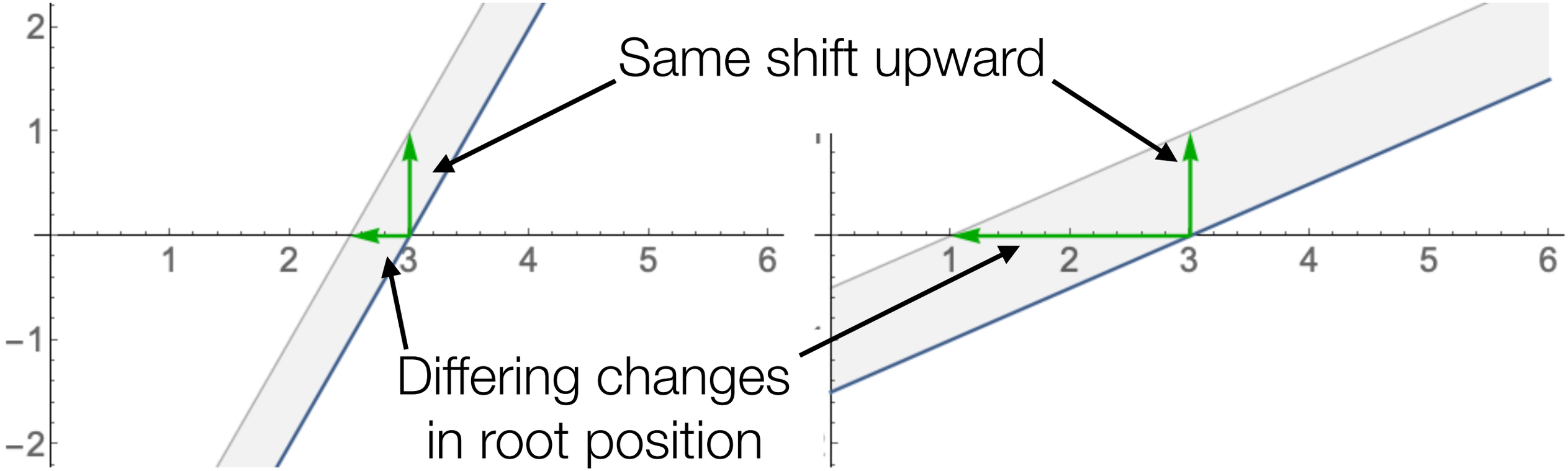


2. σ does not gauge variability globally

Probability mass does not decay evenly across variety

Example: Alpha curve, $V(y^2 - (x^3 + x^2))$

Cause: root mobility is inversely related to gradient, which varies



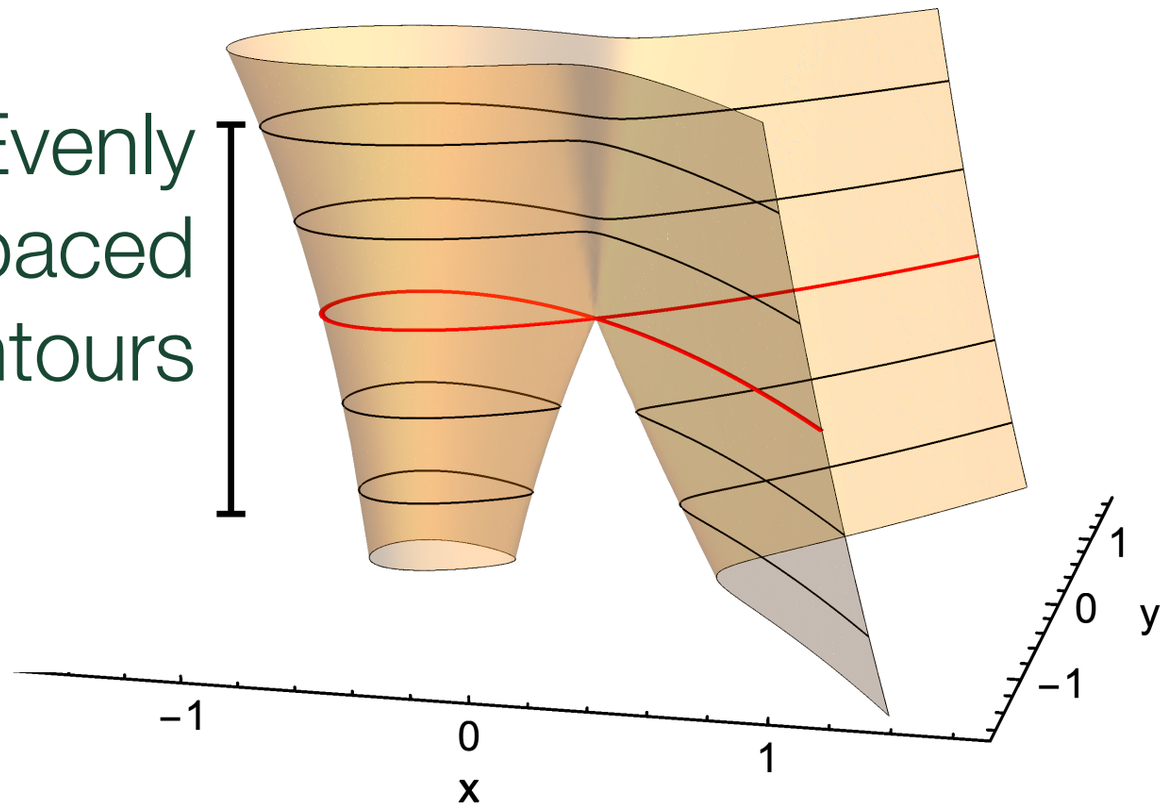
Solution: normalize g by the size of its gradient

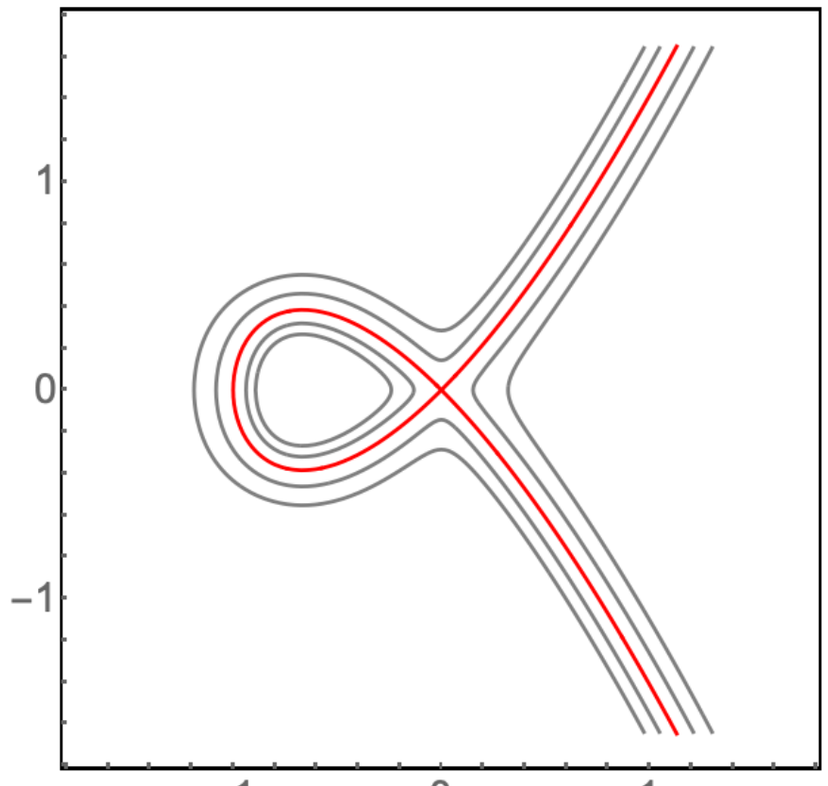
(That doesn't change the zero locus.)

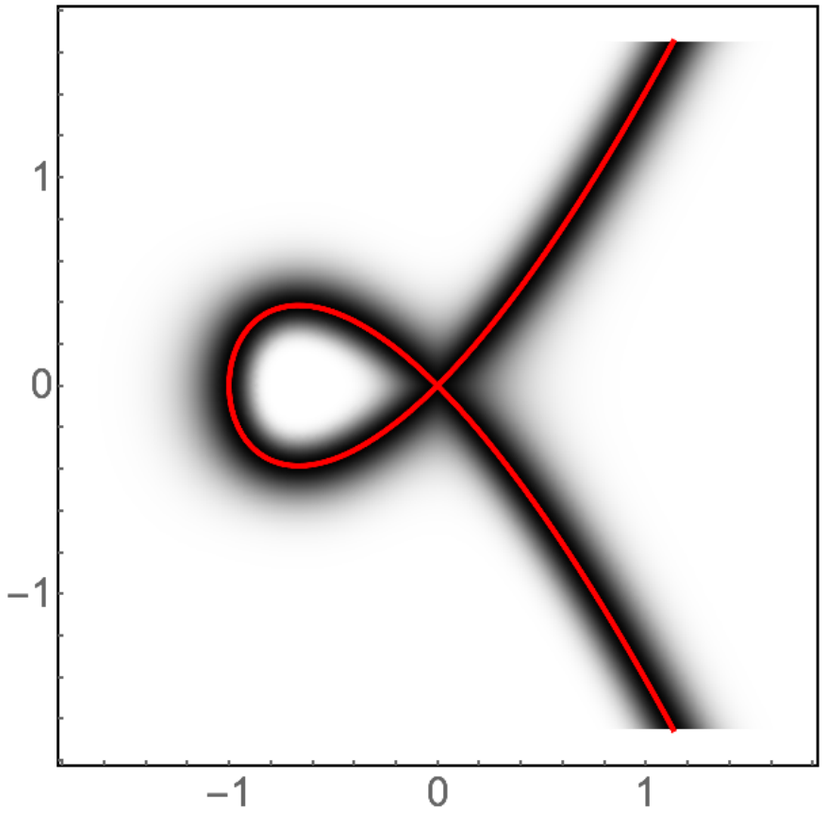
Variety normal distribution provisional

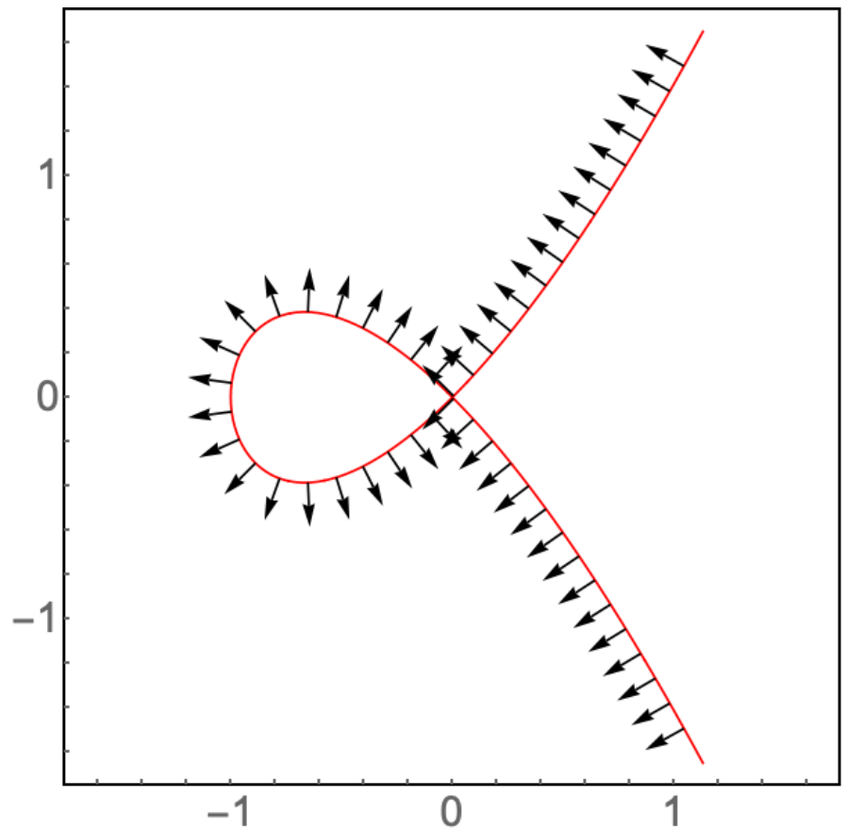
Baylor University

Evenly spaced contours





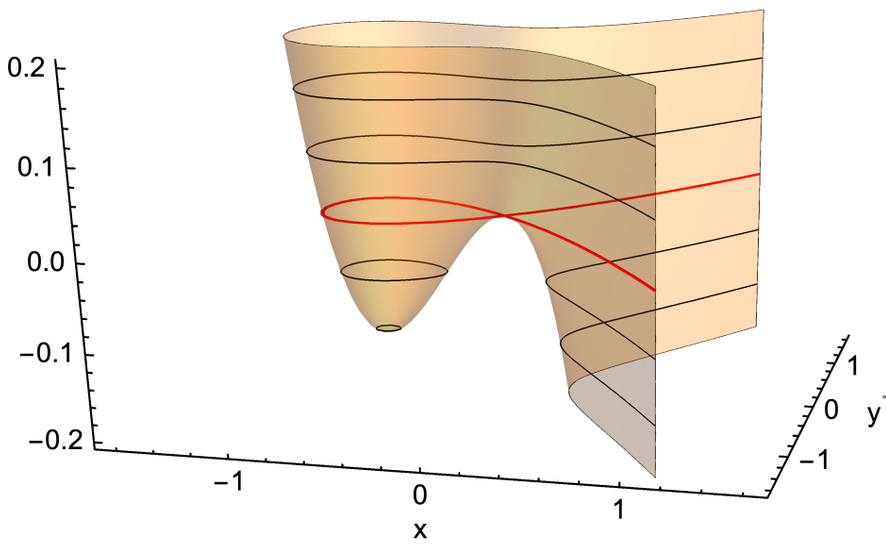


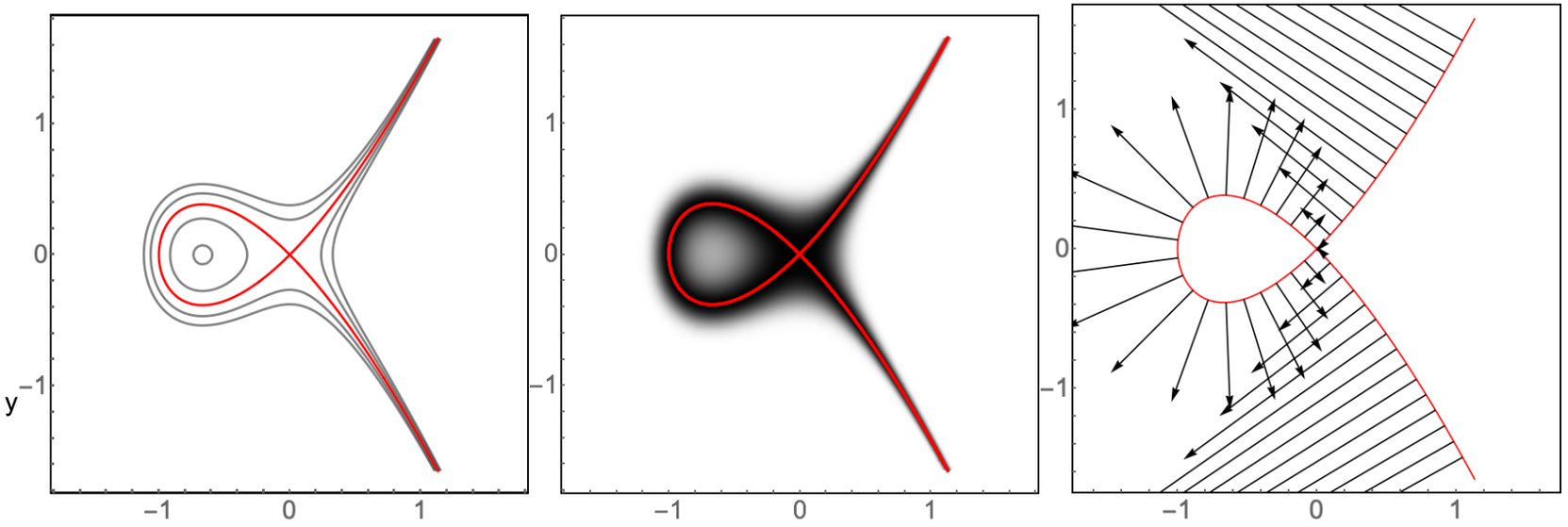


Variety normal distribution provisional

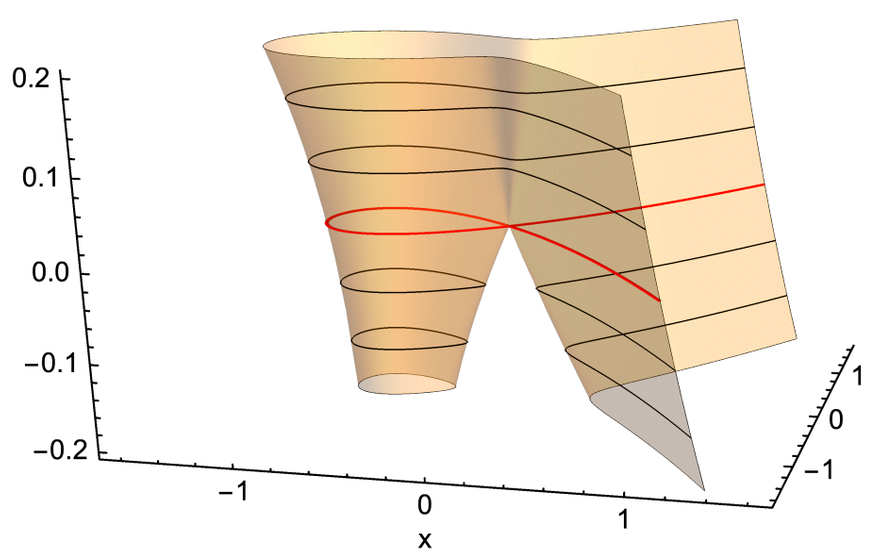
Baylor University

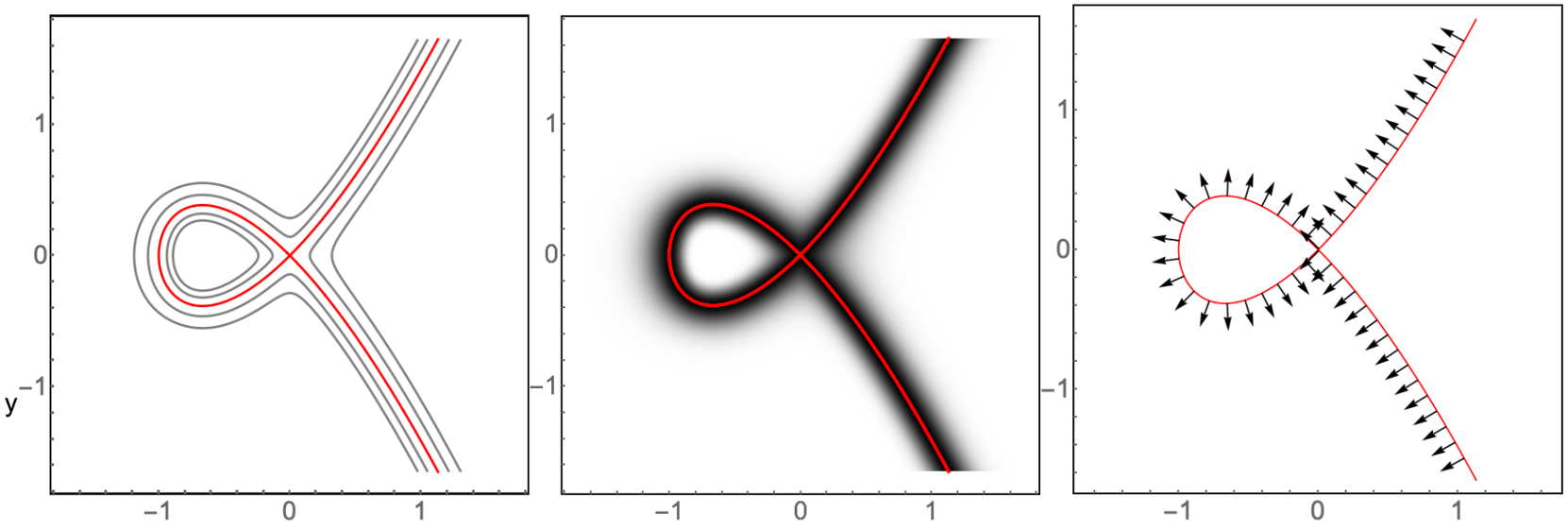
$$p(\boldsymbol{x}|g,\sigma) \propto \exp\left\{-\frac{g(\boldsymbol{x}|\boldsymbol{\beta})^2}{2\sigma^2}\right\}$$





$$p(\boldsymbol{x}|g,\sigma) \propto \exp\left\{-\frac{1}{2\sigma^2}\left(\frac{g(\boldsymbol{x}|\boldsymbol{\beta})}{||\nabla_{\boldsymbol{x}} g(\boldsymbol{x}|\boldsymbol{\beta})||_2}\right)^2\right\} = \exp\left\{-\frac{\overline{g}(\boldsymbol{x}|\boldsymbol{\beta})^2}{2\sigma^2}\right\}$$



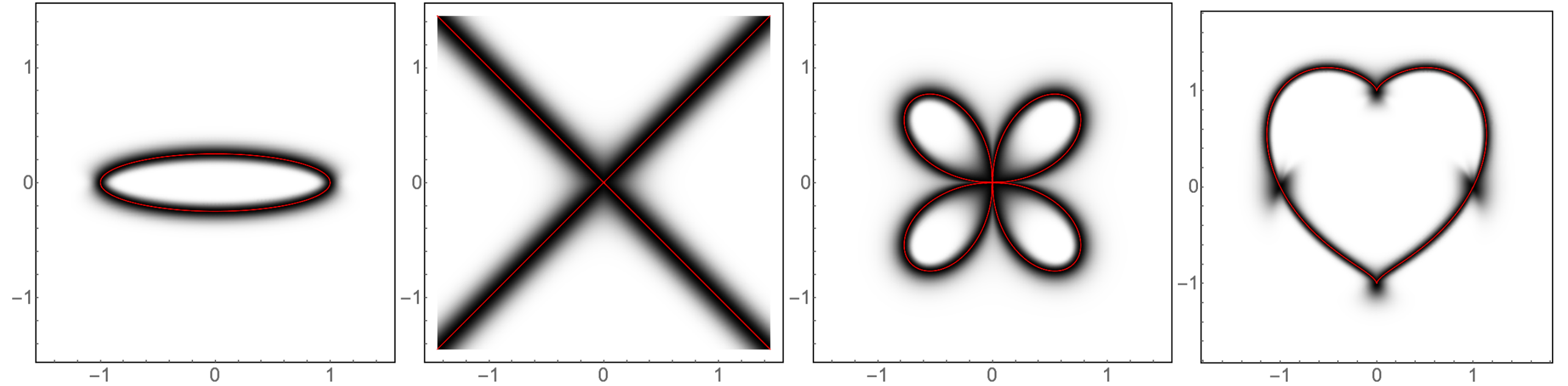


Variety normal distribution

A random vector $\boldsymbol{X}$ has the variety normal distribution if

$$p(\boldsymbol{x}|g,\sigma) \propto \exp\left\{-\frac{1}{2\sigma^2}\left(\frac{g(\boldsymbol{x}|\boldsymbol{\beta})}{||\nabla_{\boldsymbol{x}} g(\boldsymbol{x}|\boldsymbol{\beta})||_2}\right)^2\right\} = \exp\left\{-\frac{\overline{g}(\boldsymbol{x}|\boldsymbol{\beta})^2}{2\sigma^2}\right\}$$

with $g(\boldsymbol{x}|\boldsymbol{\beta}) \in \mathbb{R}[\boldsymbol{x}]$



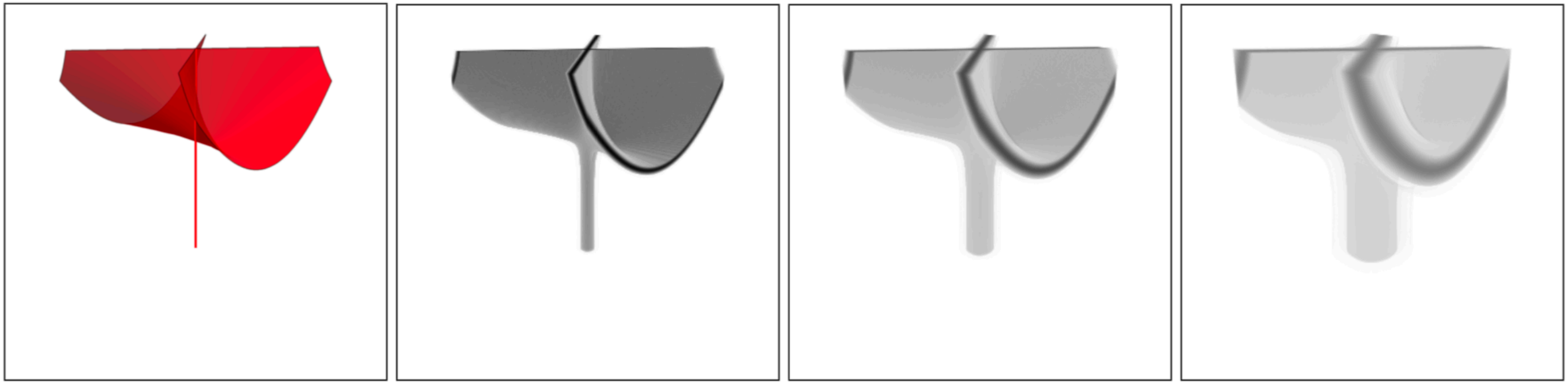
Variety normal distribution

Baylor University

A random vector $\boldsymbol{X}$ has the variety normal distribution if

$$p(\boldsymbol{x}|g,\sigma) \propto \exp\left\{-\frac{1}{2\sigma^2}\left(\frac{g(\boldsymbol{x}|\boldsymbol{\beta})}{||\nabla_{\boldsymbol{x}} g(\boldsymbol{x}|\boldsymbol{\beta})||_2}\right)^2\right\} = \exp\left\{-\frac{\overline{g}(\boldsymbol{x}|\boldsymbol{\beta})^2}{2\sigma^2}\right\}$$

with $g(\boldsymbol{x}|\boldsymbol{\beta}) \in \mathbb{R}[\boldsymbol{x}]$



Whitney umbrella $V(x^2 - y^2 z)$ for differing σ

Systems of polynomials $g_1, \ldots, g_m$ are supported by the multivariety normal distribution

The multivariate normal distribution has density

$$p(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \propto \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\}$$

The multivariety normal distribution has density

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{g}, \boldsymbol{\Sigma}) &\propto \exp\left\{-\frac{1}{2}\overline{\boldsymbol{g}}(\boldsymbol{x}|\mathbf{B})'\boldsymbol{\Sigma}^{-1}\overline{\boldsymbol{g}}(\boldsymbol{x}|\mathbf{B})\right\} \\ &= \exp\left\{-\frac{1}{2}\left(\mathbf{J}_{\boldsymbol{x}}^{+}\boldsymbol{g}(\boldsymbol{x}|\mathbf{B})\right)'\boldsymbol{\Sigma}^{-1}\left(\mathbf{J}_{\boldsymbol{x}}^{+}\boldsymbol{g}(\boldsymbol{x}|\mathbf{B})\right)\right\} \end{aligned}$$

Baylor University

Facts about the (multi)variety normal distribution

1. $m = 1$ is variety normal

2. $(g_1 = x_1 - \mu_1, \ldots, g_m = x_m - \mu_m)$ and $m = n$ is multivariate normal

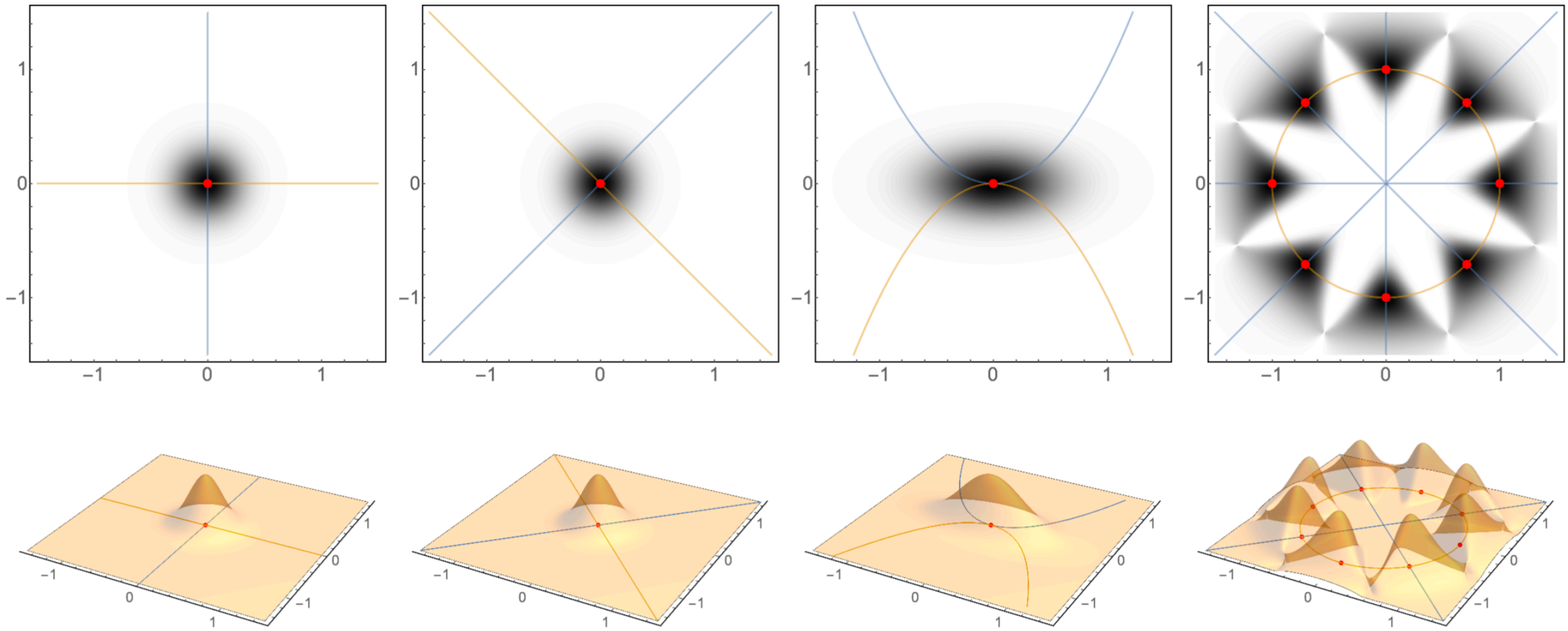
3. Σ governs "covariance", including + and – associations

I'll often refer to both as simply the "variety normal"

Multivariety normal distribution

Baylor University

$$\begin{array}{rcl} x &=& 0 \\ y &=& 0 \end{array} \qquad \begin{array}{rcl} y &=& x \\ y &=& -x \end{array} \qquad \begin{array}{rcl} y &=& x^2 \\ y &=& -x^2 \end{array} \qquad \begin{array}{rcl} 9xy^3 &=& x^3 y \\ x^2 + (3y)^2 &=& 1 \end{array}$$



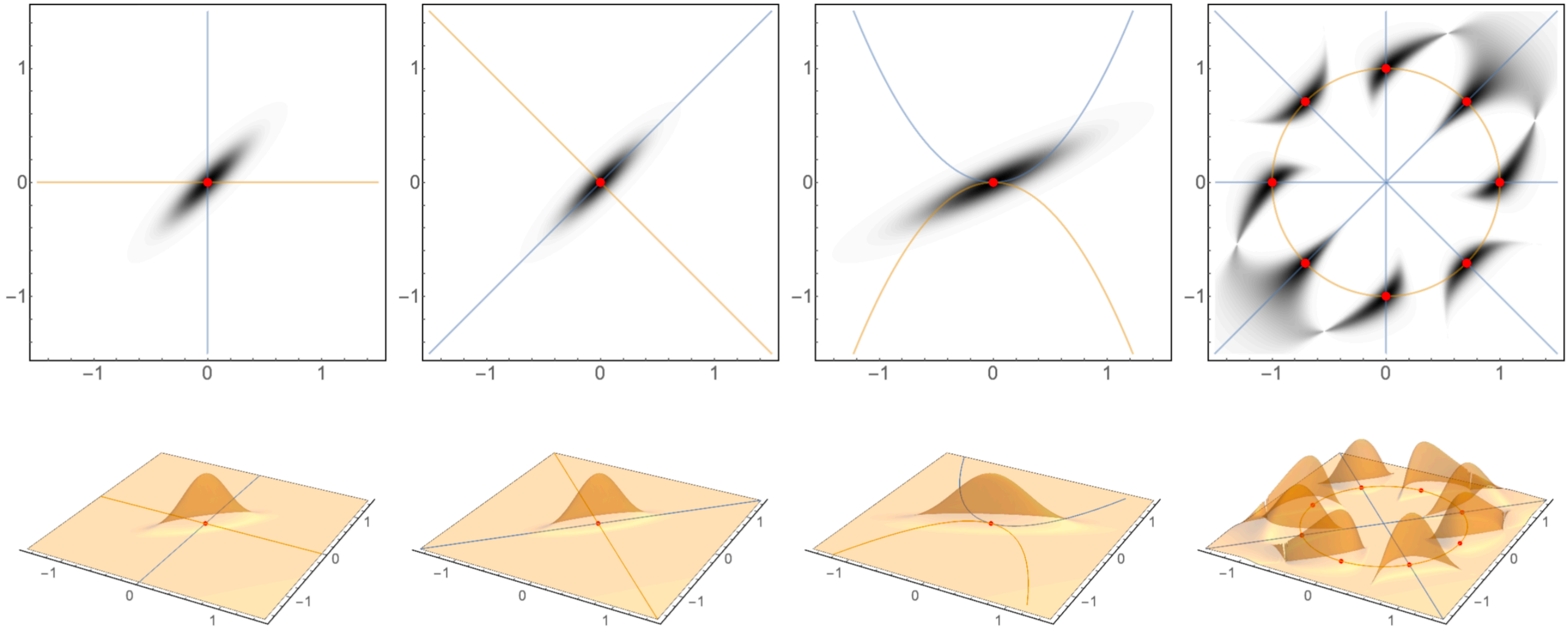
$$\rho = 0$$

$$m = n$$

Multivariety normal distribution

Baylor University

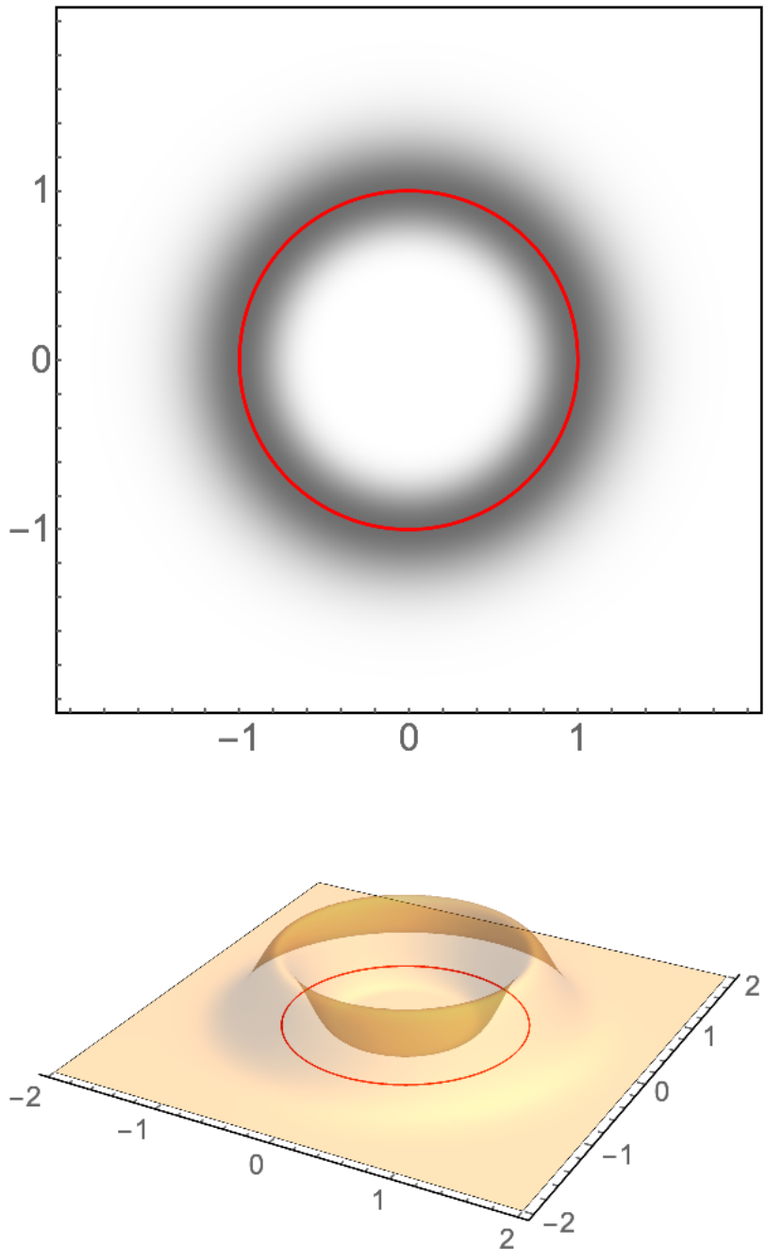
$$\begin{array}{rcl} x & = & 0 \\ y & = & 0 \end{array} \qquad \begin{array}{rcl} y & = & x \\ y & = & -x \end{array} \qquad \begin{array}{rcl} y & = & x^2 \\ y & = & -x^2 \end{array} \qquad \begin{array}{rcl} 9xy^3 & = & x^3y \\ x^2+(3y)^2 & = & 1 \end{array}$$



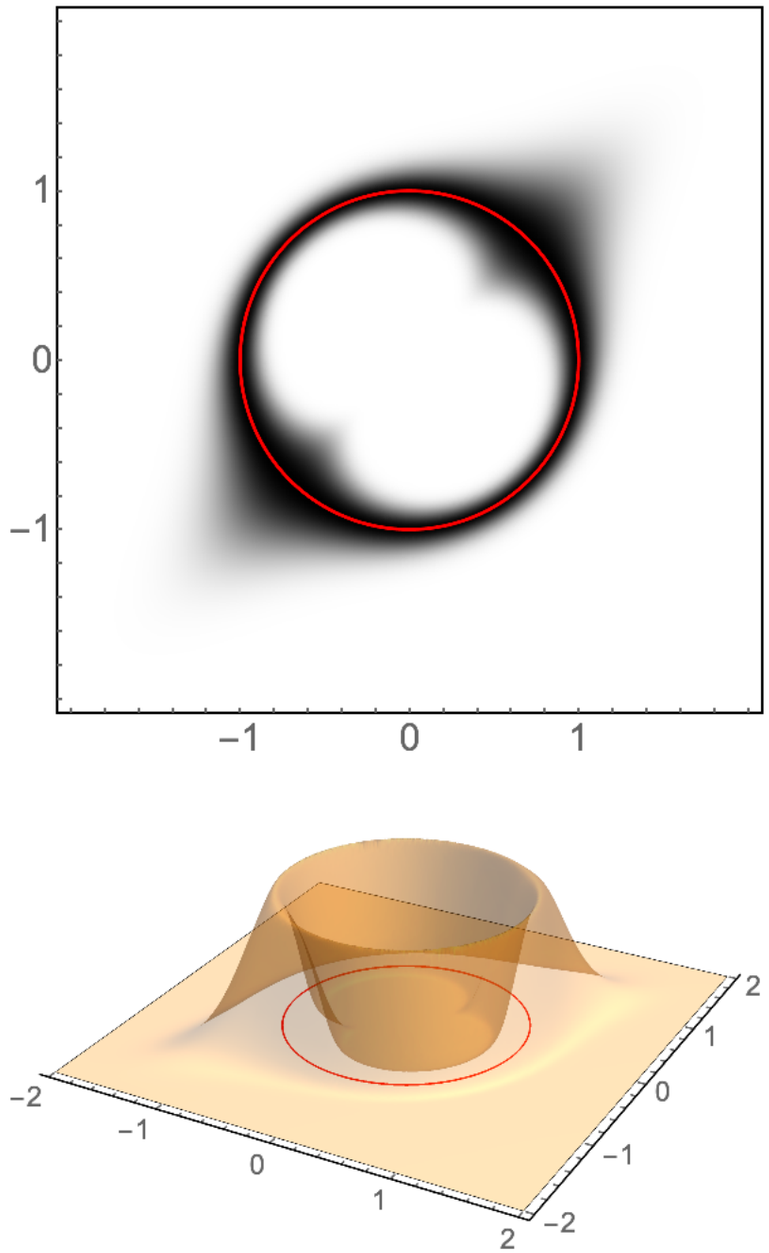
$$\rho = .9$$

$$m = n$$

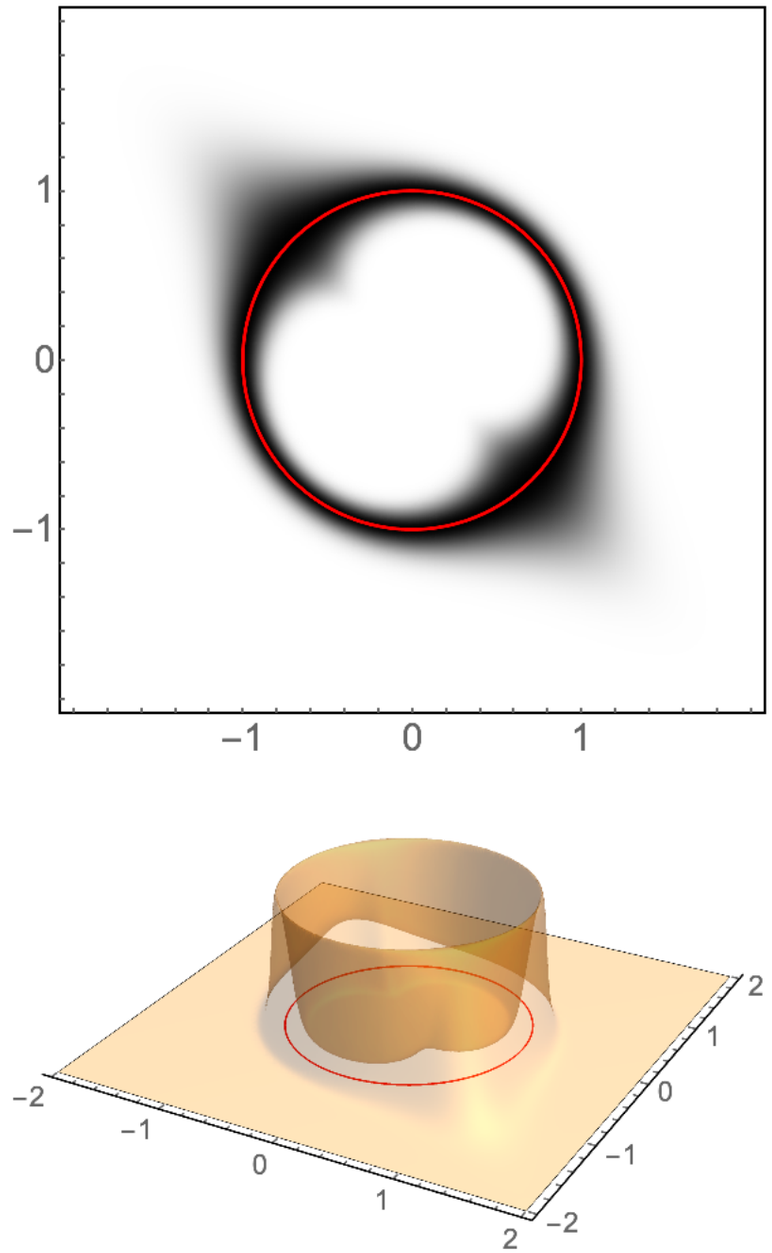
Multivariety normal distribution



$\rho = 0$



$\rho = .9$



$\rho = -.9$

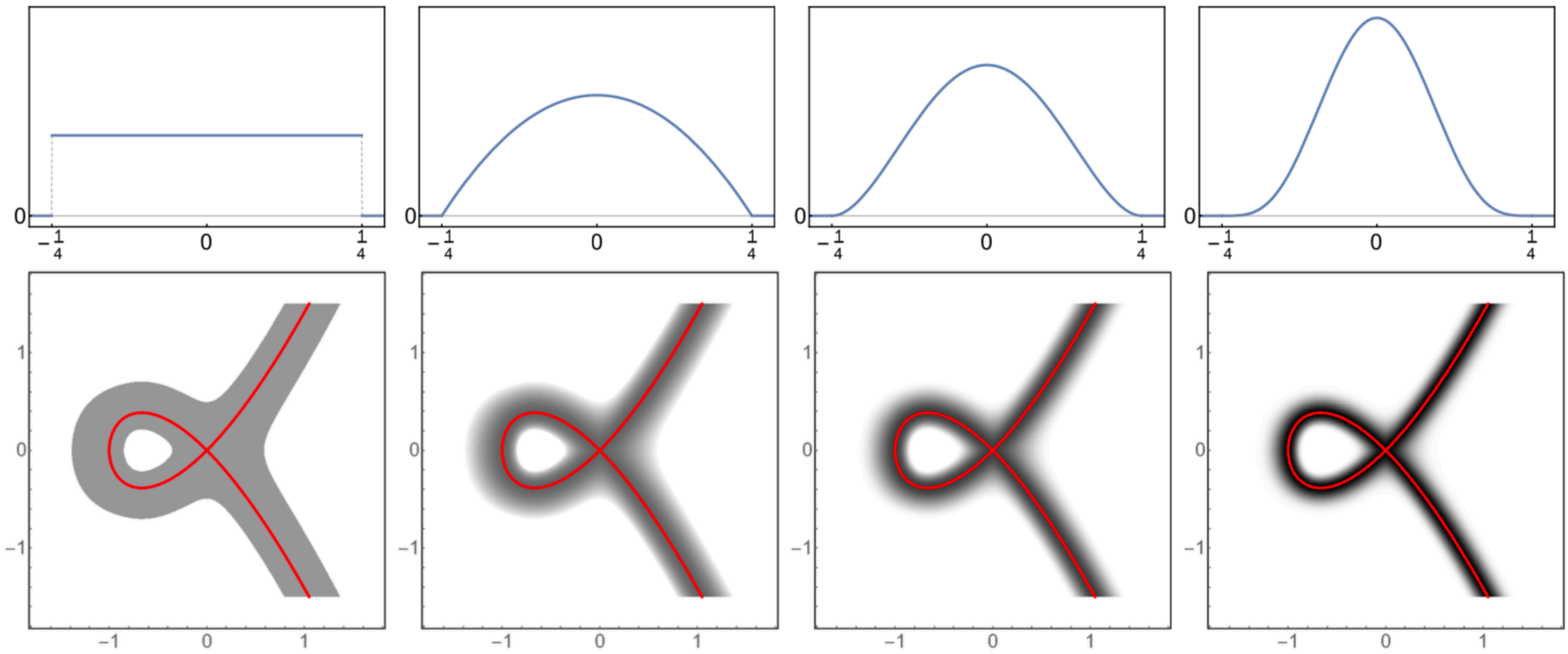
$m < n$

Variety induced distributions

Baylor University

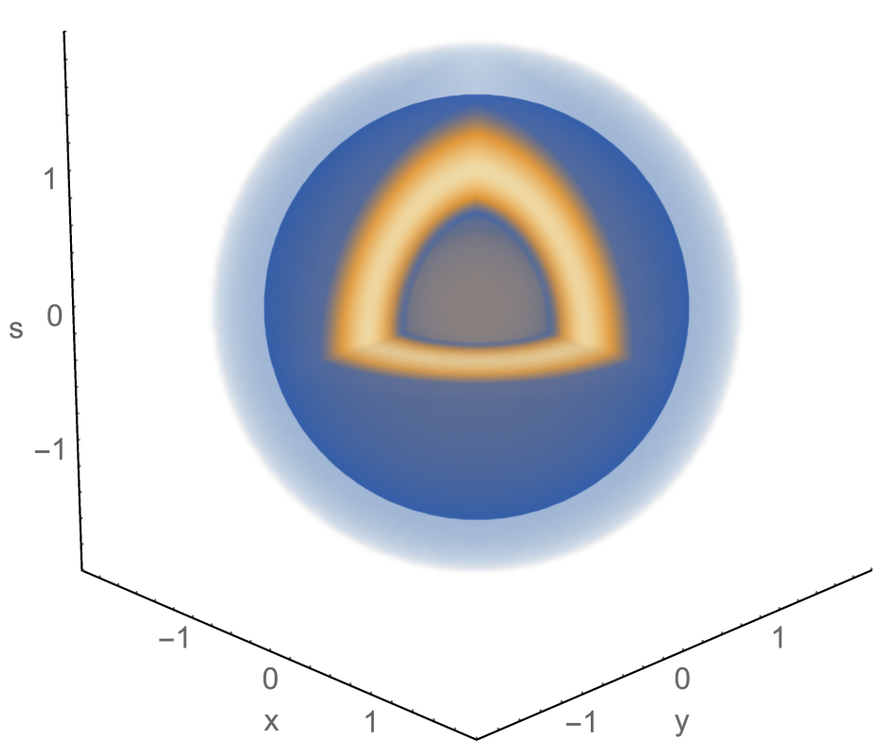
The kernel of any PDF can be used to induce variety distributions via location-scale transformations

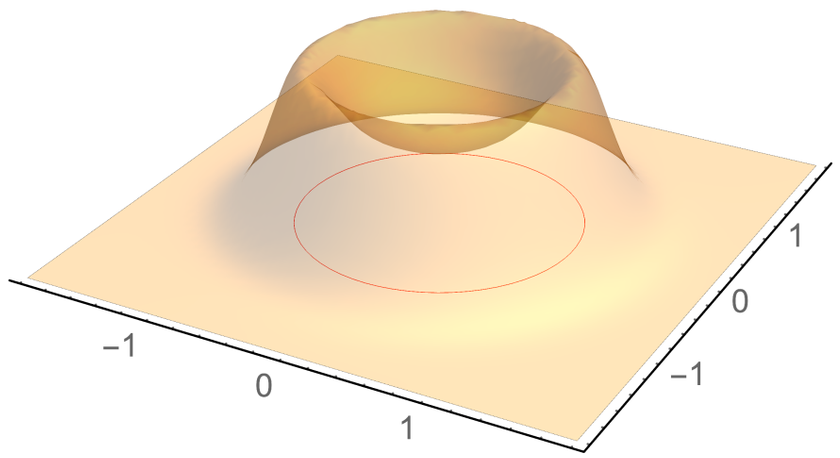
Example. Beta distributions scaled and shifted by 1/2

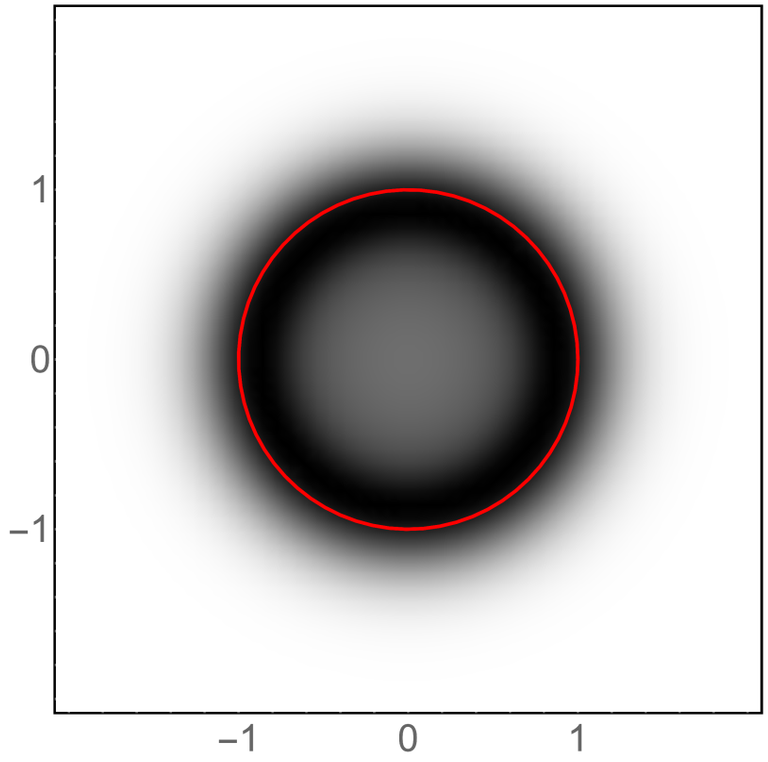


Distributions about semi-algebraic regions can be created through slack variables and marginalization

$$x^2 + y^2 \leq 1 \iff 0 \leq 1 - (x^2 + y^2)$$
$$\iff 0 = 1 - (x^2 + y^2) - s^2$$







Sampling and implementation

Markov chain Monte Carlo

Baylor University

Markov chain Monte Carlo (MCMC) is a class of algorithms for sampling probability distributions

- Stationary distribution is the target distribution
- Target distribution does not need to be normalized
- Foundational in Bayesian statistics ⇒ good software (BUGS, Stan)

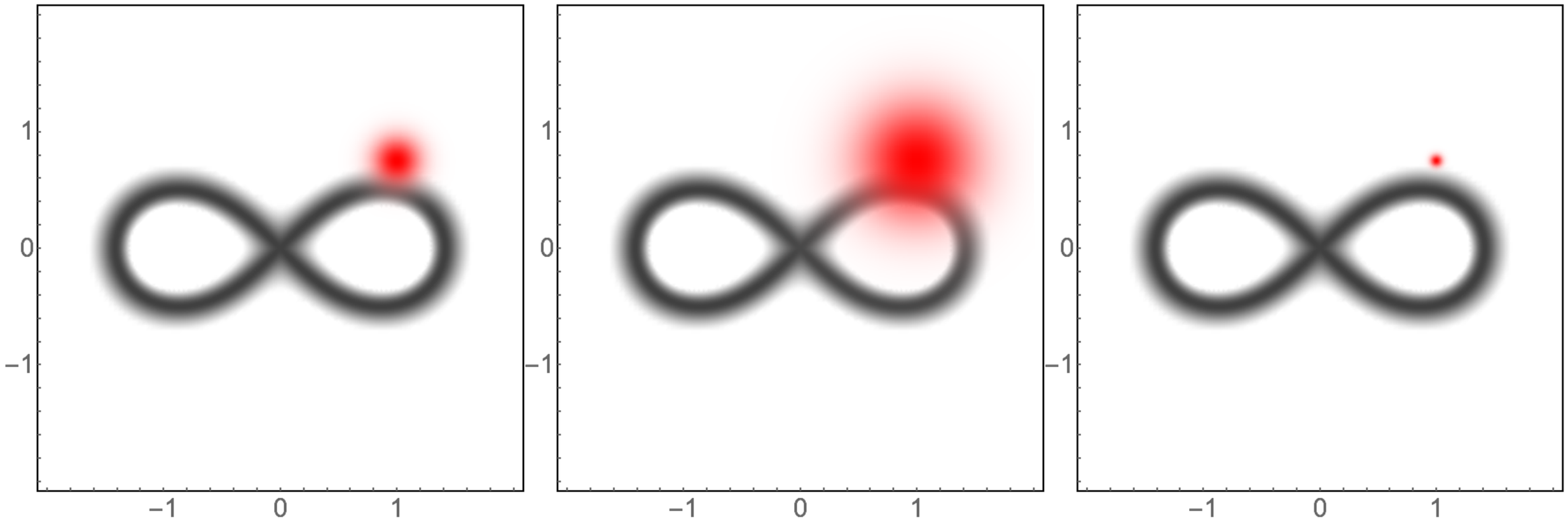
Iterate two basic steps (MCMC used here)

1. Generate an observation that might come from target (proposal)
2. Accept/reject probabilistically according to Metropolis-Hastings

Best case: Starting anywhere, chain converges to draws from target distribution

Random walk Metropolis

From current location, propose multivariate normal step



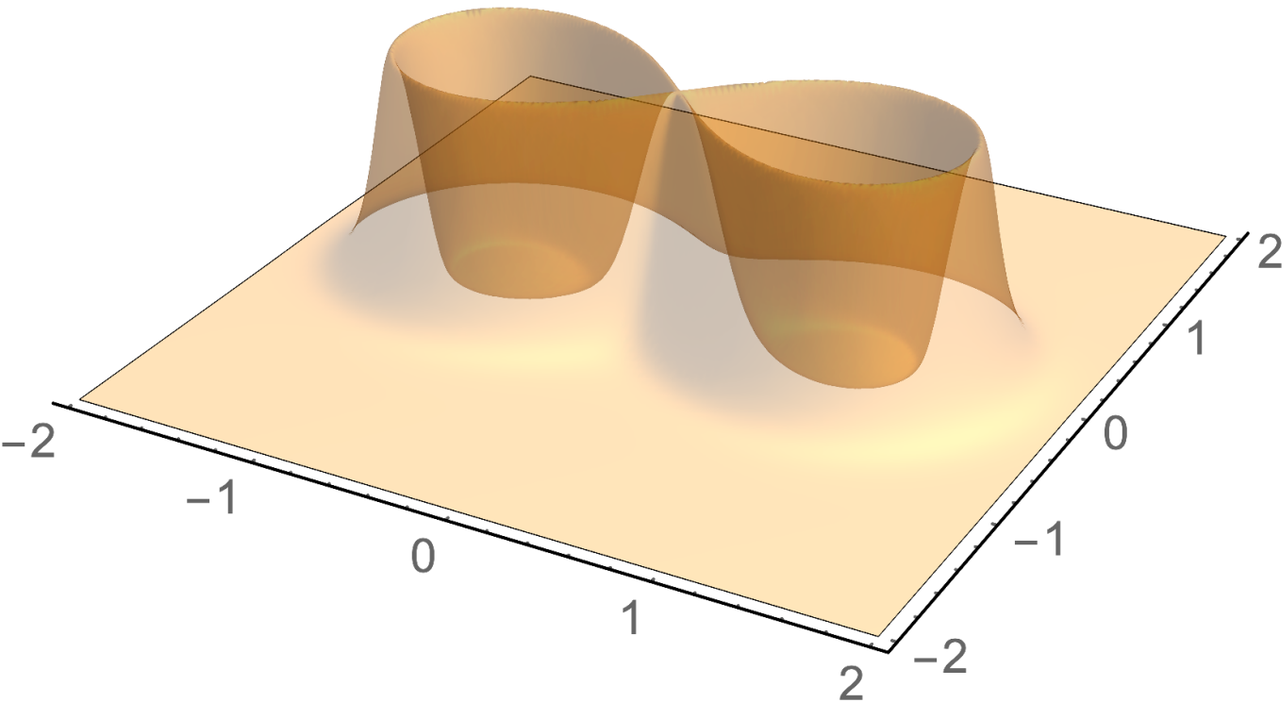
If variability is too large, unacceptably low acceptance rate

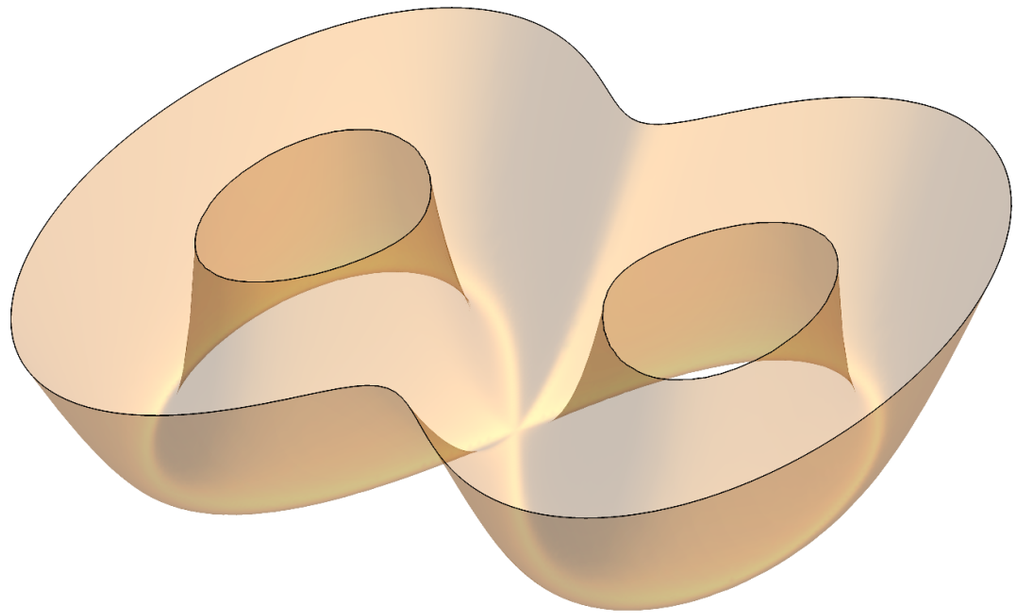
If variability is too small, unacceptably slow exploration

Both problems get worse in high dimensions

From current, propose step from physics simulation

Marble rolling on $(\bar{g}^2/\sigma^2)$'s surface, frictionless, given initial flick



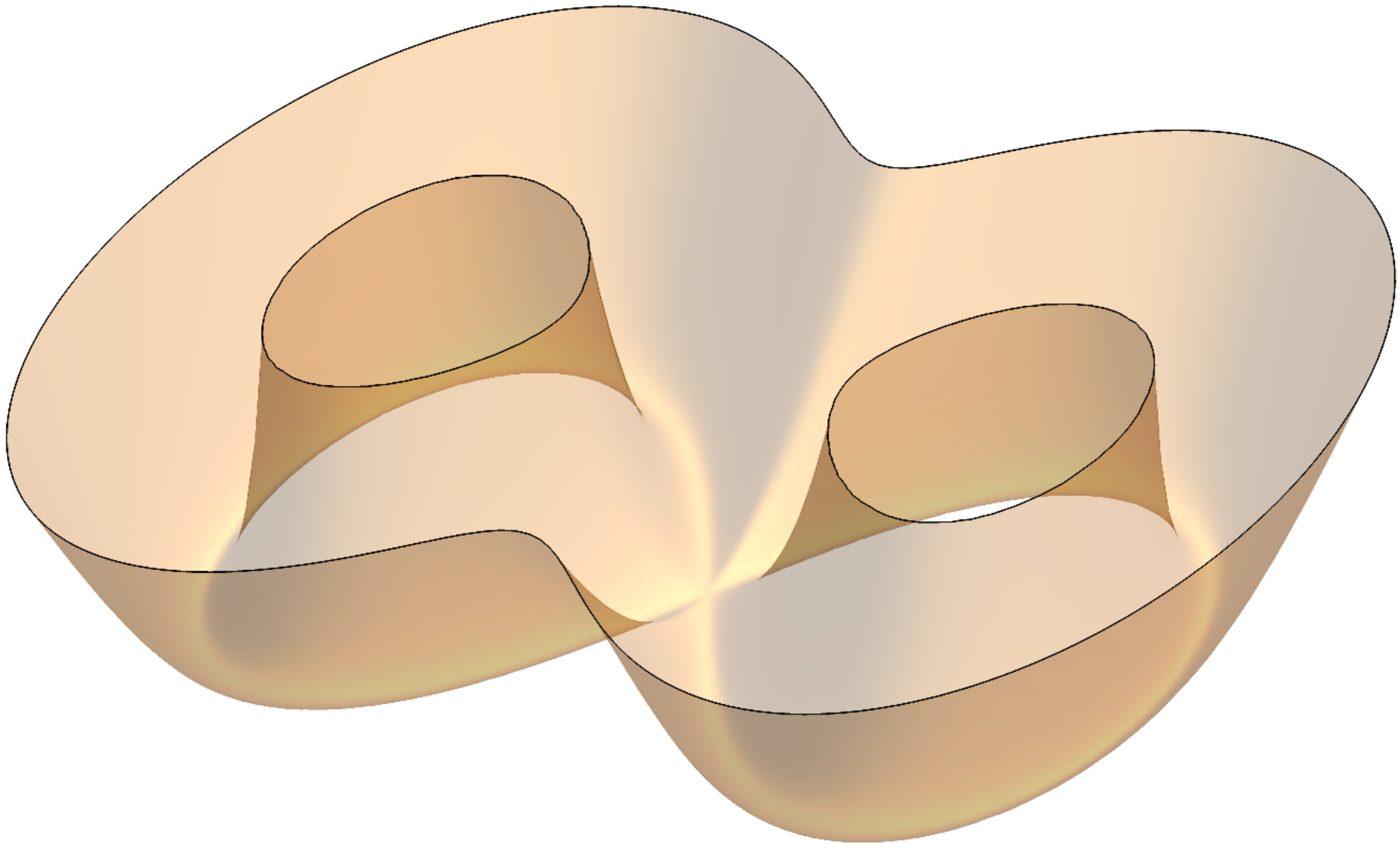


$$p(\boldsymbol{x}|\overline{\boldsymbol{g}}, \boldsymbol{\Sigma}) \qquad -\log p(\boldsymbol{x}|\overline{\boldsymbol{g}}, \boldsymbol{\Sigma}) = \overline{\boldsymbol{g}}'\boldsymbol{\Sigma}^{-1}\overline{\boldsymbol{g}}$$

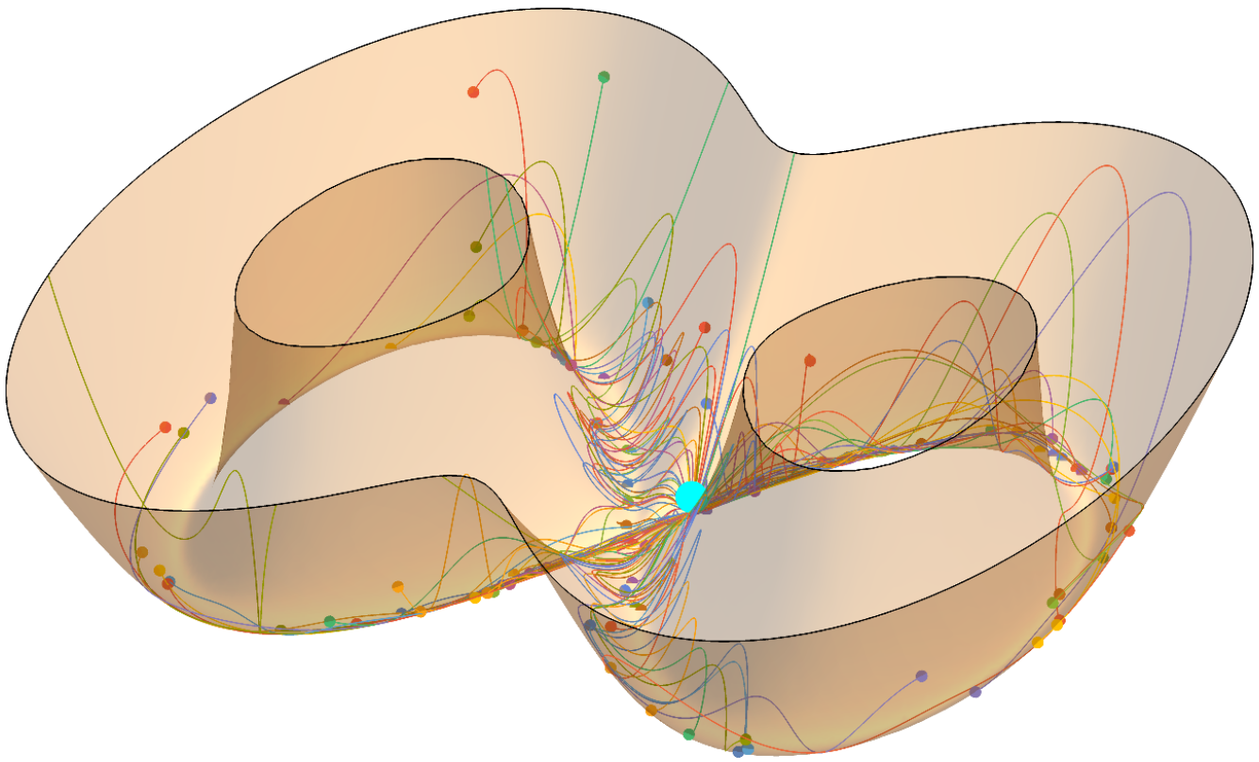
Impart random momentum, track position numerically, stop

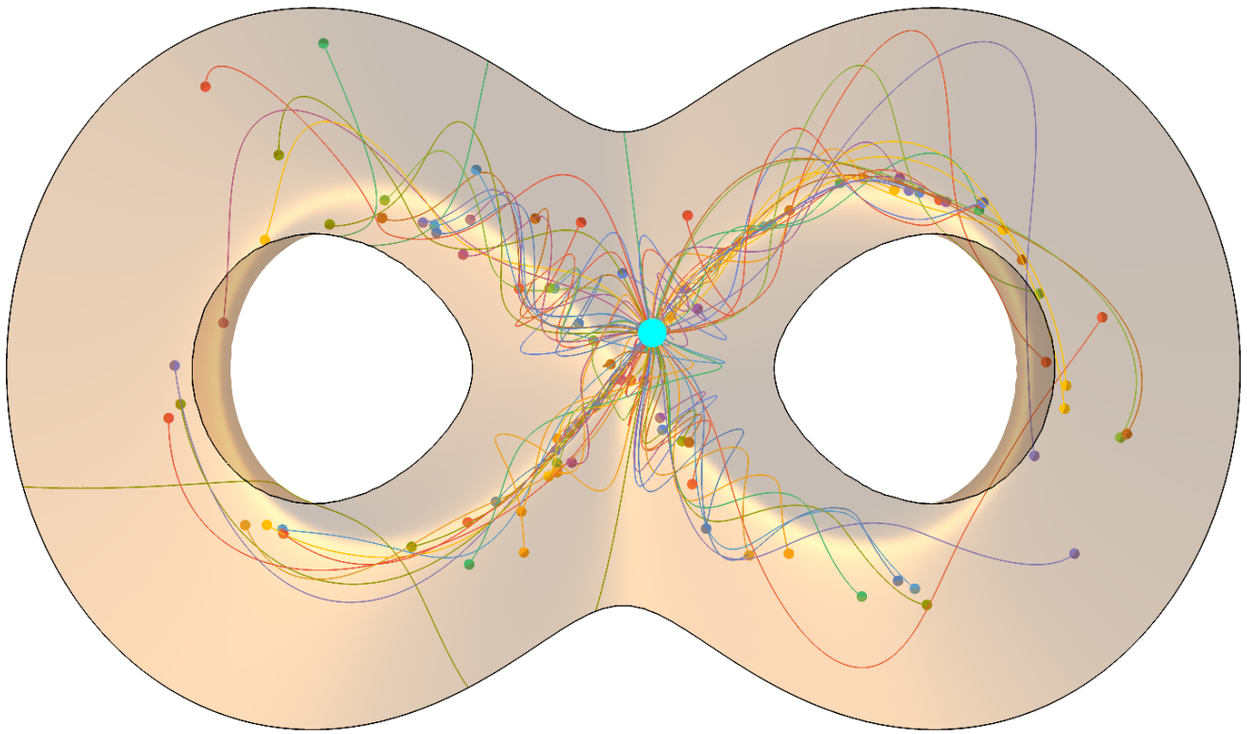
Hamiltonian Monte Carlo (HMC)

Baylor University

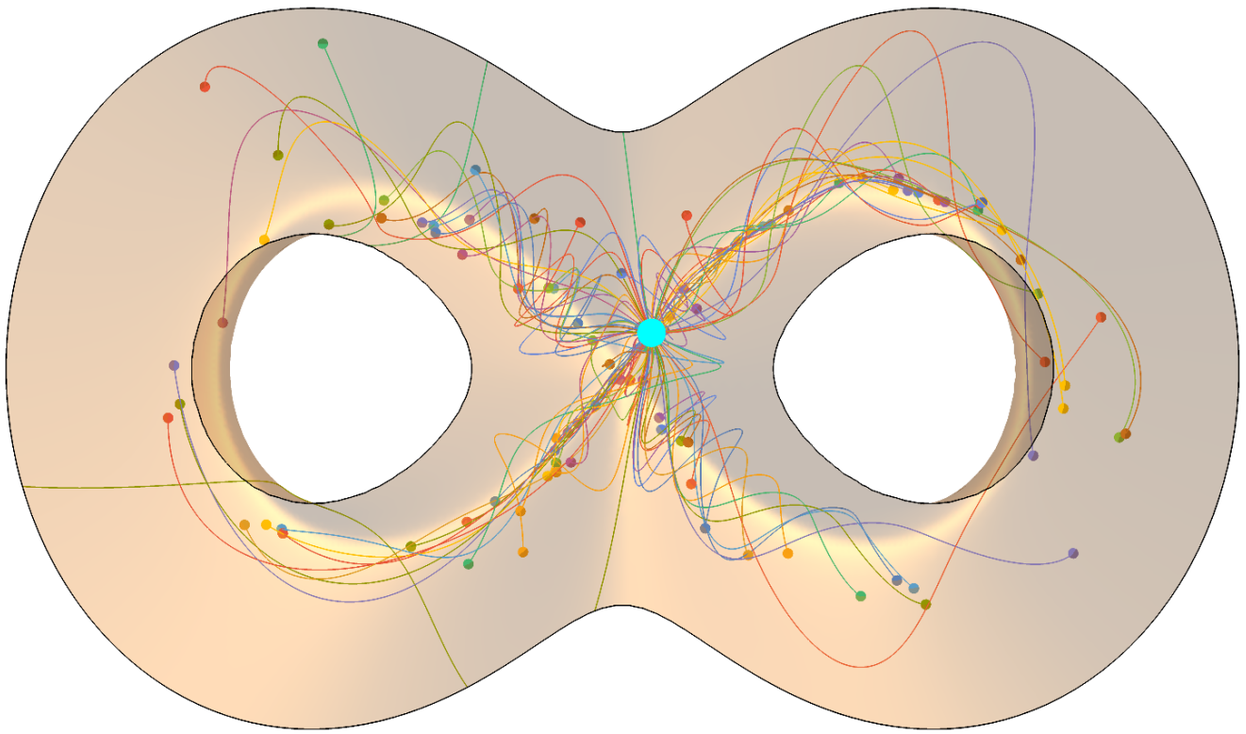


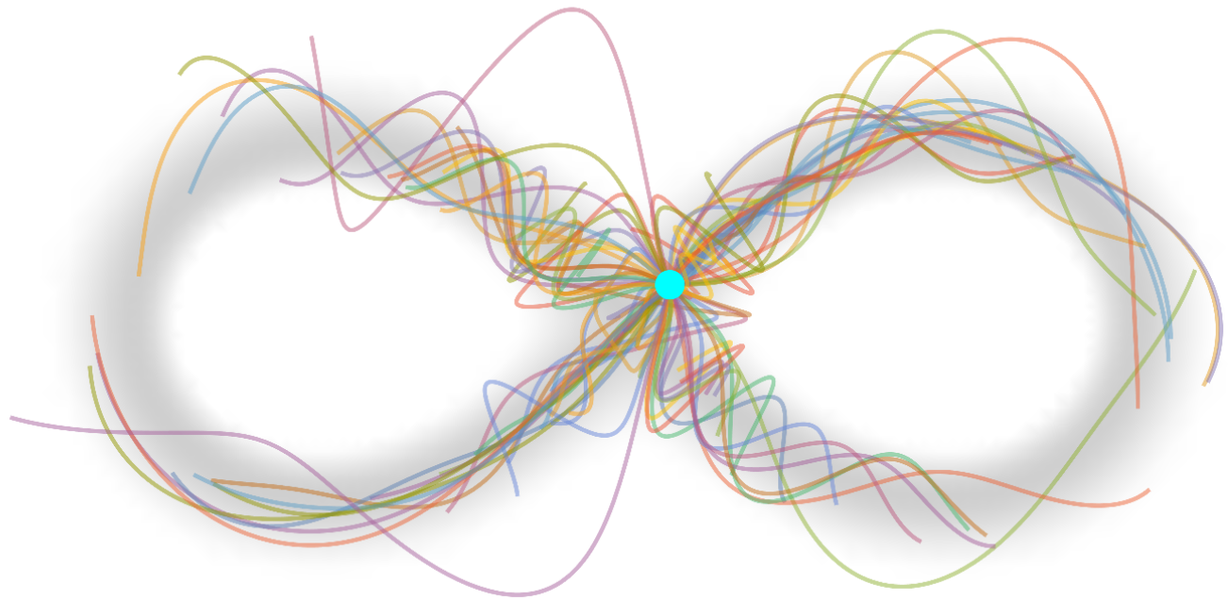
100 proposed transitions from a single state





100 proposed transitions from a single state

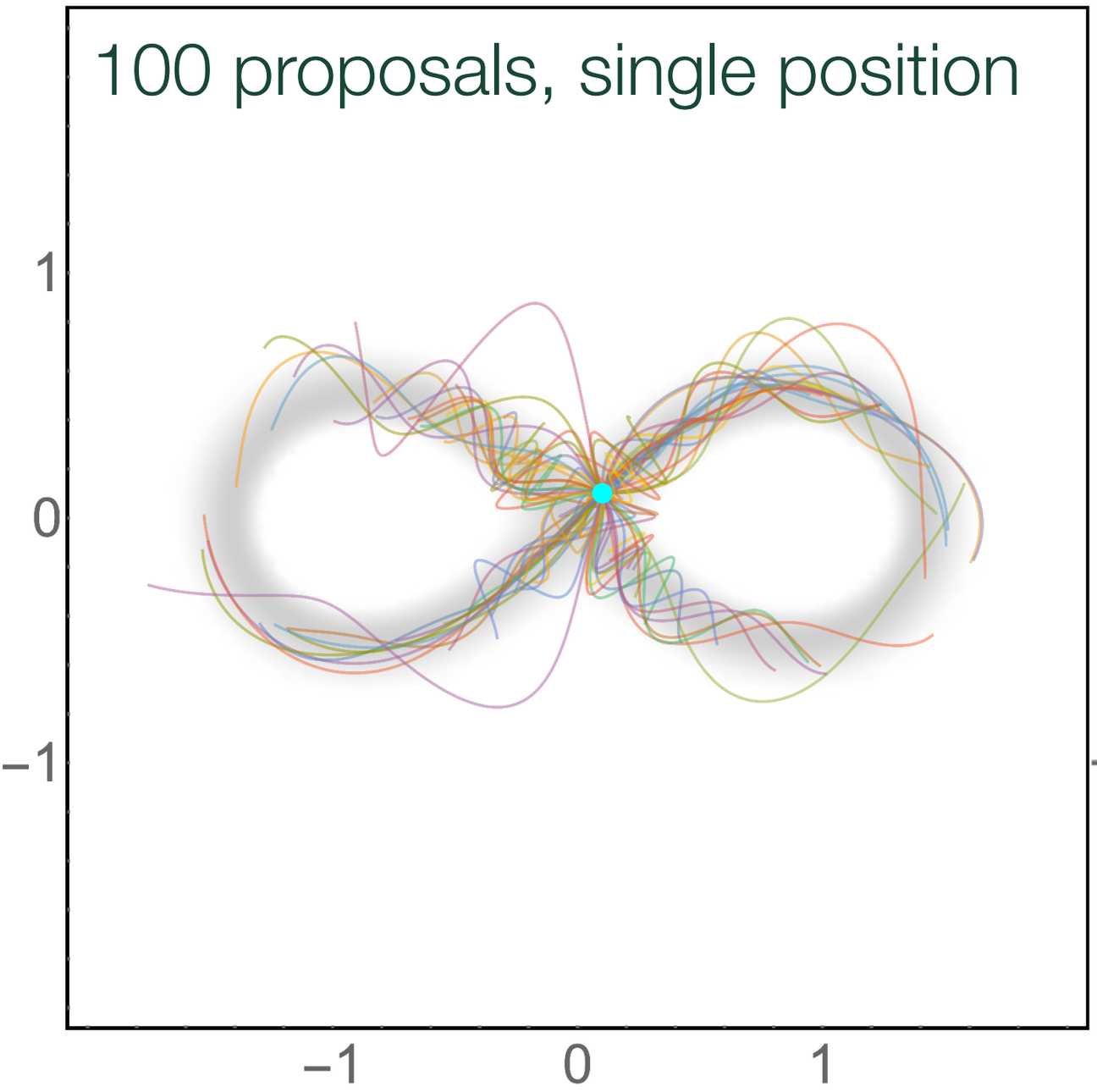


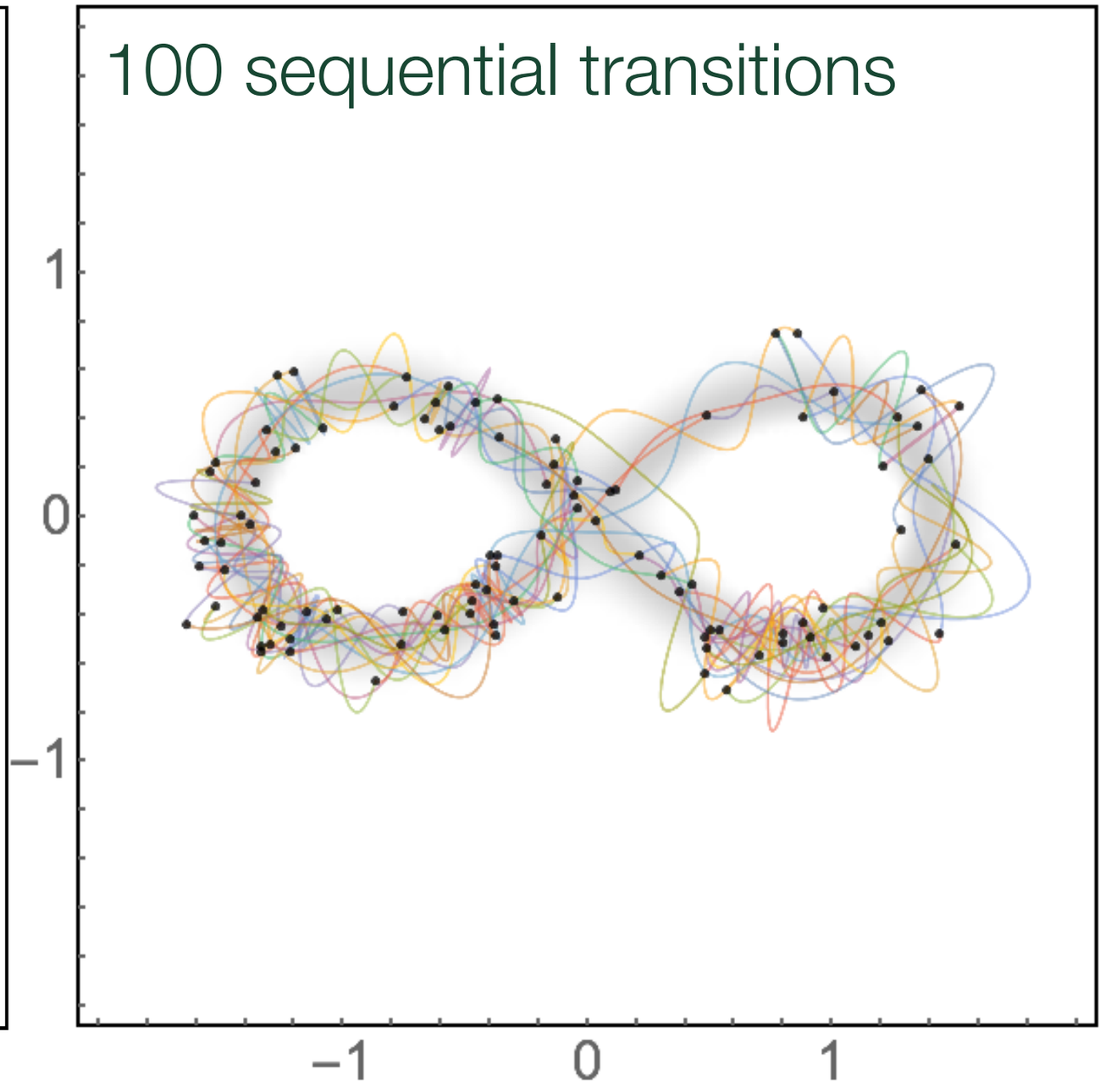


(Path colors inconsistent)

Hamiltonian Monte Carlo (HMC)

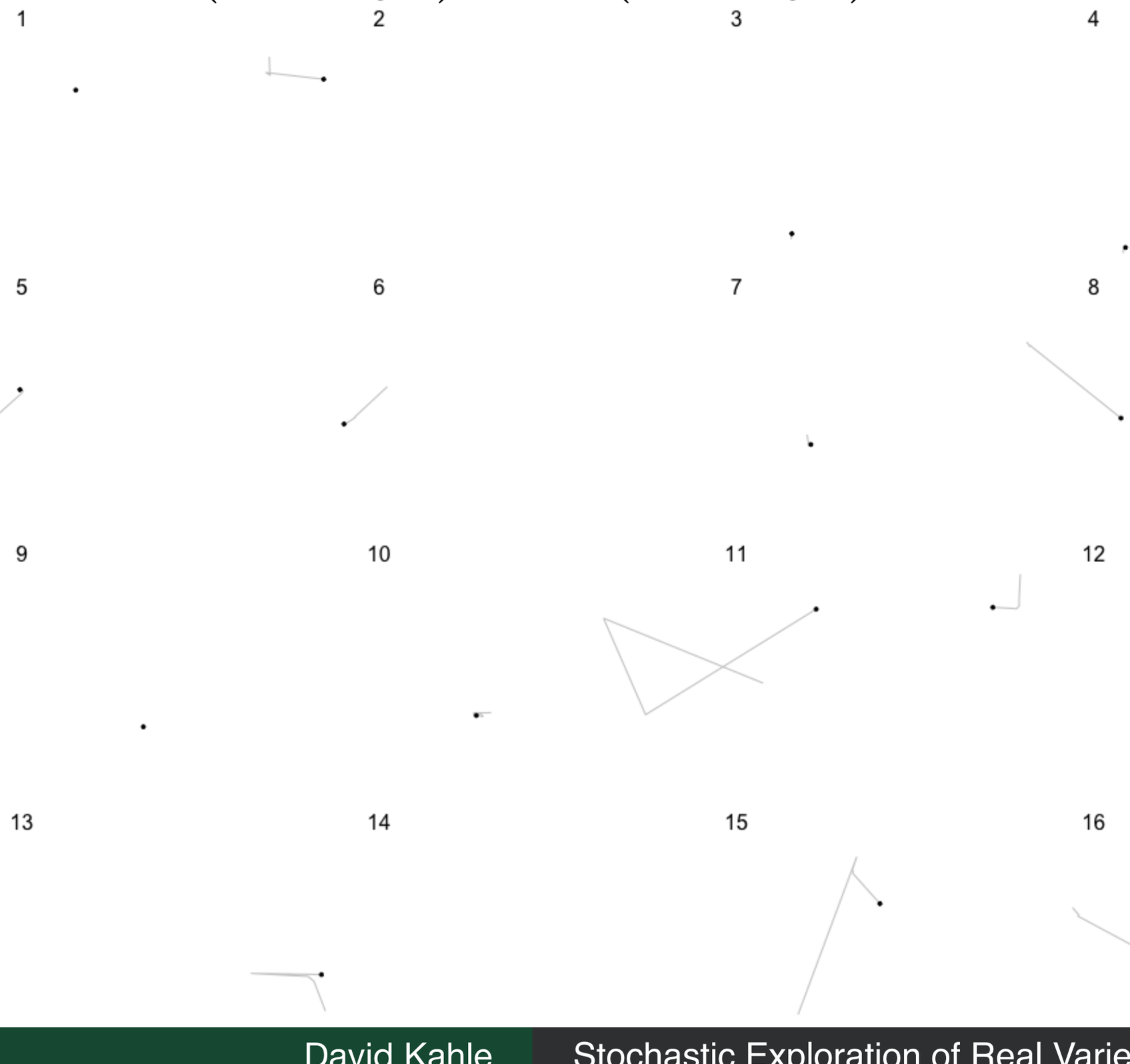
Baylor University





Hamiltonian Monte Carlo (HMC)

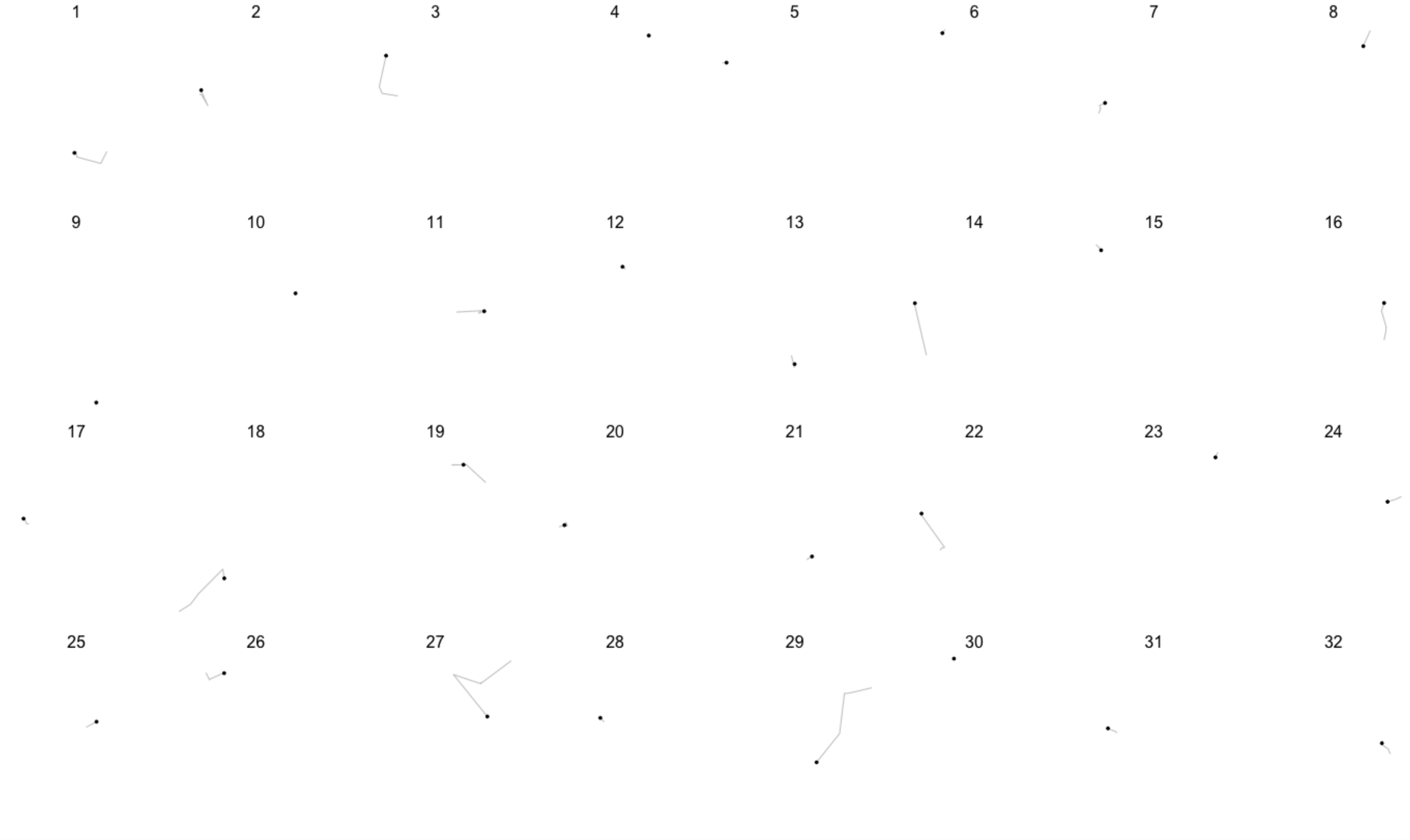
$$(x^2 + y^2)^2 - 2(x^2 - y^2)$$



Hamiltonian Monte Carlo (HMC)

Baylor University

$$(x^2 + y^2 - 1)^3 - x^2 y^3$$



HMC is implemented in Stan, a probabilistic programming language and Bayesian engine

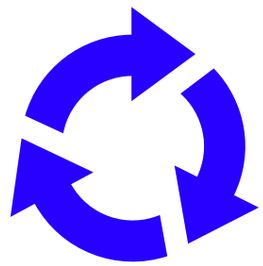
Stan specification

```
data {
  real y_obs;
  real<lower=0> si;
}

parameters {
  real x;
  real y;
}

transformed parameters {
  real g = (x^2 + (4*y)^2 - 1);
  real ndg = sqrt((2*x)^2 + (2*(4*y)*4)^2);
  real gbar = g / ndg;
}

model {
  y_obs ~ normal(gbar, si);
}
```

C++

```
// Code generated by Stan version 2.17.0

#include <stan/model/model_header.hpp>

namespace model867873611022_stan_code_namespace {

using std::istream;
using std::string;
using std::stringstream;
using std::vector;
using stan::io::dump;
using stan::math::lgamma;
using stan::model::prob_grad;
using namespace stan::math;

typedef Eigen::Matrix<double,Eigen::Dynamic,1> vec
typedef Eigen::Matrix<double,1,Eigen::Dynamic> row
typedef Eigen::Matrix<double,Eigen::Dynamic,Eigen:
```

Sample



Interfaces : R, Julia, Python, CLI, …

Many chains can be run in parallel

MVN as a posterior

The MVN distribution can be represented as the posterior distribution of a non-identifiable model

Bayes' theorem is

$$\underbrace{p(\boldsymbol{\theta}|\boldsymbol{y})}_{\text{Posterior}} = \frac{p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}} \propto \underbrace{p(\boldsymbol{y}|\boldsymbol{\theta})}_{\text{Likelihood}}\underbrace{p(\boldsymbol{\theta})}_{\text{Prior}}$$

Likelihood: Data, Parameter

Prior: Parameter

MVN is the posterior of the model $\boldsymbol{Y} \sim \mathcal{N}_m(\overline{\boldsymbol{g}}, \boldsymbol{\Sigma})$ with an improper flat prior on $\boldsymbol{x}$ and $\boldsymbol{y} = \boldsymbol{0}$ is observed

Roles of data and parameter swap

Bayes – Greek varies, Latin fixed/known

Here – Greek fixed/known, Latin varies

MVN as a posterior

Baylor University

Bayes' theorem is

$$\underbrace{p(\boldsymbol{\theta}|\boldsymbol{y})}_{\text{Posterior}} = \frac{p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}} \propto \underbrace{p(\boldsymbol{y}|\boldsymbol{\theta})}_{\text{Likelihood}}\underbrace{p(\boldsymbol{\theta})}_{\text{Prior}}$$

Data Parameter

$$p(\boldsymbol{x}|\boldsymbol{g}, \boldsymbol{\Sigma}) \propto \underbrace{\exp\left\{-\frac{1}{2}(\mathbf{0} - \overline{\boldsymbol{g}}(\boldsymbol{x}|\boldsymbol{\beta}))'\boldsymbol{\Sigma}^{-1}(\mathbf{0} - \overline{\boldsymbol{g}}(\boldsymbol{x}|\boldsymbol{\beta}))\right\}}_{\text{Likelihood}} \underbrace{\times 1}_{\text{Prior}}$$

Data Given Parameter

Baylor University

HMC is implemented in Stan, a probabilistic programming language and Bayesian engine

Stan specification

```
data {
  real y_obs;
  real<lower=0> si;
}

parameters {
  real x;
  real y;
}

transformed parameters {
  real g = (x^2 + (4*y)^2 - 1);
  real ndg = sqrt((2*x)^2 + (2*(4*y)*4)^2);
  real gbar = g / ndg;
}

model {
  y_obs ~ normal(gbar, si);
}
```

C++

```
// Code generated by Stan version 2.17.0

#include <stan/model/model_header.hpp>

namespace model867873611022_stan_code_namespace {

using std::istream;
using std::string;
using std::stringstream;
using std::vector;
using stan::io::dump;
using stan::math::lgamma;
using stan::model::prob_grad;
using namespace stan::math;

typedef Eigen::Matrix<double,Eigen::Dynamic,1> vec
typedef Eigen::Matrix<double,1,Eigen::Dynamic> row
typedef Eigen::Matrix<double,Eigen::Dynamic,Eigen:
```

Sample

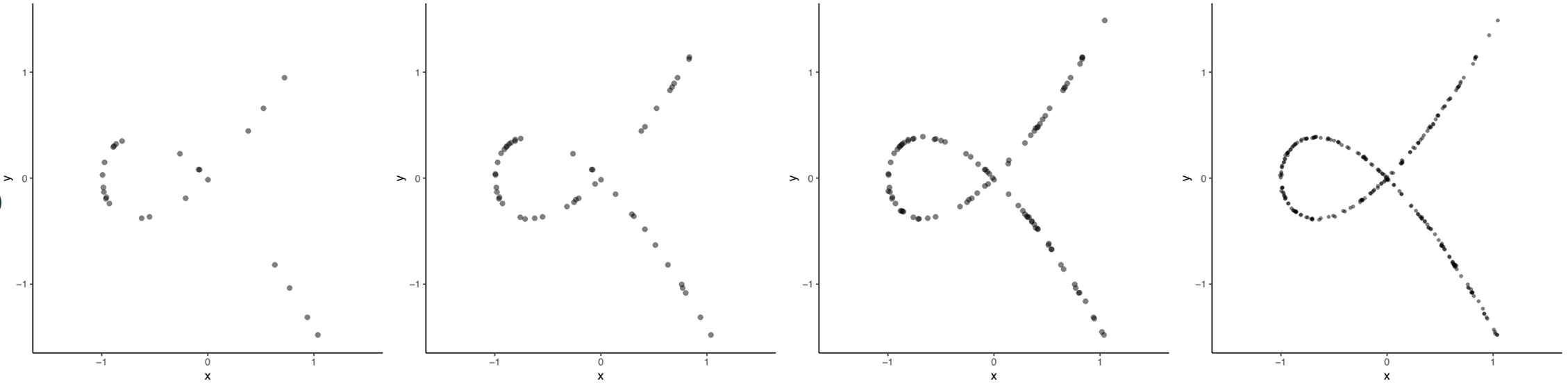
Interfaces : R, Julia, Python, CLI, …

Many chains can be run in parallel

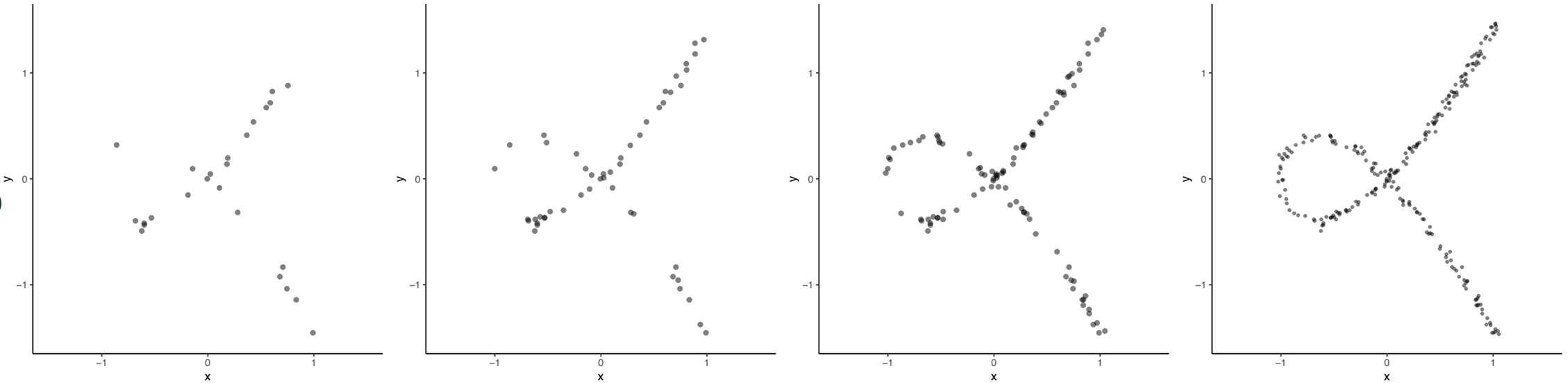
Examples

n = 25 | n = 50 | n = 100 | n = 250

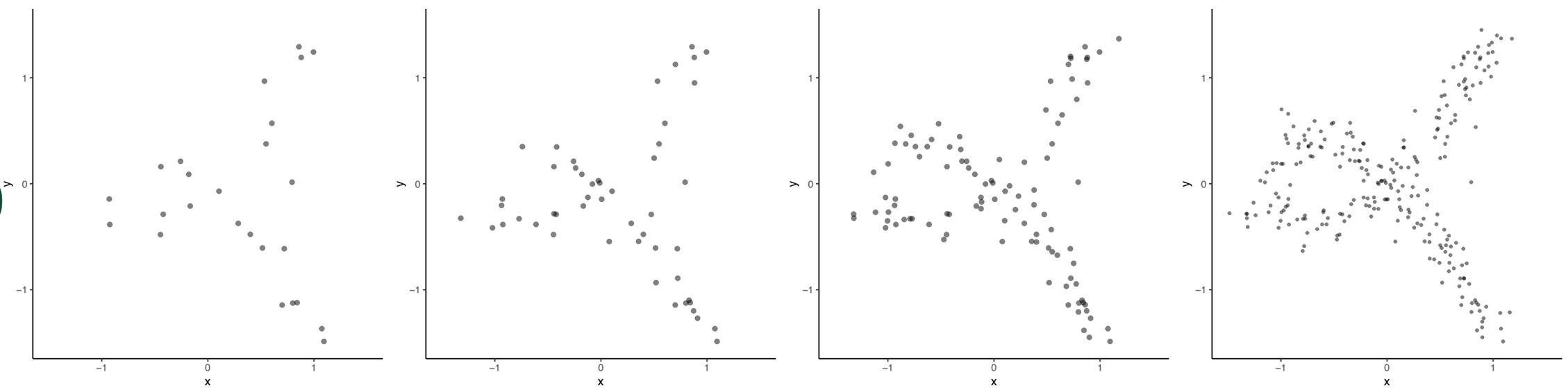
σ = .005



σ = .025

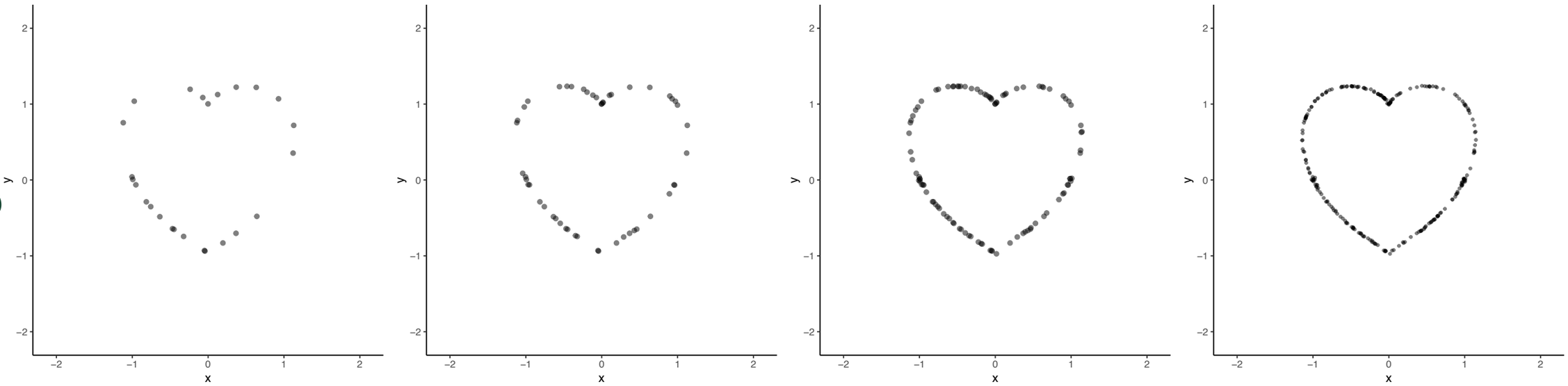


σ = .100

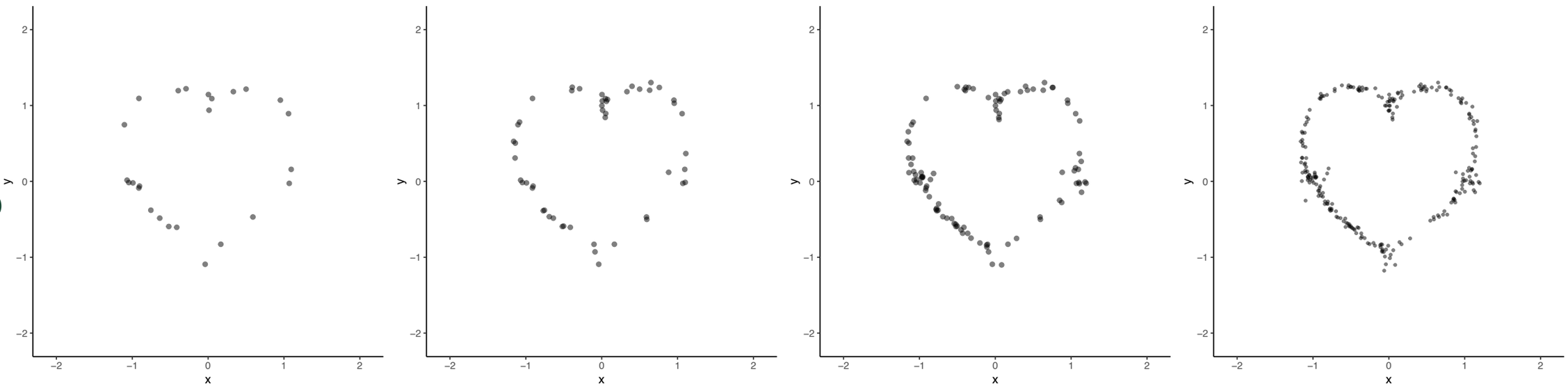


n = 25 | n = 50 | n = 100 | n = 250

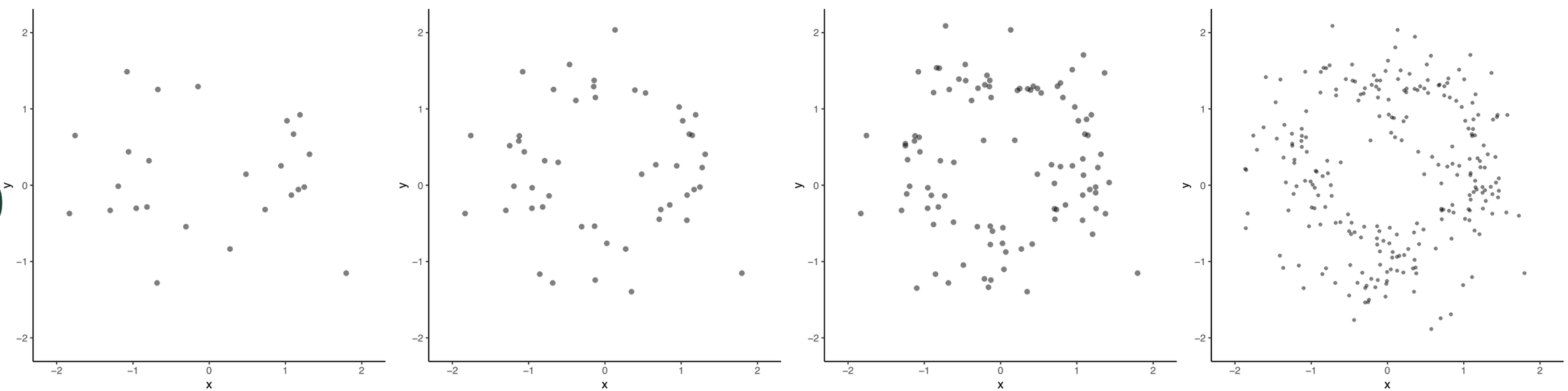
σ = .005



σ = .025

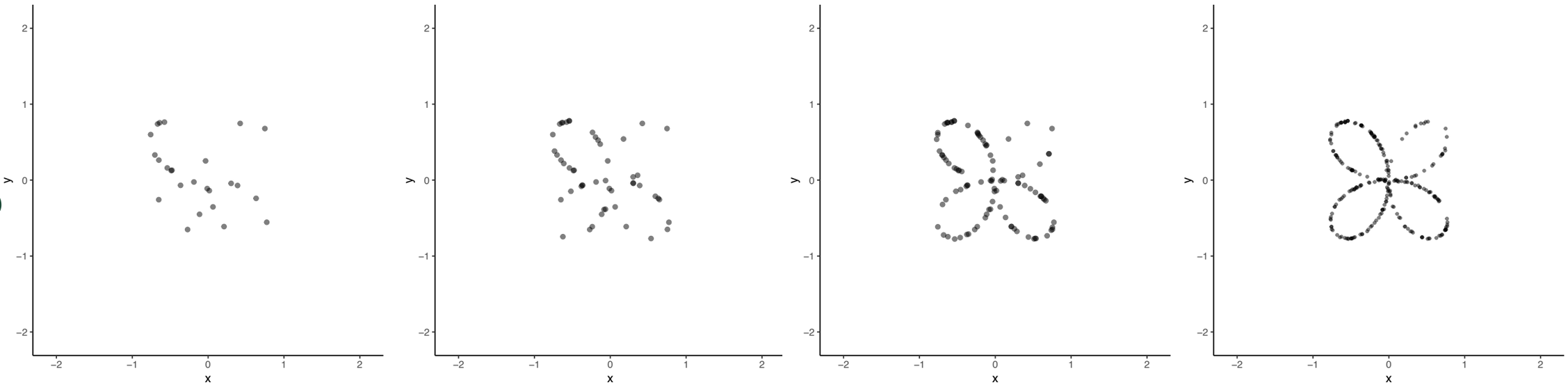


σ = .100

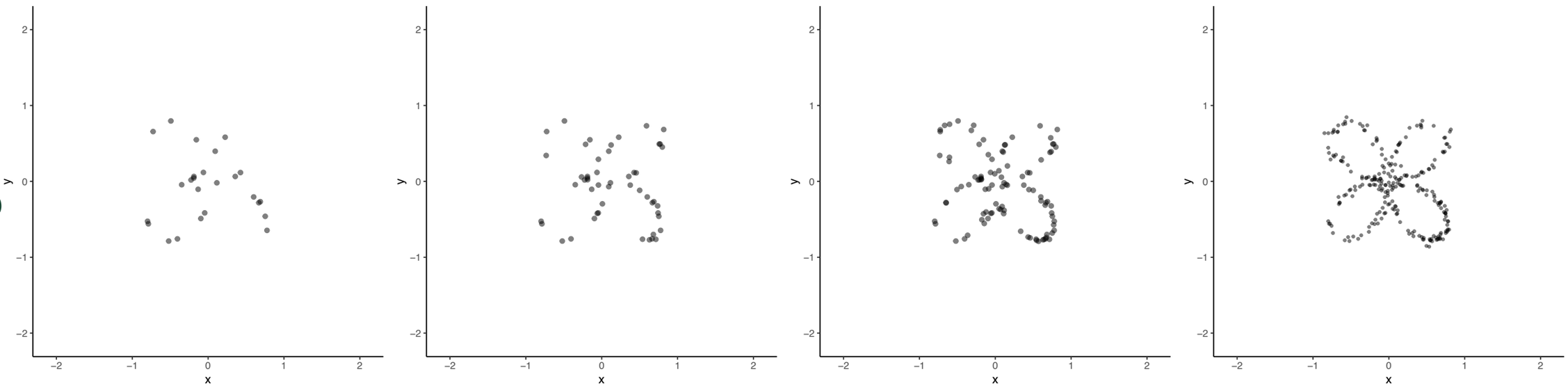


n = 25 | n = 50 | n = 100 | n = 250

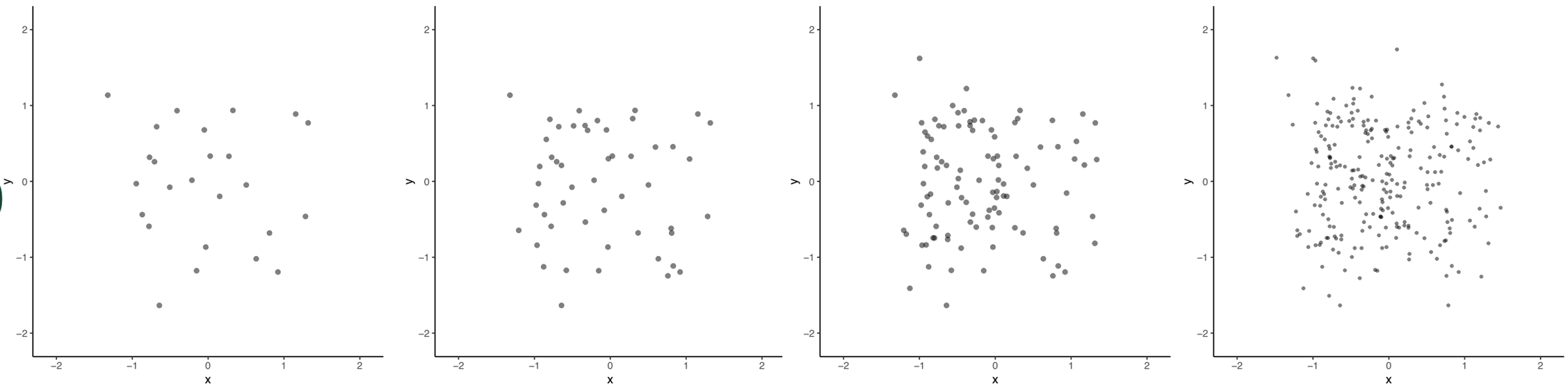
σ = .005



σ = .025

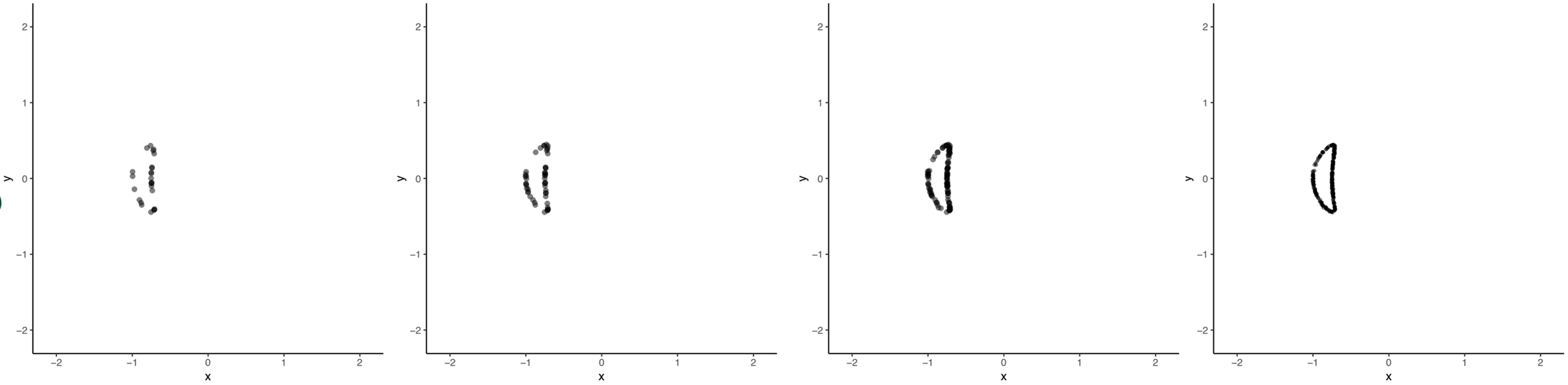


σ = .100

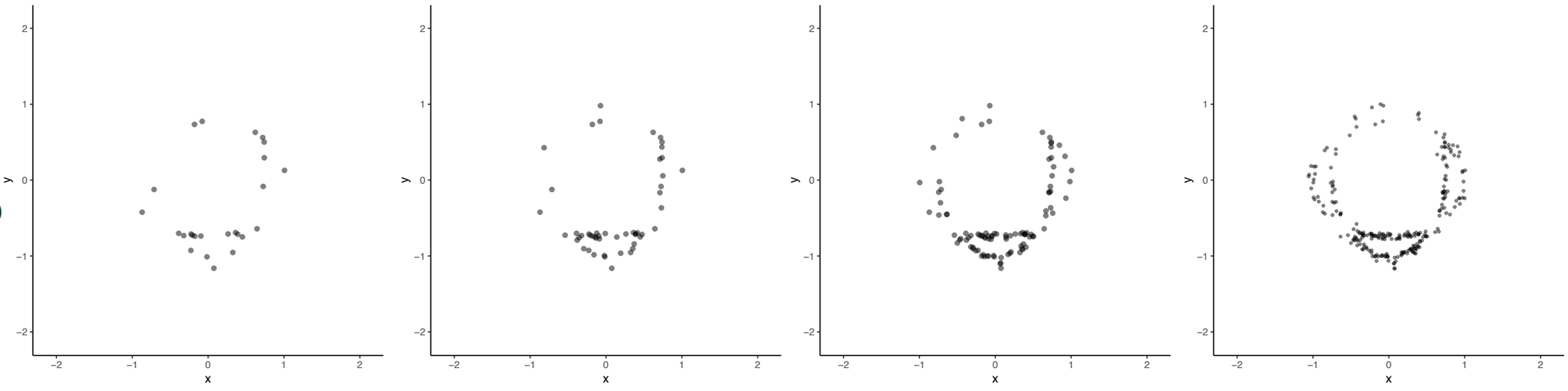


n = 25 | n = 50 | n = 100 | n = 250

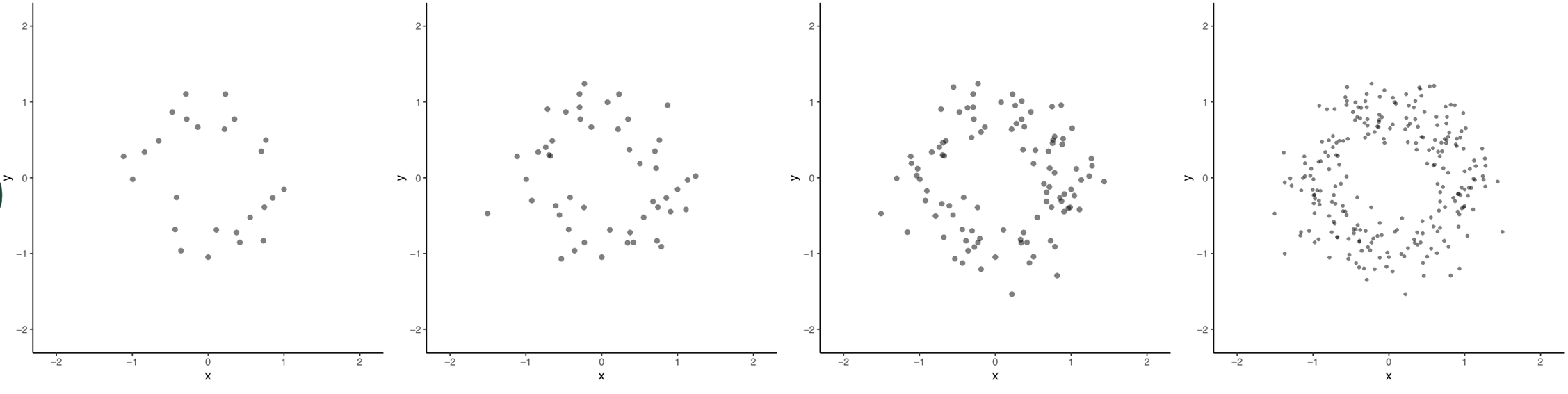
σ = .005



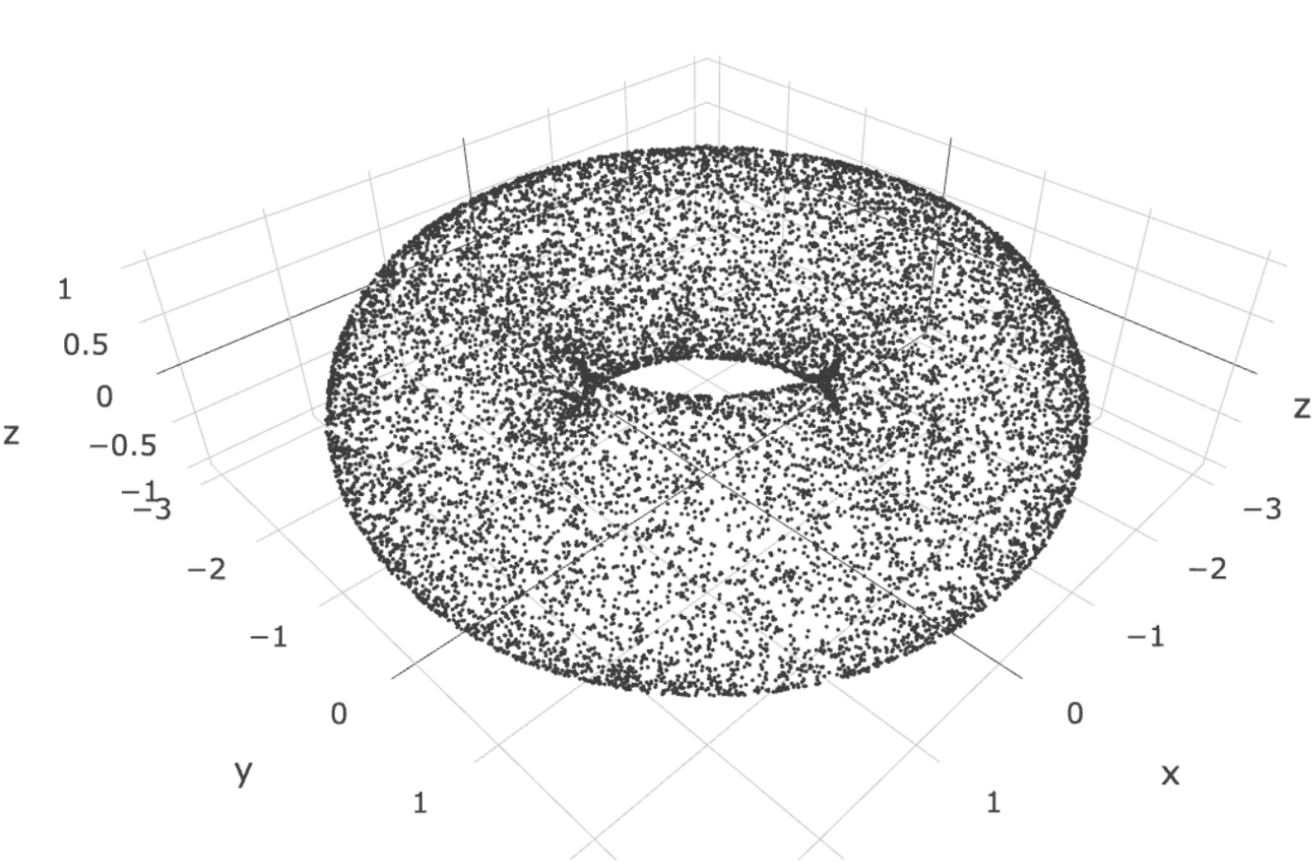
σ = .025

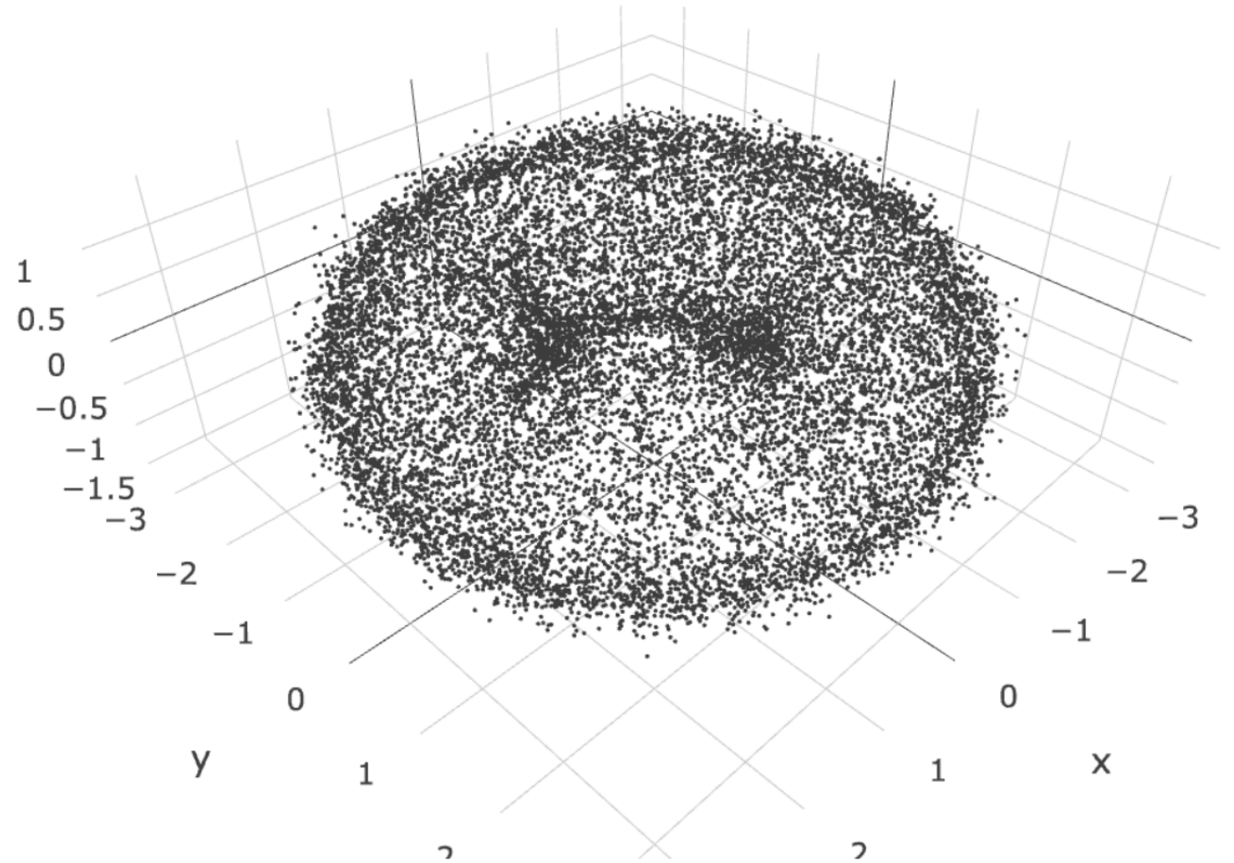


σ = .100



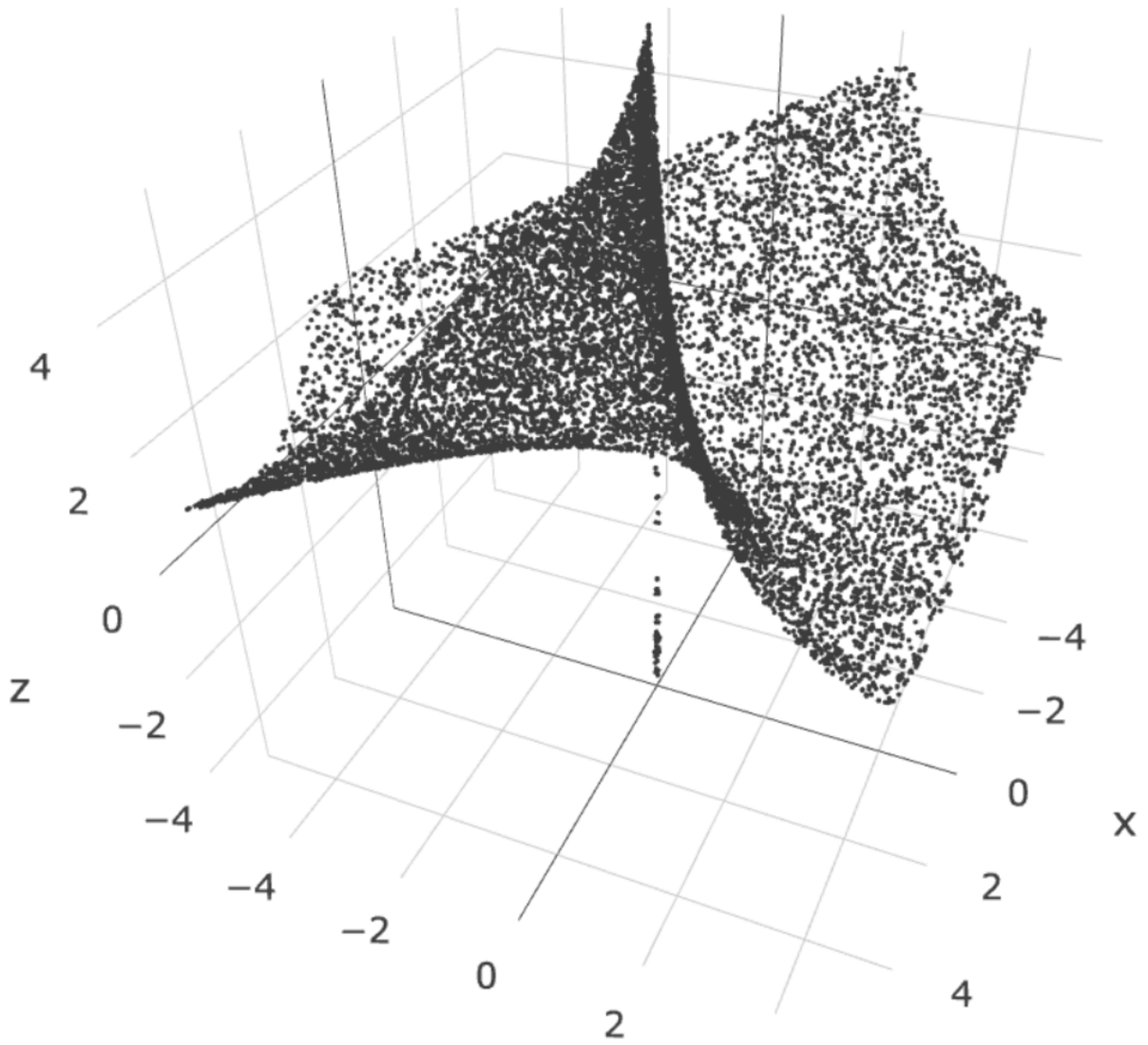
VN(torus, σ = .005/.100); 2000 points

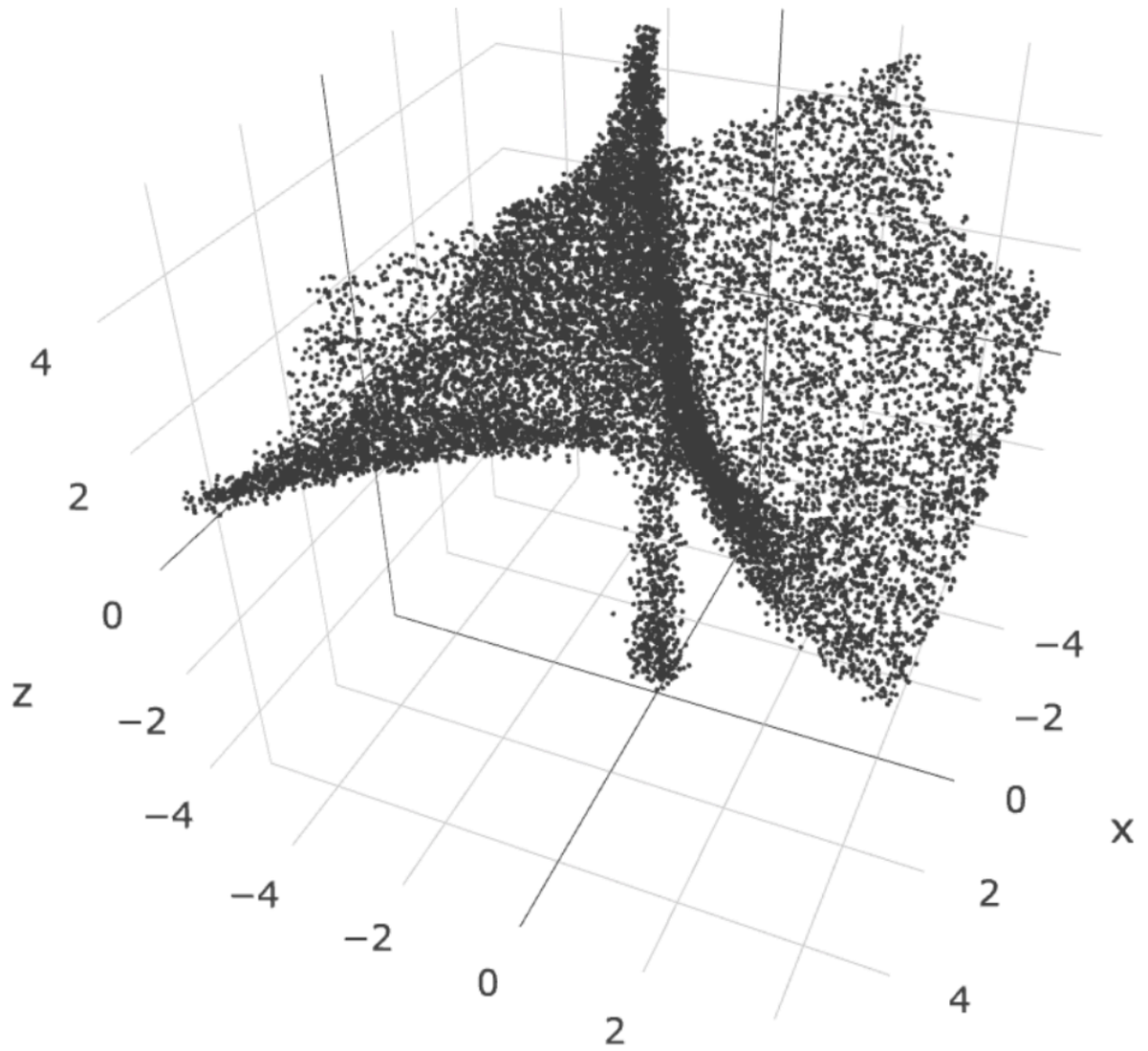




Baylor University

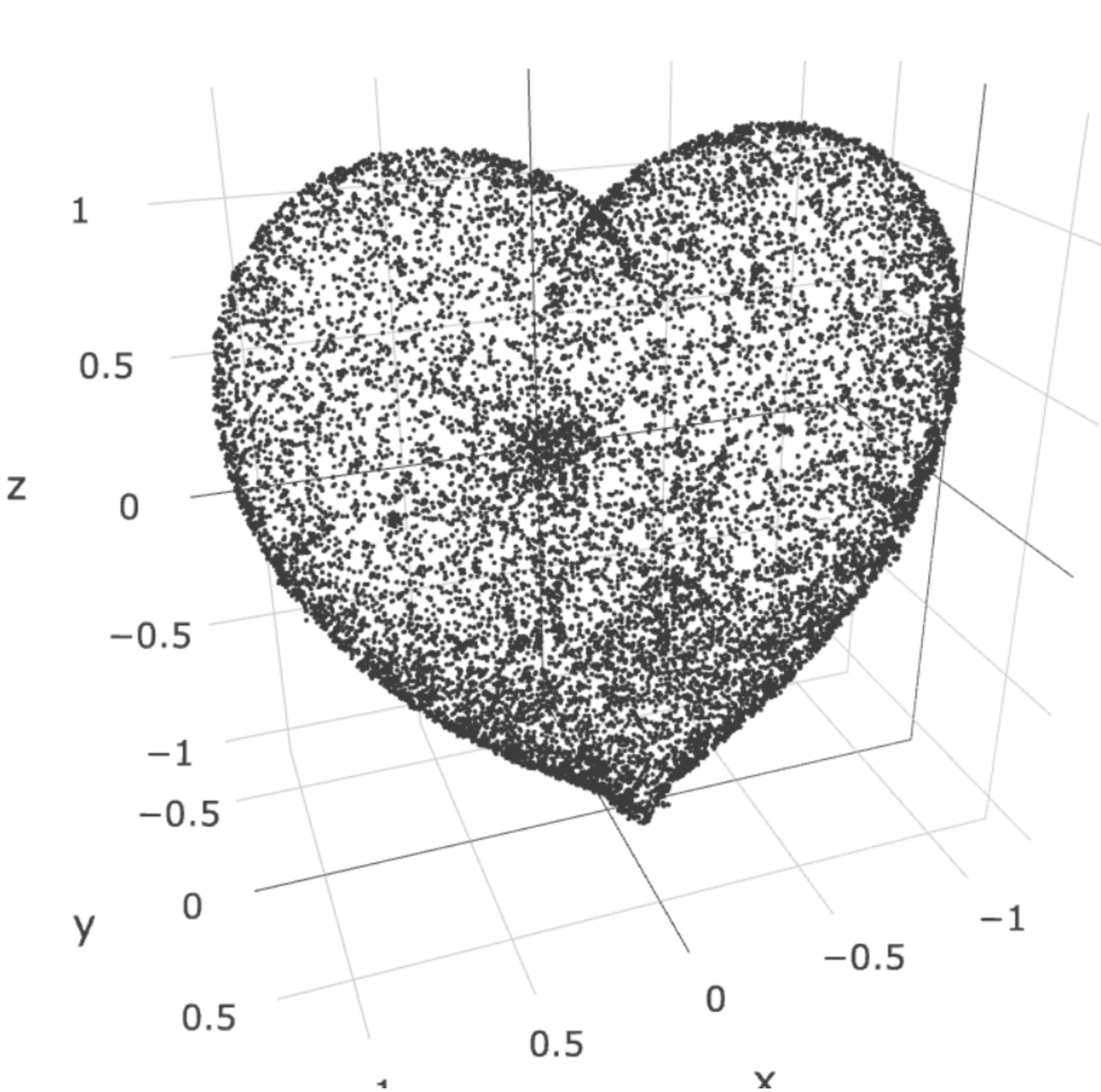
VN(whitney, σ = .010/.100); 2000 points

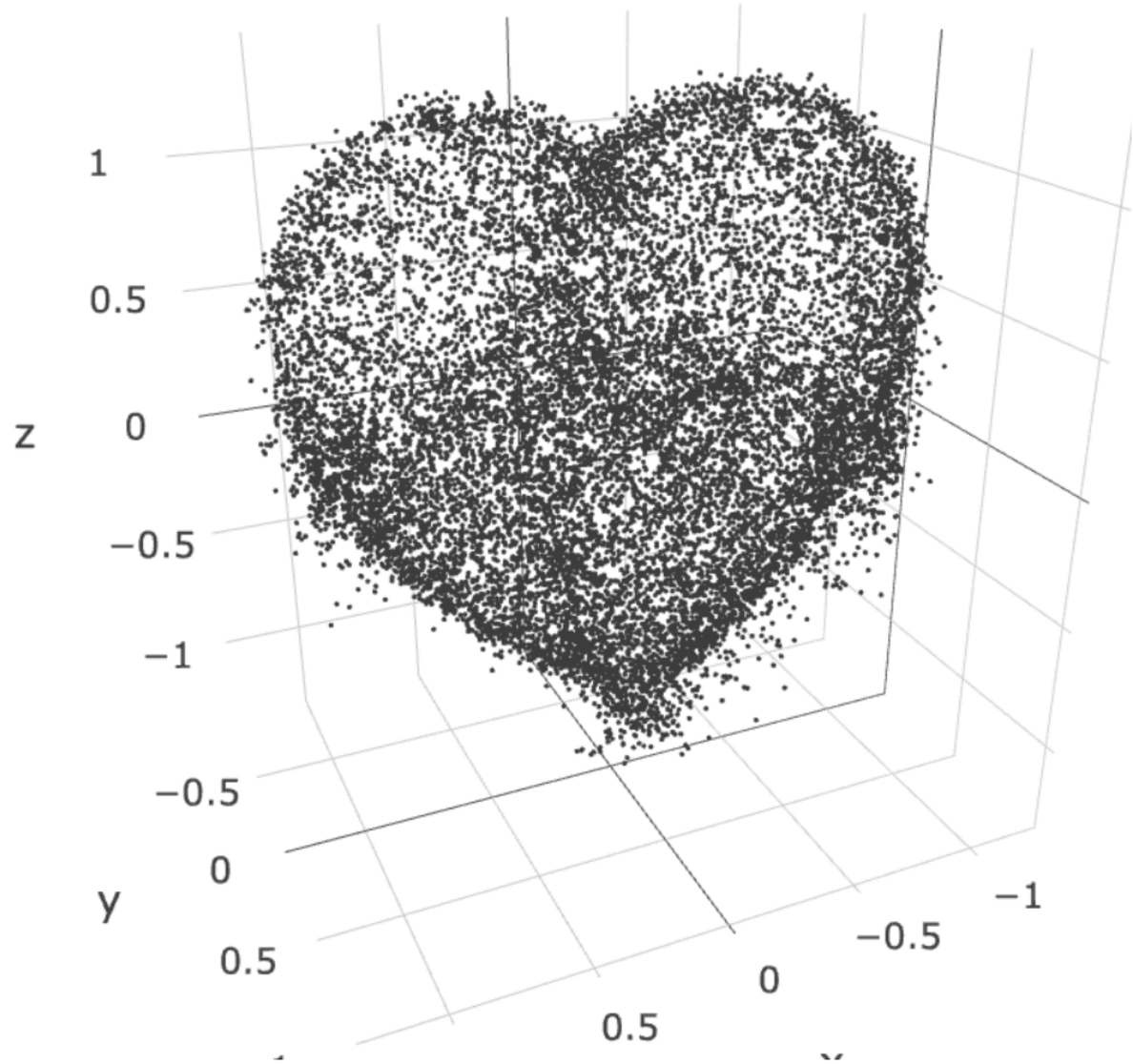




Baylor University

VN(3d heart, σ = .005/.025); 2000 points





Other thoughts

For points on variety, we can use endgames from NAG

To move a point $\boldsymbol{x}$ to a point $\boldsymbol{x}_V$ on the variety, track

$$\boldsymbol{H}(\boldsymbol{x}_V, \boldsymbol{\lambda}; t) = \begin{bmatrix} \boldsymbol{g}(\boldsymbol{x}_V) - t\boldsymbol{g}(\boldsymbol{x}) \\ \lambda_0(\boldsymbol{x}_V - \boldsymbol{x}) + \sum_{i=1}^{k} \lambda_i \nabla g_i(\boldsymbol{x}_V) \end{bmatrix} = \boldsymbol{0}$$

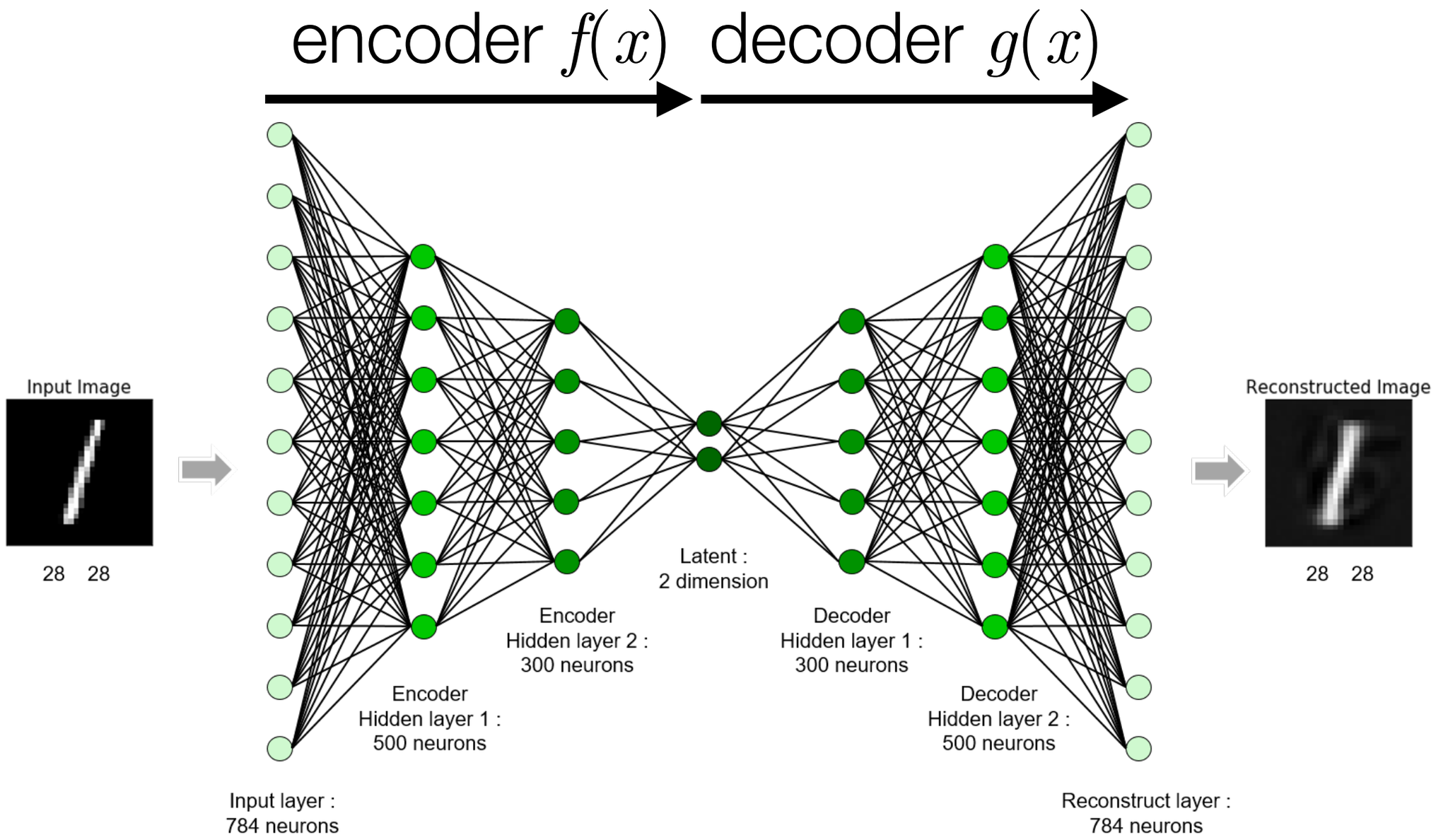
where $\boldsymbol{\lambda} \in \mathbb{P}^m$, from $t = 1$ and $(\boldsymbol{x}, [1, 0, ..., 0])$ to $t = 0$

If $\mathbf{J}_x(\boldsymbol{x})$ is full-rank and $\boldsymbol{x}$ close enough to V, the path is smooth and ends; terminal point $\boldsymbol{x}^* \approx \boldsymbol{x}_V$ can be computed efficiently

Endgames mitigating curvature can also be used

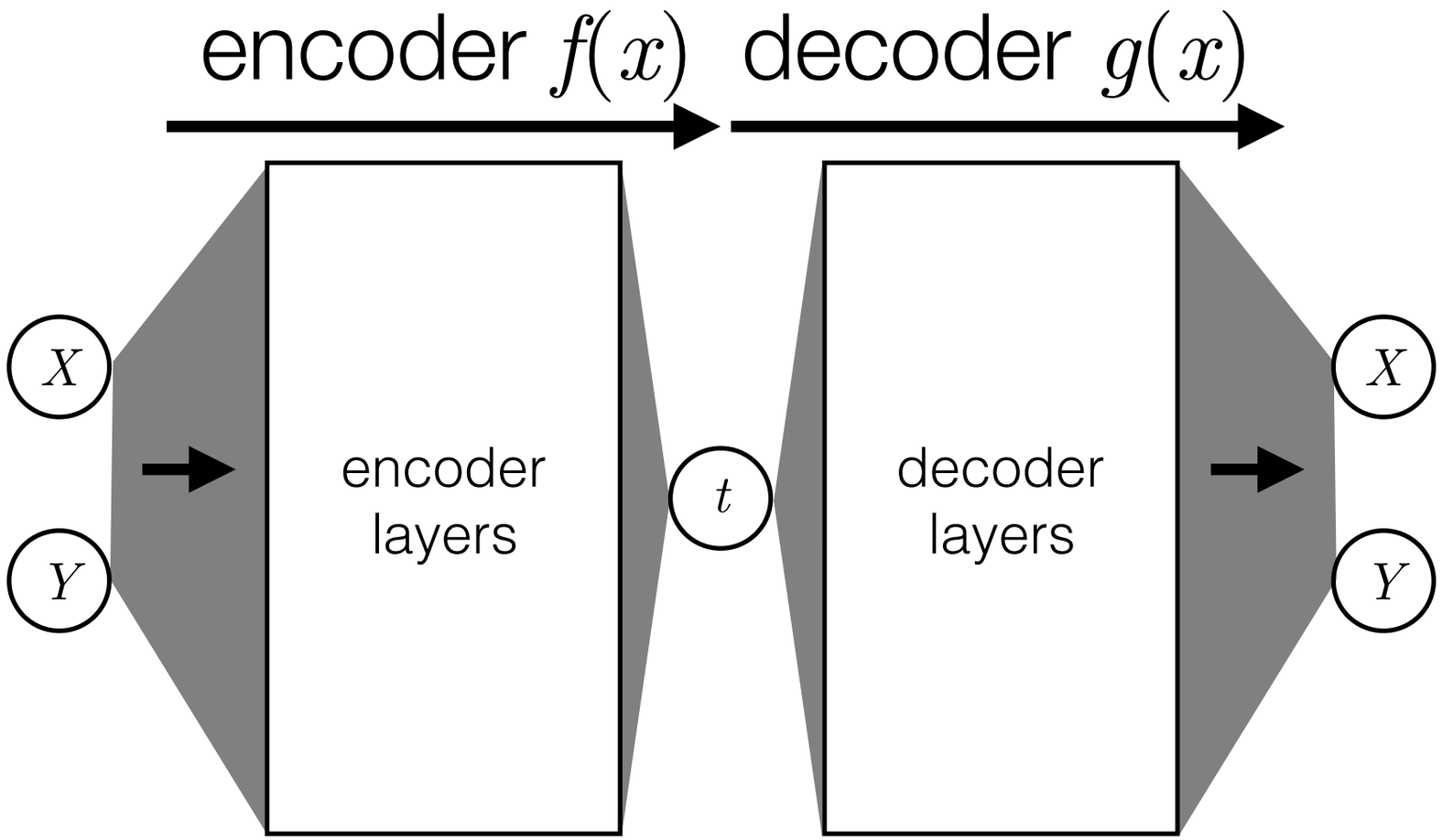
Baylor University

Autoencoders are neural networks commonly used for nonlinear dimension reduction

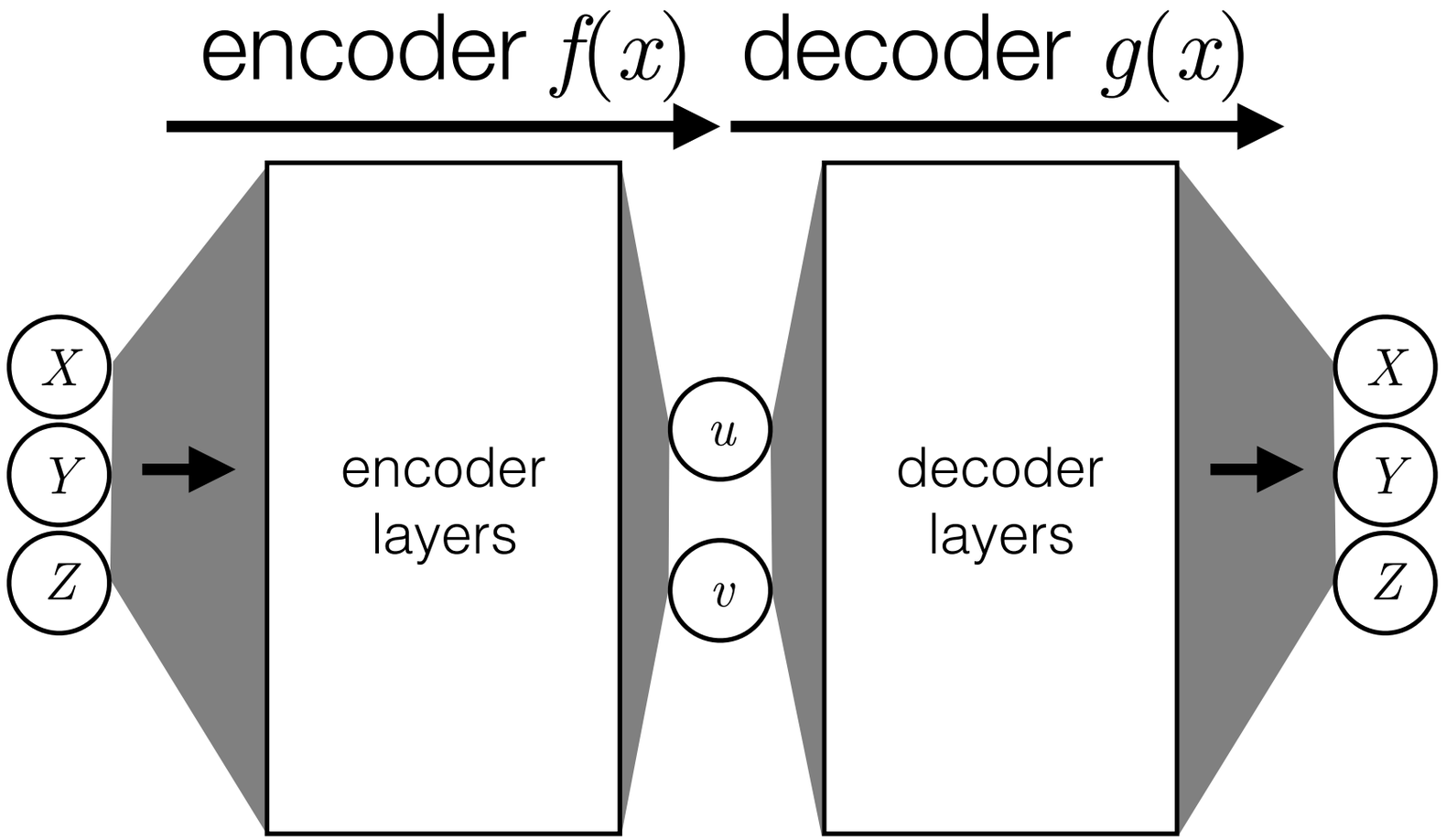


Attribution for image: Prof. Seungchul Lee, iSystems Design Lab, http://isystems.unist.ac.kr/, UNIST

Autoencoders are neural networks commonly used for nonlinear dimension reduction

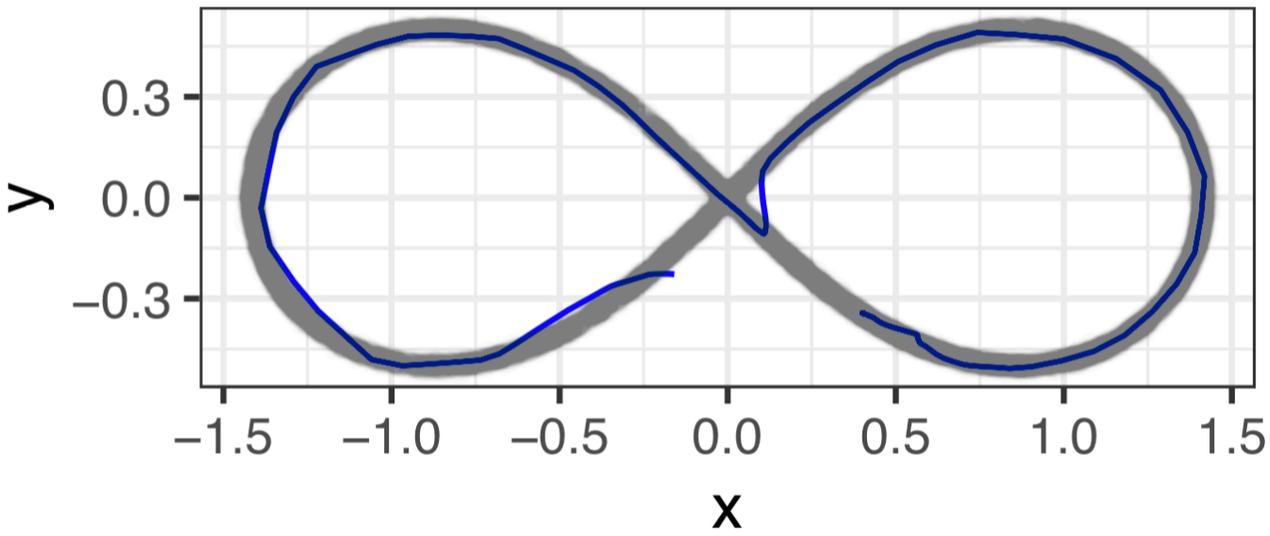


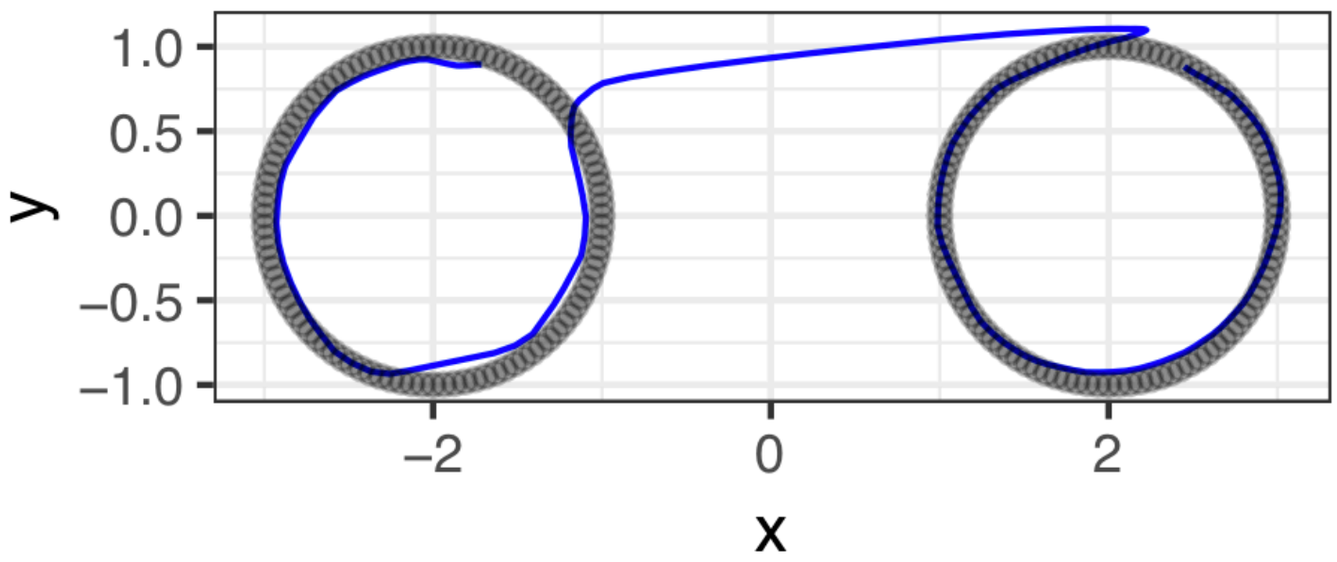
Autoencoders are neural networks commonly used for nonlinear dimension reduction

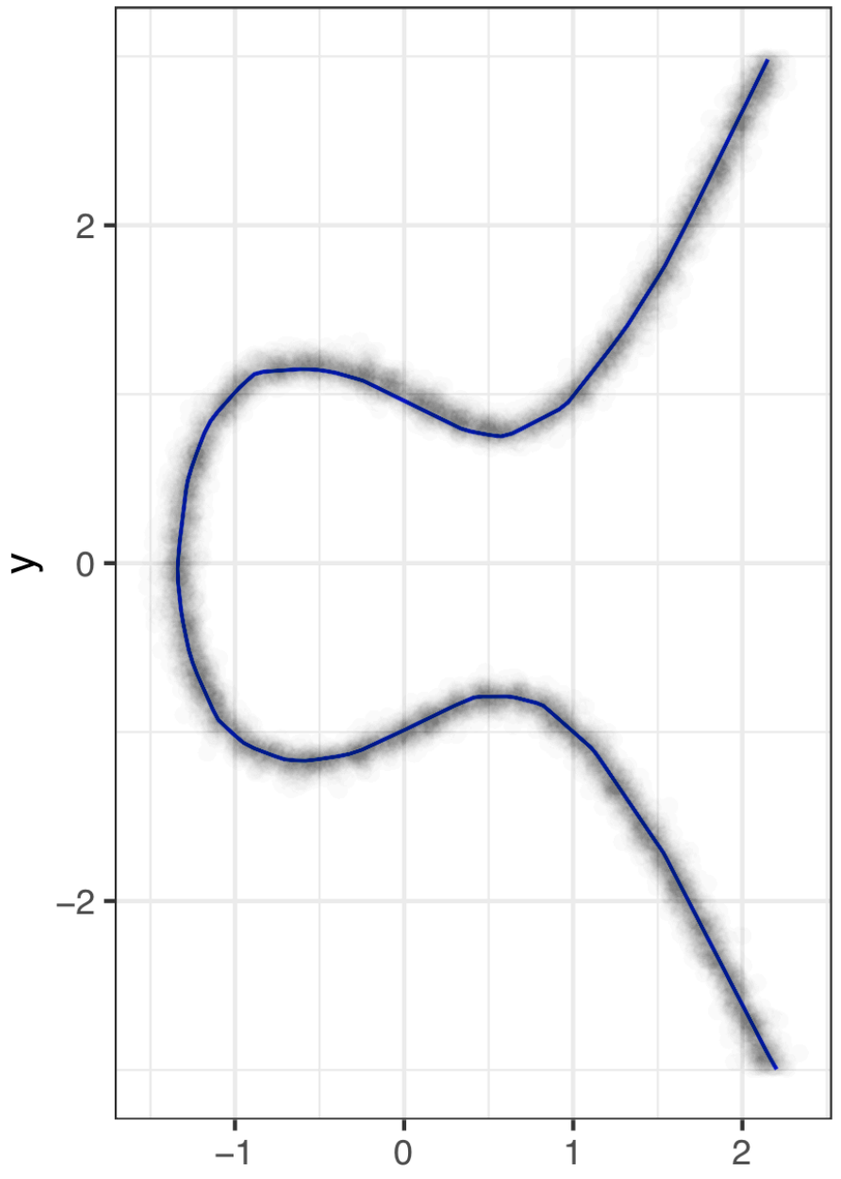


Baylor University

Autoencoders are neural networks commonly used for nonlinear dimension reduction







Attribution: Jerry Ma, Baylor

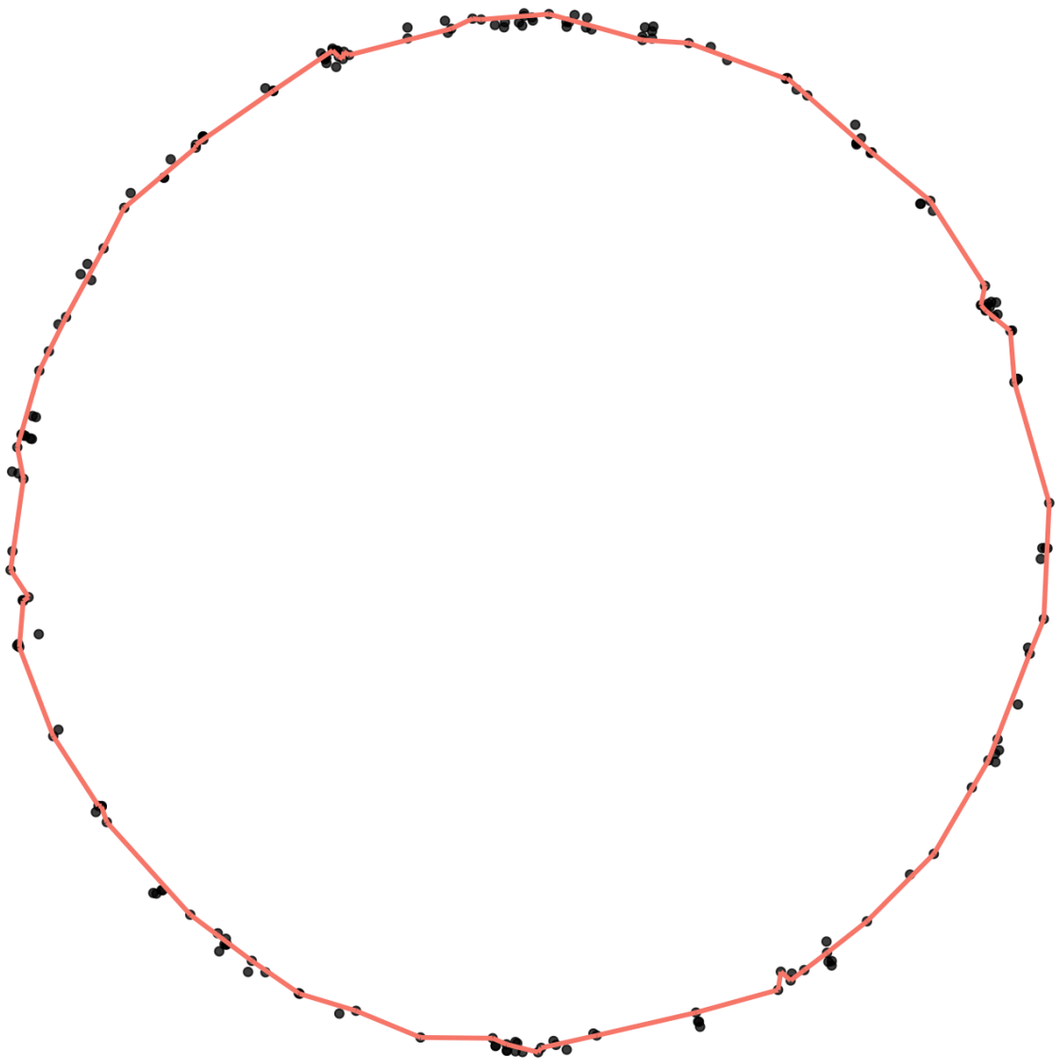
Points generated can be used to test TDA techniques

Polynomial(s) → points → persistent homology (or others)

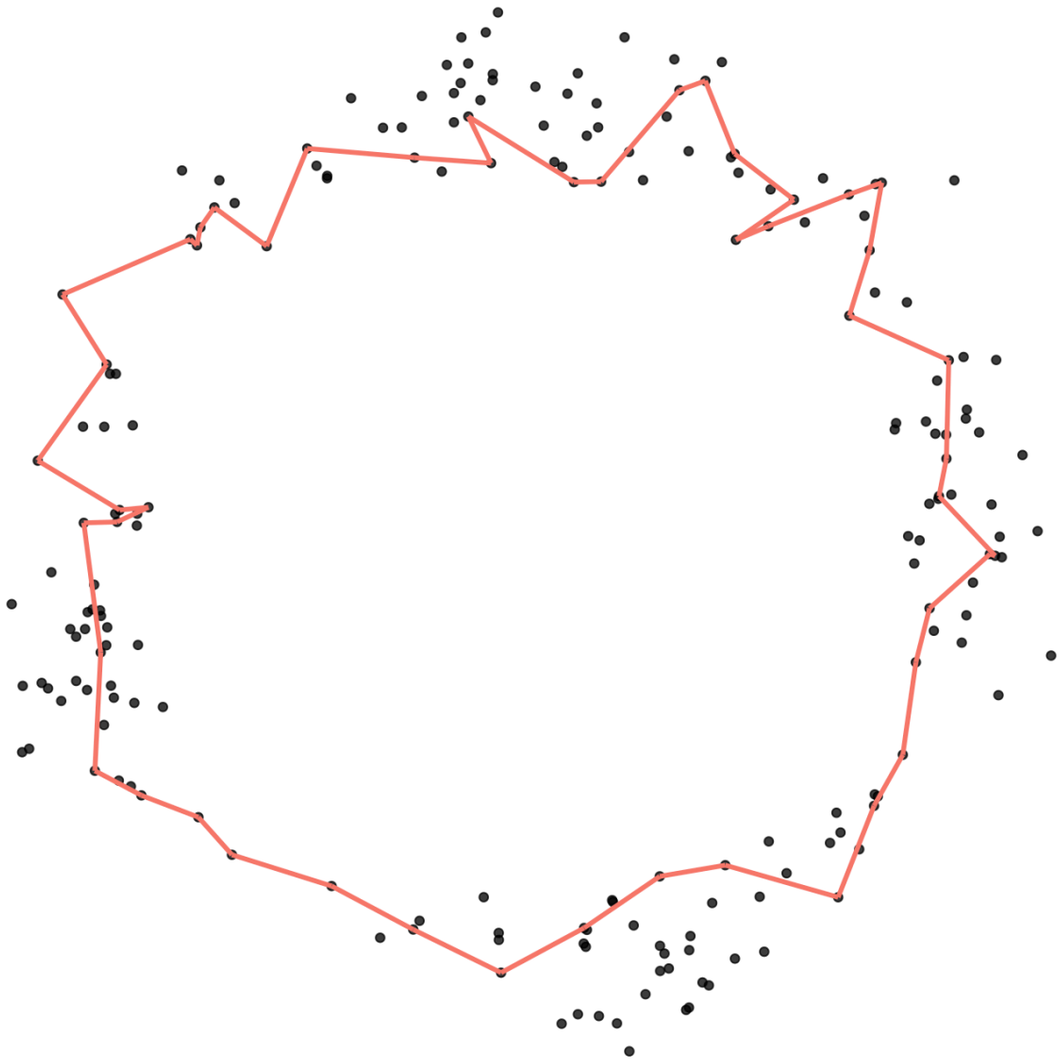
Topological data analysis

Baylor University

σ = .01

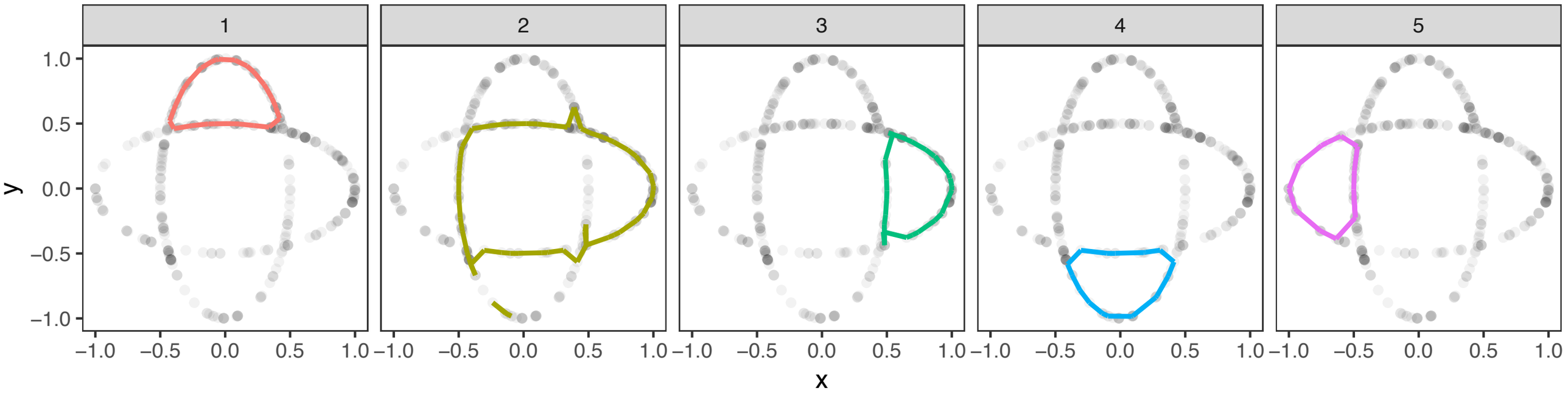


σ = .1



Points generated can be used to test TDA techniques

Polynomial(s) → points → persistent homology (or others)



This also feeds back to trying to understand the topology of the underlying variety itself

Empirically the strategy seems to work well

Disconnected components are best found by initializing multiple chains with dispersed initial values

Singularities manifest as over-dispersed regions

σ cannot be set too large

Great references:

Betancourt, M. "A Conceptual Introduction to Hamiltonian Monte Carlo." arXiv. (2018)

Neal, R. "MCMC Using Hamiltonian Dynamics" in Handbook of Markov Chain Monte Carlo. Eds. S. Brooks, A. Gelman, G. Jones, X. Meng. (2011)

Thank you!!

www.kahle.io

This material is based upon work supported by the National Science Foundation under Grant Nos. 1622449 and 1622369.