STAT 547E: LECTURE 1

INTRODUCTION

Saifuddin Syed

COURSE OUTLINE

- **Logistics:** 1.5 credit topics Graduate course.
  - 12 Lectures over 6 weeks
  - MW 16:00—17:30 in ESB 4192
- **Website:**
  - Content posted on the course website:

    `www.saifsyed.com/teaching/stat547e.html`
  - Canvas for announcements, discussion and submit assignments
- **Evaluation:** 3 Assignments 50% + Project 50%
  - 3 Assignments (~1 week each) due Week 3, 5, 7
  - Course project (~5 page report) + presentation due week 8
- **AI policy:** This is a grad course, and we are all adults.
- **Auditors:** Please formally audit!
- We are going to figure this out together, feedback is always welcome!

MOTIVATION

MOTIVATION

- Black holes are the largest, most powerful objects in our galaxy.

MOTIVATION

- Black holes are the largest, most powerful objects in our galaxy.
- Most galaxies have a supermassive black hole at the center

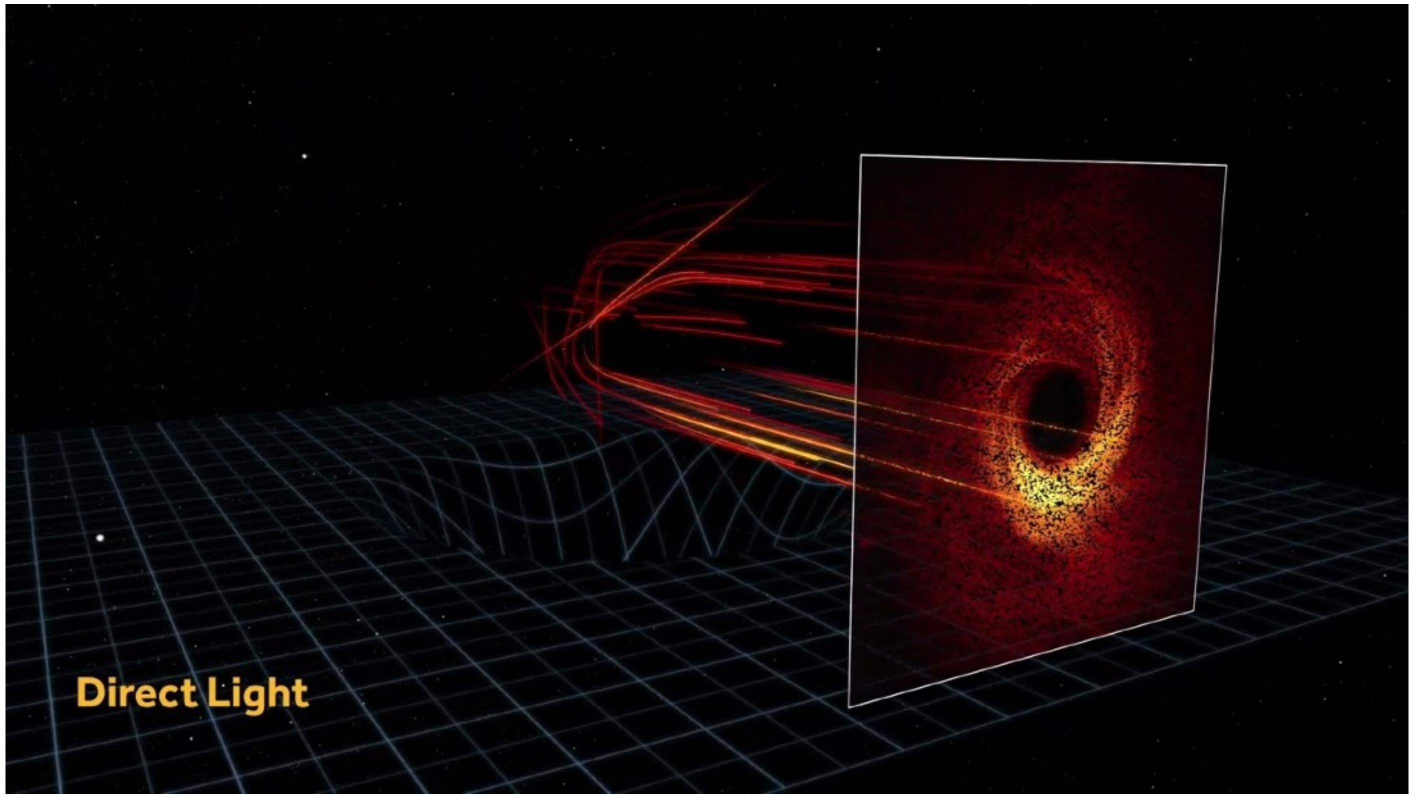
MOTIVATION

- Black holes are the largest, most powerful objects in our galaxy.
- Most galaxies have a supermassive black hole at the center
- They bend and even trap light if you get too close but some escapes

MOTIVATION

- Black holes are the largest, most powerful objects in our galaxy.
- Most galaxies have a supermassive black hole at the center
- They bend and even trap light if you get too close but some escapes

https://www.youtube.com/watch?v=4i65CYm6Sak



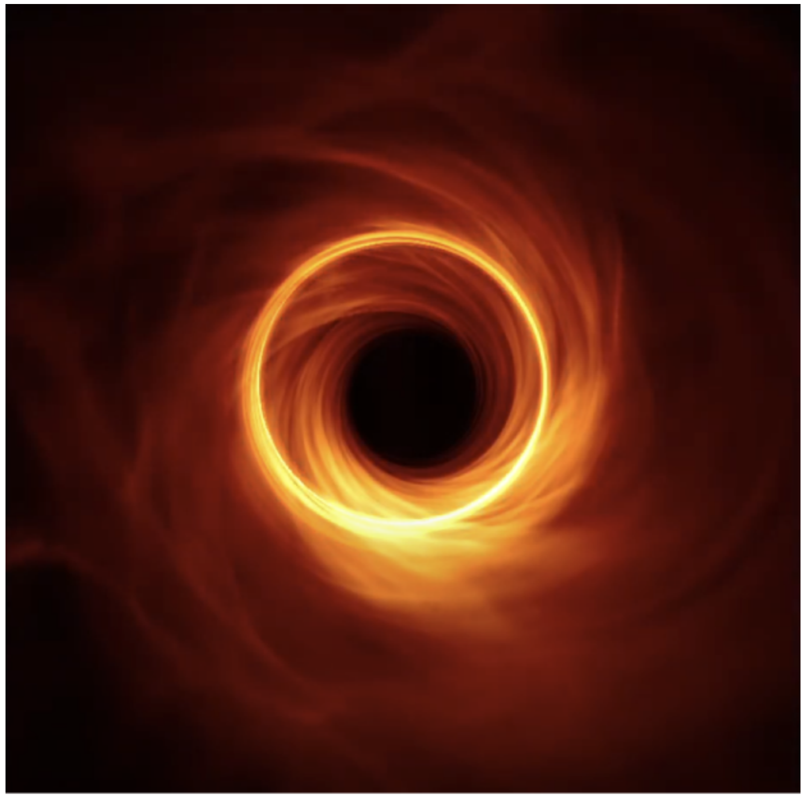
MOTIVATION

MOTIVATION

- **Example:** Suppose we want to infer an image $x \in \mathbb{X}$ of a black hole.

MOTIVATION

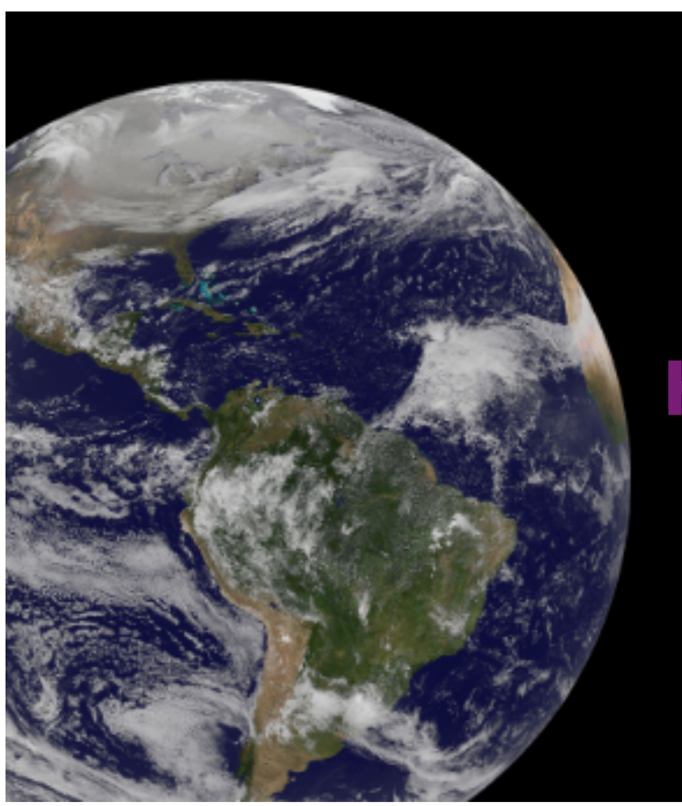
- **Example:** Suppose we want to infer an image $x \in \mathbb{X}$ of a black hole.
  - We want to capture a light that leaves the orbit of a black hole



MOTIVATION

- **Example:** Suppose we want to infer an image $x \in \mathbb{X}$ of a black hole.
  - We want to capture a light that leaves the orbit of a black hole

- **Problem:** Black holes are far…



~10^{-8} degrees

~54×10^{6} lightyears

~10×10^{21} Km

MOTIVATION

- **Example:** Suppose we want to infer an image $x \in \mathbb{X}$ of a black hole.
  - We want to capture a light that leaves the orbit of a black hole
- **Problem:** Black holes are far…
  - Solution: turn the earth into a telescope

~10^{-8} degrees

~54×10^6 lightyears

~10×10^{21} Km

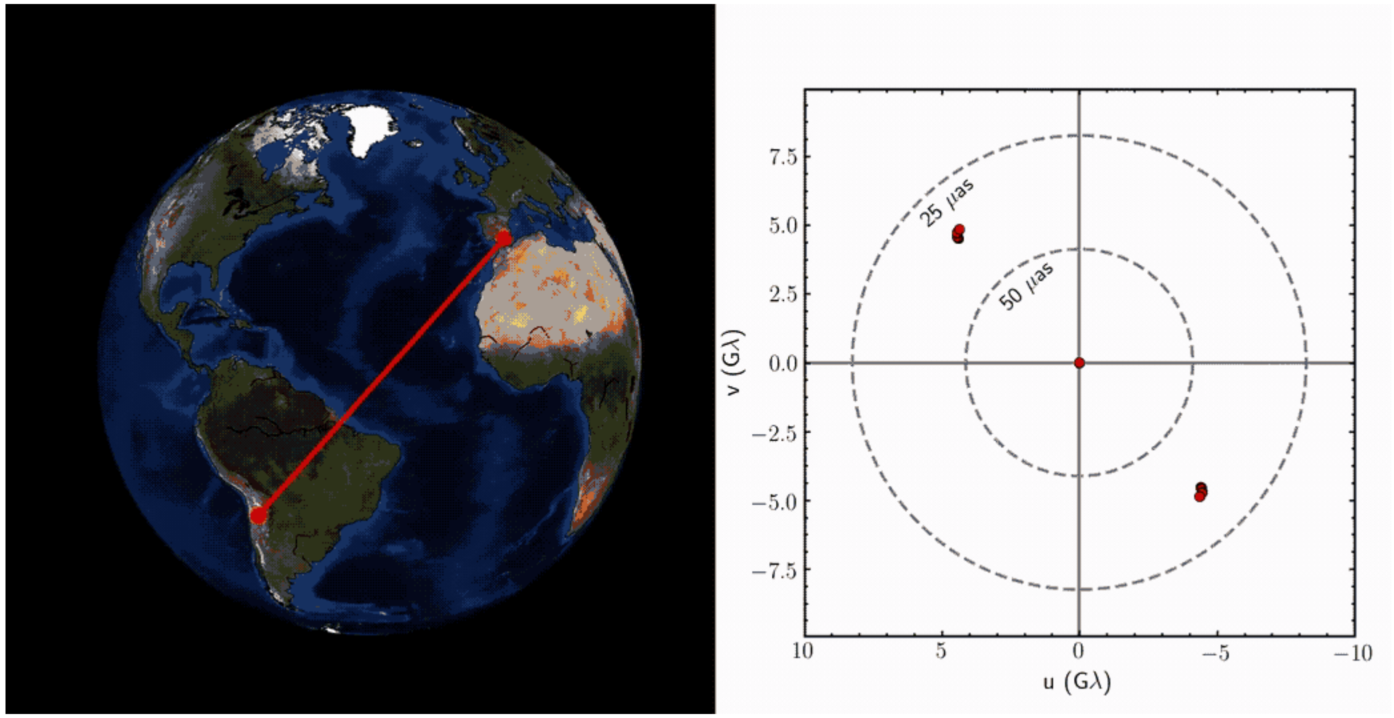
VERY LONG BASELINE INFEROMETRY

VERY LONG BASELINE INFEROMETRY

- Space out mutliple syncronised telescopes around the earth and collect data $y \in \mathbb{Y}$

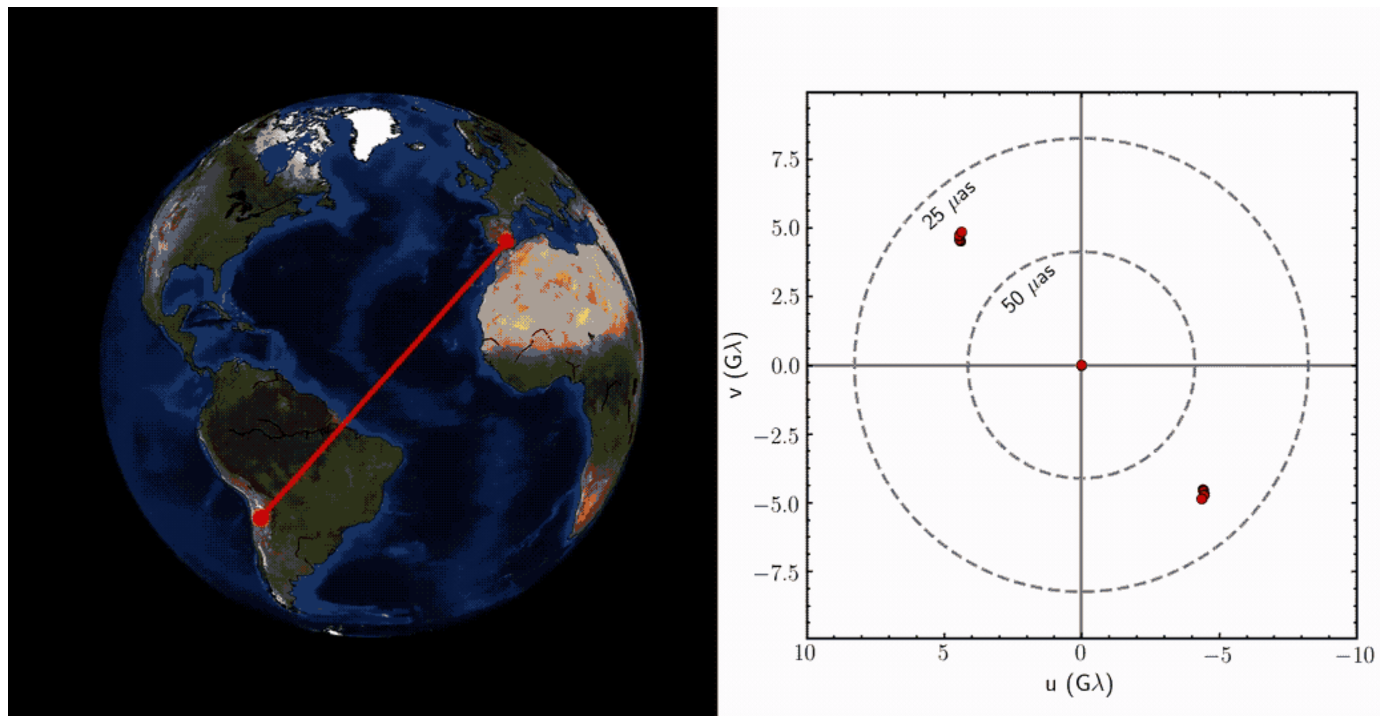
VERY LONG BASELINE INFEROMETRY

- Space out mutliple syncronised telescopes around the earth and collect data $y \in \mathbb{Y}$



VERY LONG BASELINE INFEROMETRY

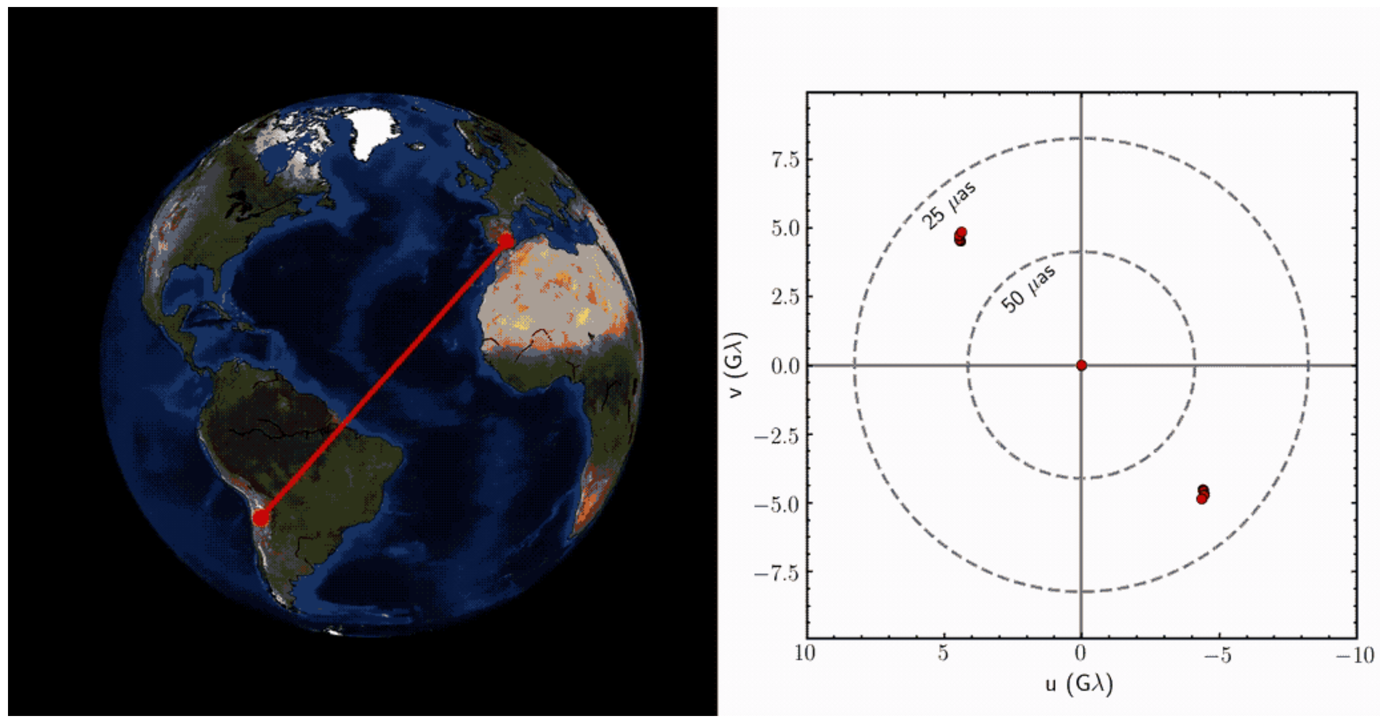
- Space out mutliple syncronised telescopes around the earth and collect data $y \in \mathbb{Y}$



- Model the data $y \in \mathbb{Y}$ using some parameters $x \in \mathbb{X}$ with likelihood $L(y \mid x)$

VERY LONG BASELINE INFEROMETRY

- Space out mutliple syncronised telescopes around the earth and collect data $y \in \mathbb{Y}$

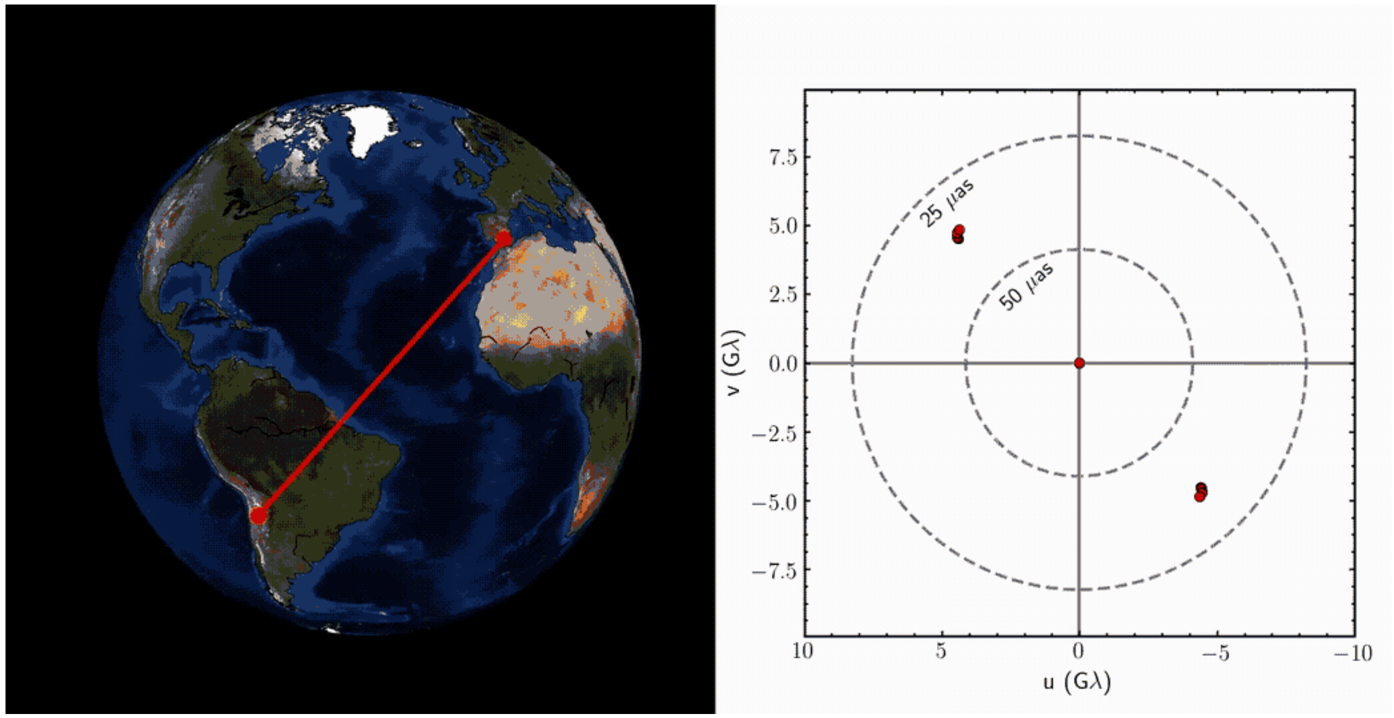


- Model the data $y \in \mathbb{Y}$ using some parameters $x \in \mathbb{X}$ with likelihood $L(y \mid x)$
  - $A : \mathbb{X} \to \mathbb{Y}$ is a non-linear transformation and ϵ is the mearurement error

$$y = A(x) + \epsilon, \quad \epsilon \sim N(0, \Sigma)$$

VERY LONG BASELINE INFEROMETRY

- Space out mutliple syncronised telescopes around the earth and collect data $y \in \mathbb{Y}$



- Model the data $y \in \mathbb{Y}$ using some parameters $x \in \mathbb{X}$ with likelihood $L(y|x)$
    - $A : \mathbb{X} \to \mathbb{Y}$ is a non-linear transformation and ϵ is the mearurement error

$$y = A(x) + \epsilon, \quad \epsilon \sim N(0,\Sigma)$$

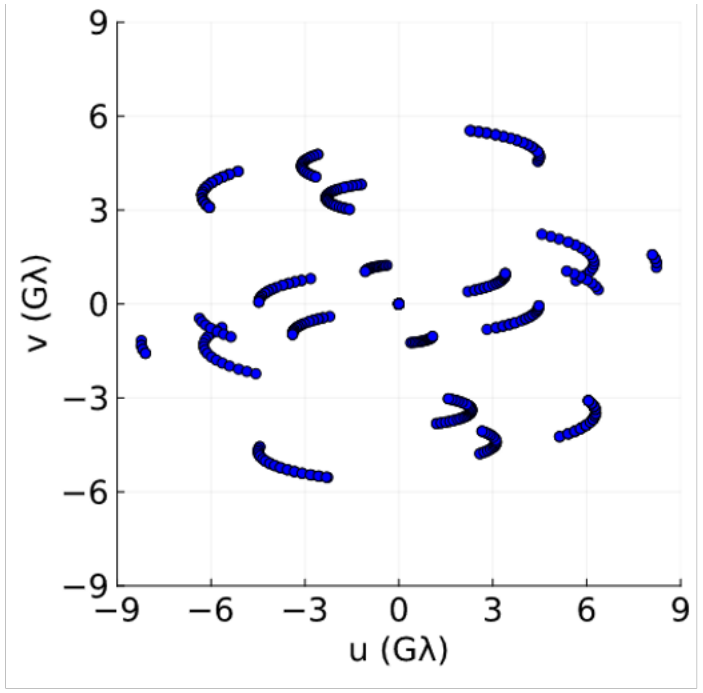
- Given the data, we want to predict what image $x \in \mathbb{X}$ was generated.

$$x = A^{-1}(y - \epsilon)$$

BAYESIAN INVERSE PROBLEMS

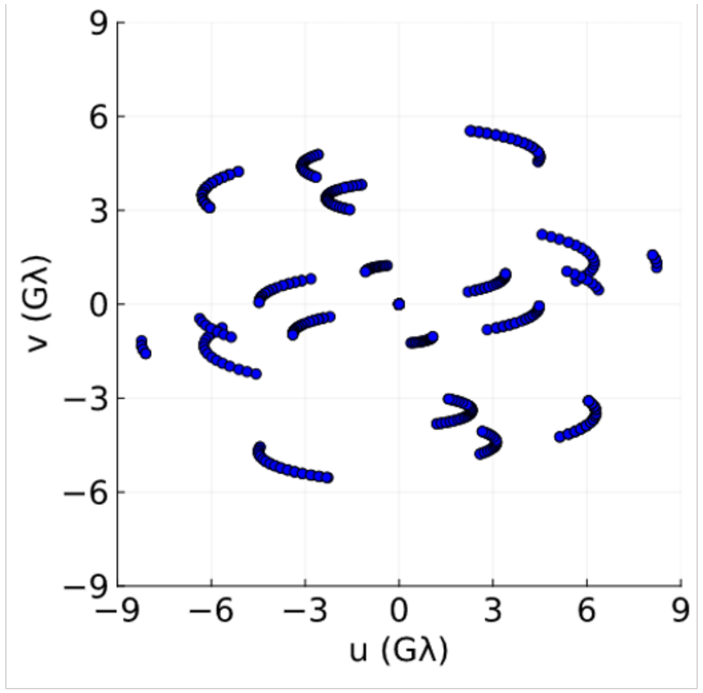
BAYESIAN INVERSE PROBLEMS

- **Problem:** Data is sparse, noisy, temporally corellated, and in Fourier space



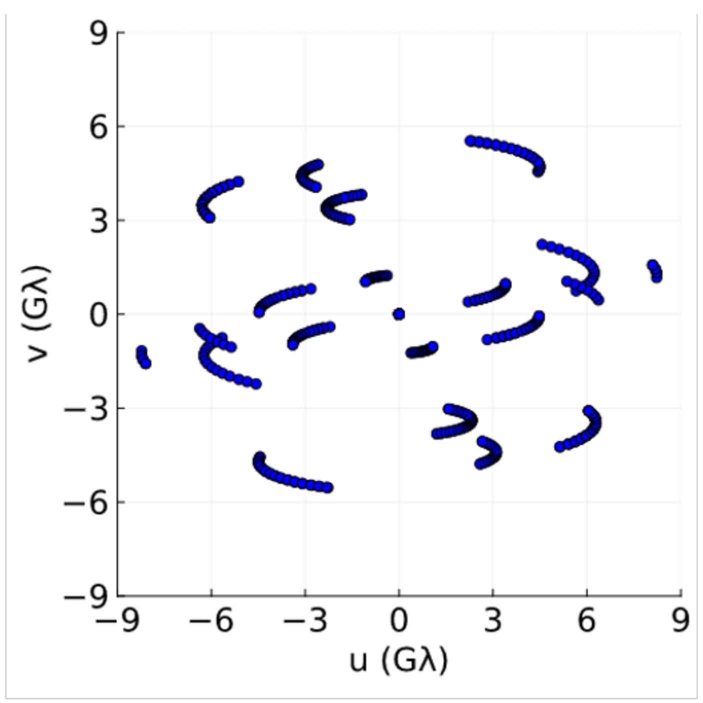
BAYESIAN INVERSE PROBLEMS

- **Problem:** Data is sparse, noisy, temporally corellated, and in Fourier space
    - Inverse might not exist and if it does it might not be unique



BAYESIAN INVERSE PROBLEMS

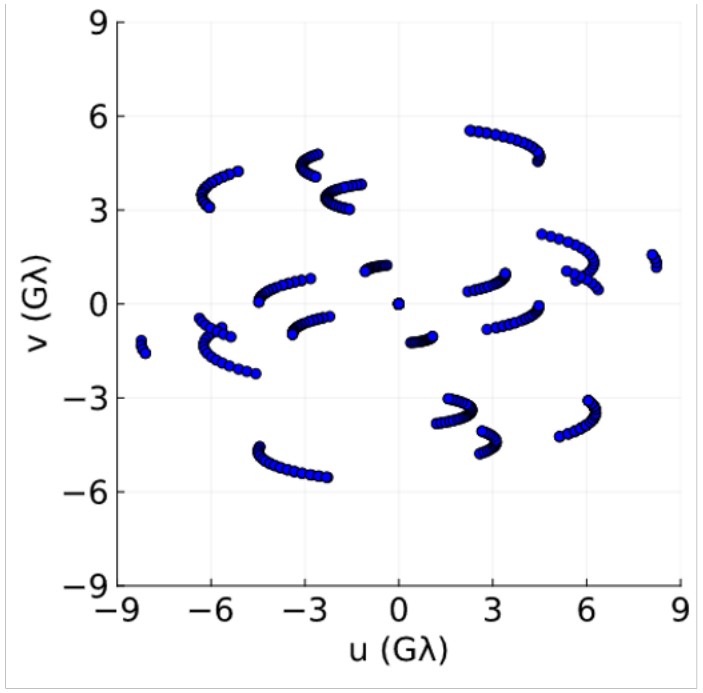
- **Problem:** Data is sparse, noisy, temporally corellated, and in Fourier space
  - Inverse might not exist and if it does it might not be unique
  - Some images are more likely than others



BAYESIAN INVERSE PROBLEMS

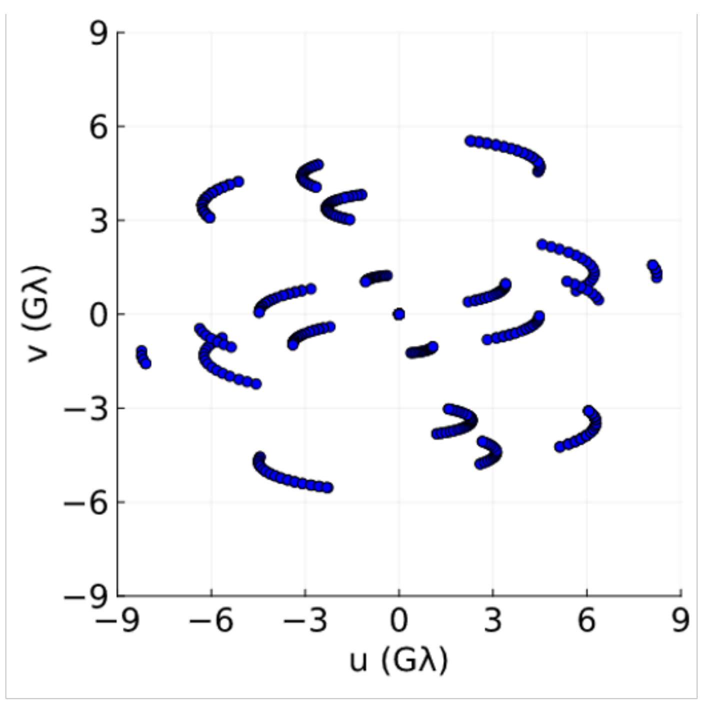
- **Problem:** Data is sparse, noisy, temporally corellated, and in Fourier space
    - Inverse might not exist and if it does it might not be unique
    - Some images are more likely than others

- **Solution:** Introduce a prior $p(x)$ on the space of images $\mathbb{X}$ corresponding to feasible



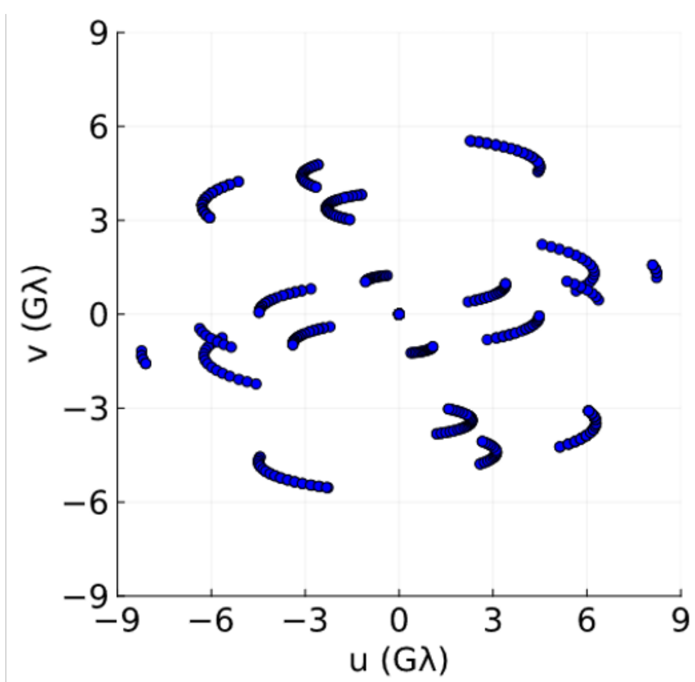
BAYESIAN INVERSE PROBLEMS

- **Problem:** Data is sparse, noisy, temporally corellated, and in Fourier space
  - Inverse might not exist and if it does it might not be unique
  - Some images are more likely than others
- **Solution:** Introduce a prior $p(x)$ on the space of images $\mathbb{X}$ corresponding to feasible
  - e.g. a log-gaussian process with unknown kernel parameters



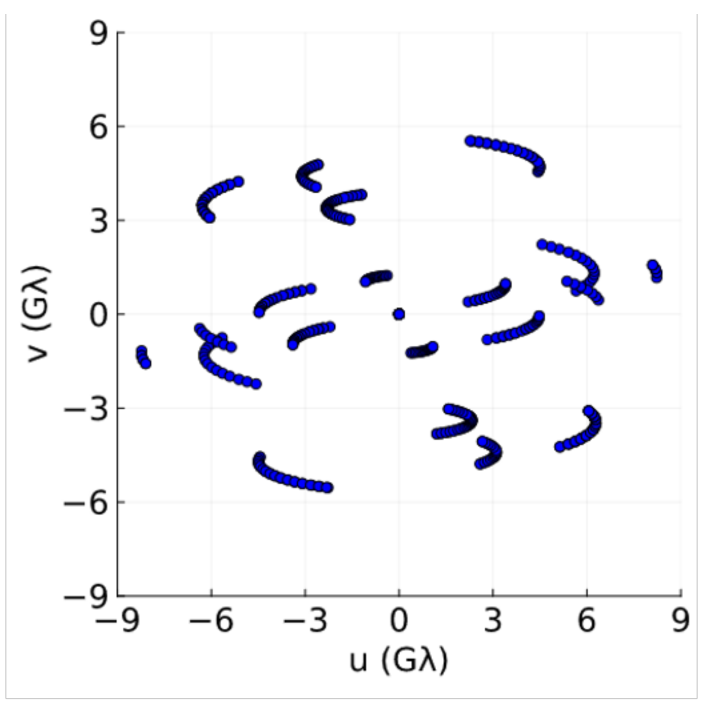
BAYESIAN INVERSE PROBLEMS

- **Problem:** Data is sparse, noisy, temporally corellated, and in Fourier space
  - Inverse might not exist and if it does it might not be unique
  - Some images are more likely than others
- **Solution:** Introduce a prior $p(x)$ on the space of images $\mathbb{X}$ corresponding to feasible
  - e.g. a log-gaussian process with unknown kernel parameters
  - Infer the image from the posterior $p(x \mid y)$



BAYESIAN INVERSE PROBLEMS

- **Problem:** Data is sparse, noisy, temporally corellated, and in Fourier space
  - Inverse might not exist and if it does it might not be unique
  - Some images are more likely than others
- **Solution:** Introduce a prior $p(x)$ on the space of images $\mathbb{X}$ corresponding to feasible
  - e.g. a log-gaussian process with unknown kernel parameters
  - Infer the image from the posterior $p(x \mid y)$



Inference →

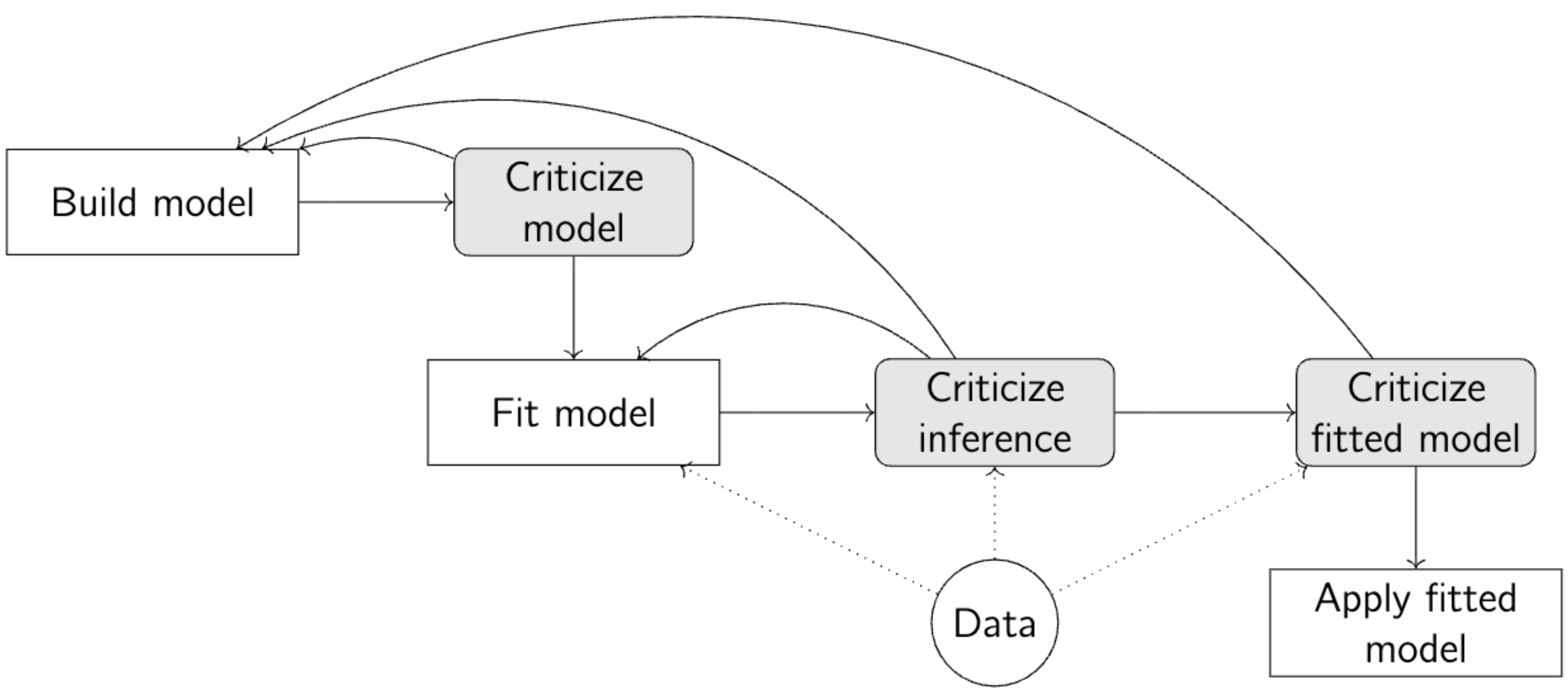
M87, 2019

$\sim p(x \mid y)$

BAYESIAN WORKFLOW

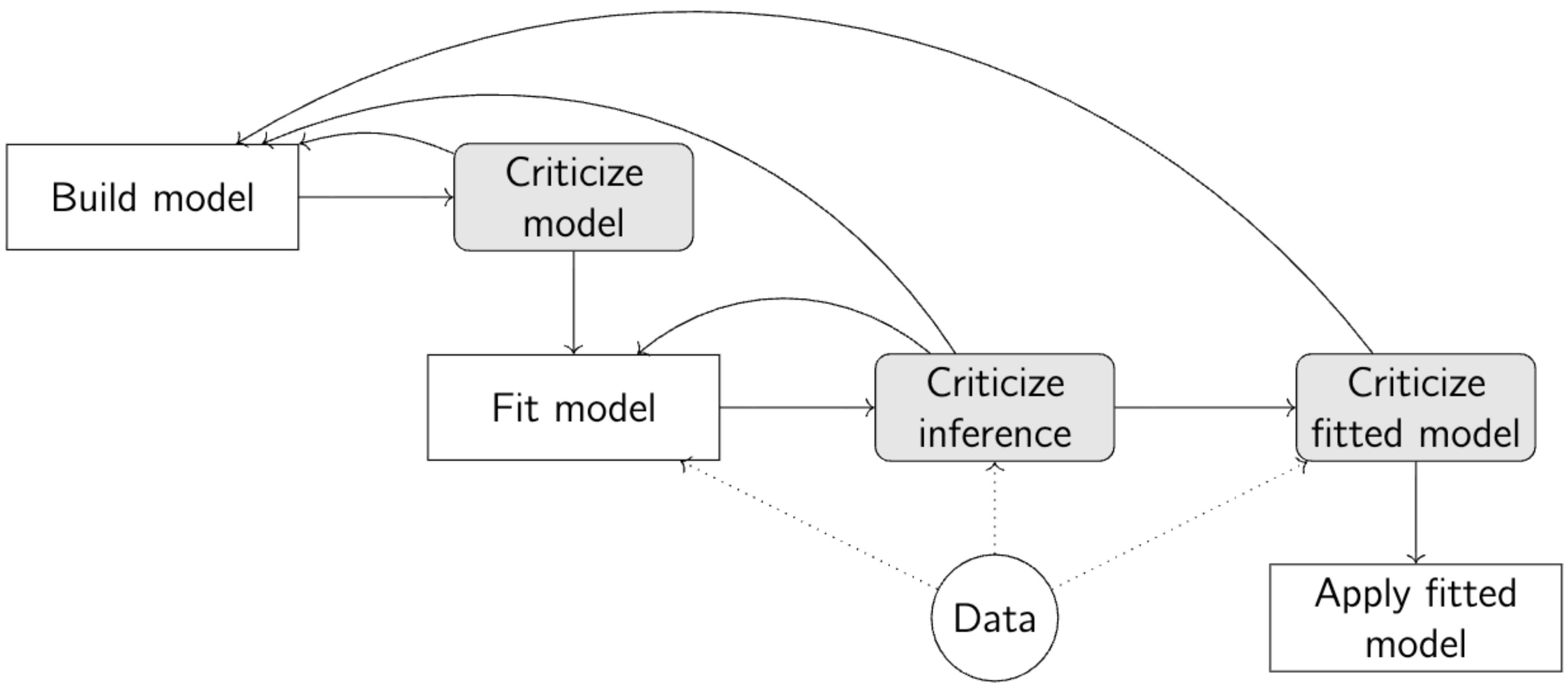
BAYESIAN WORKFLOW

- Bayesian modelling requires model building, fitting, applications, diagonistics, etc



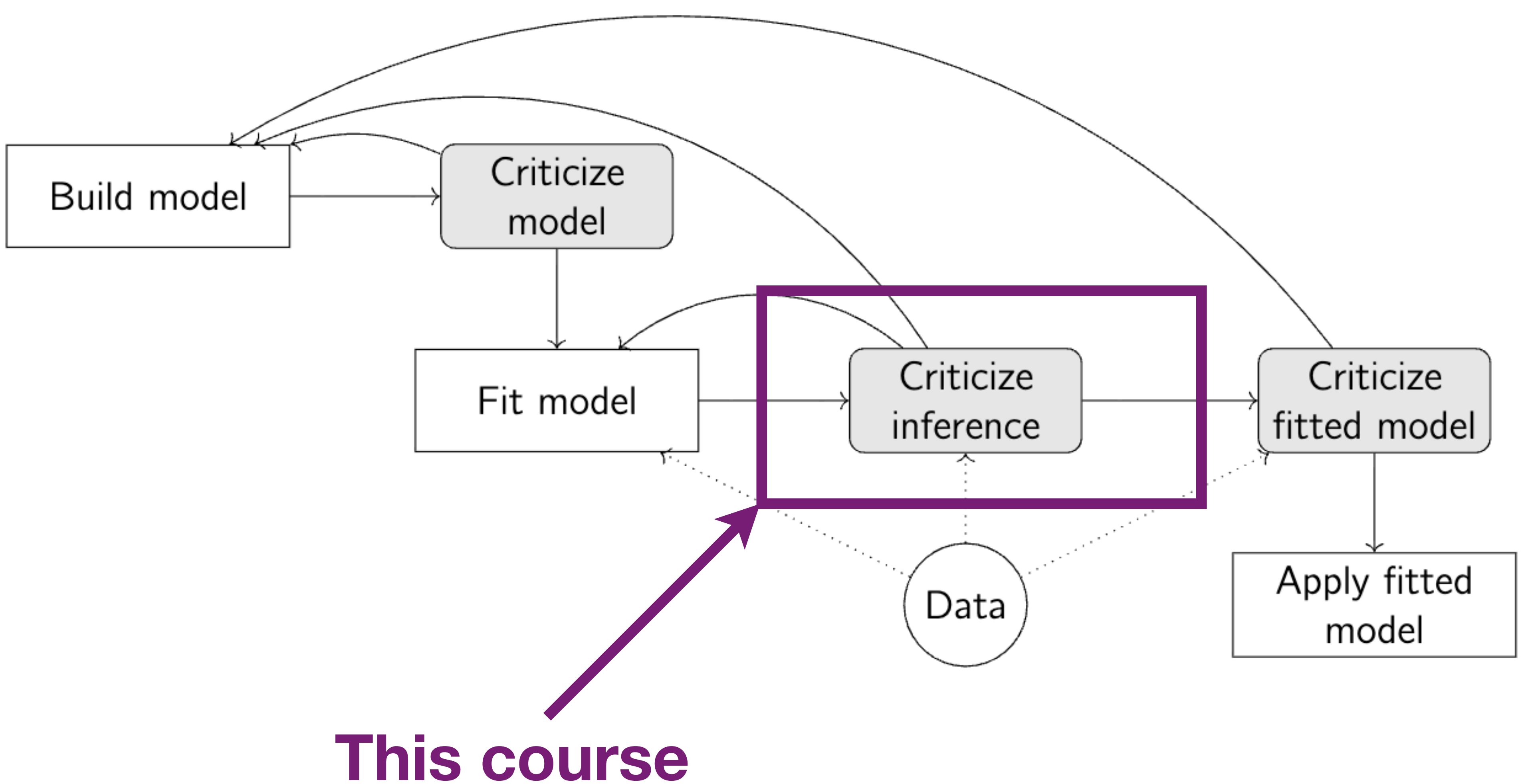
BAYESIAN WORKFLOW

- Bayesian modelling requires model building, fitting, applications, diagonistics, etc
  - It depends on the data, applications, model class, etc



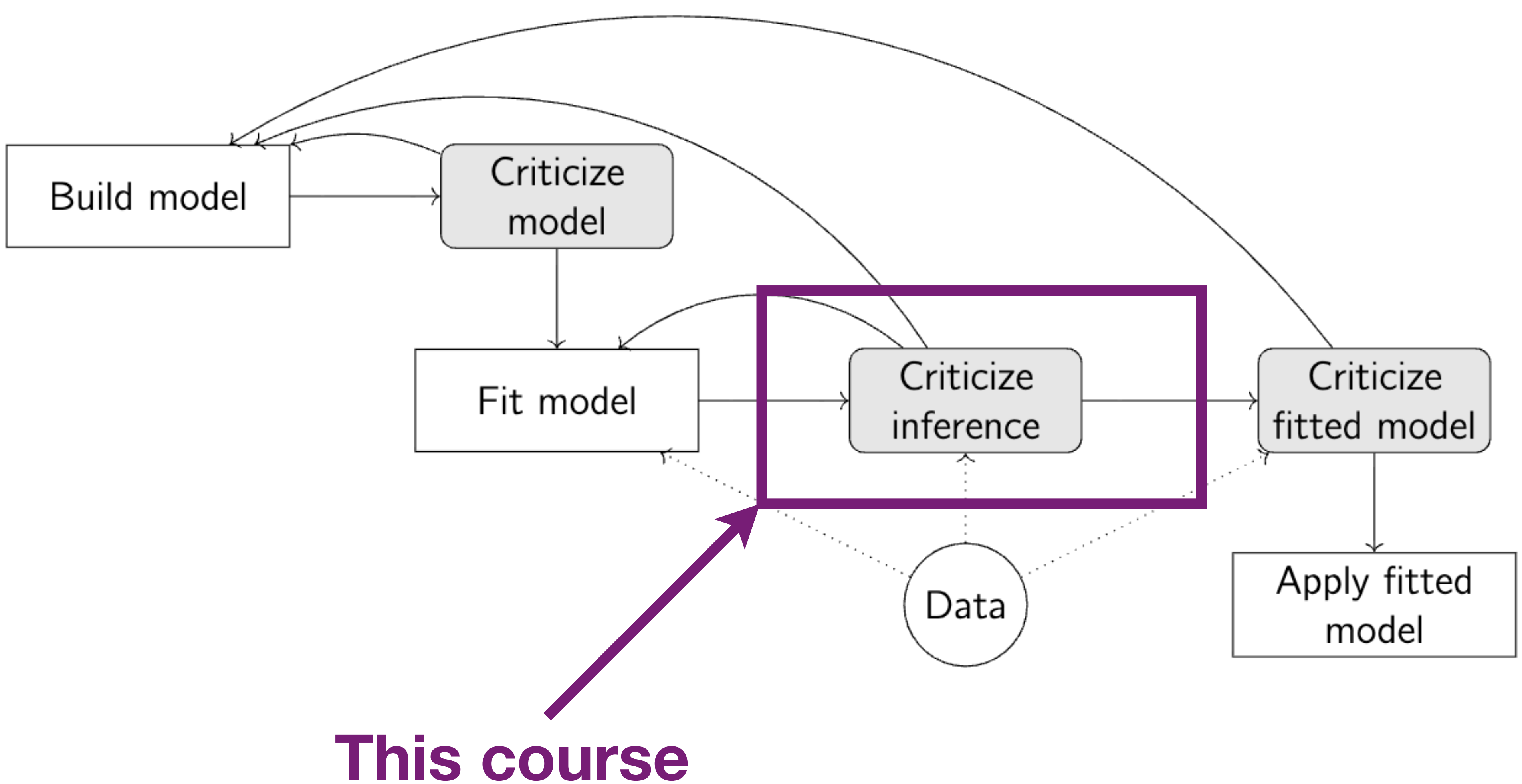
BAYESIAN WORKFLOW

- Bayesian modelling requires model building, fitting, applications, diagonistics, etc
  - It depends on the data, applications, model class, etc
- In this course, we only focus on **inference**.



BAYESIAN WORKFLOW

- Bayesian modelling requires model building, fitting, applications, diagonistics, etc
  - It depends on the data, applications, model class, etc
- In this course, we only focus on **inference**.
  - We don't judge where the distribtution comes from or its scienfic importance



BAYESIAN INFERENCE

BAYESIAN INFERENCE

- **Sample**: make inferences from the posterior (e.g. sample candidate images)

$$X \sim p(x \mid y)$$

BAYESIAN INFERENCE

- **Sample**: make inferences from the posterior (e.g. sample candidate images)

$$X \sim p(x \mid y)$$

- **Probabilities**: credible intervals (e.g. quantify uncertainty in the images)

$$p[X \in A \mid y] = \int_A p(y \mid x) \mathrm{d}x$$

BAYESIAN INFERENCE

- **Sample**: make inferences from the posterior (e.g. sample candidate images)

$$X \sim p(x \,|\, y)$$

- **Probabilities**: credible intervals (e.g. quantify uncertainty in the images)

$$p[X \in A \,|\, y] = \int_A p(y \,|\, x) \mathrm{d}x$$

- **Expectations**: expected statistics e.g. mass, polarisation, size, etc

$$p[f \,|\, y] = \int_{\mathbb{X}} f(x) p(y \,|\, x) \mathrm{d}x$$

BAYESIAN INFERENCE

- **Sample**: make inferences from the posterior (e.g. sample candidate images)

$$X \sim p(x \,|\, y)$$

- **Probabilities**: credible intervals (e.g. quantify uncertainty in the images)

$$p[X \in A \,|\, y] = \int_A p(y \,|\, x) \mathrm{d}x$$

- **Expectations**: expected statistics e.g. mass, polarisation, size, etc

$$p[f \,|\, y] = \int_{\mathbb{X}} f(x) p(y \,|\, x) \mathrm{d}x$$

- **Evidence**: e.g. for model comparison

$$p(y) = \int_{\mathbb{X}} L(y \,|\, x) p(x) \mathrm{d}x$$

SAMPLING: DENSITIES

SAMPLING: DENSITIES

- Suppose $\mathbb{X}$ is some statespace:

SAMPLING: DENSITIES

- Suppose $\mathbb{X}$ is some statespace:
  - Discrete, $\mathbb{R}^d$, a manifold, image, a graph, tree, a molecule, etc

SAMPLING: DENSITIES

- Suppose $\mathbb{X}$ is some statespace:
  - Discrete, $\mathbb{R}^d$, a manifold, image, a graph, tree, a molecule, etc

- Given a probability distribution π we want to sample over some stateapce $\mathbb{X}$

$$\pi(x) = \frac{\gamma(x)}{Z}$$

SAMPLING: DENSITIES

- Suppose $\mathbb{X}$ is some statespace:
  - Discrete, $\mathbb{R}^d$, a manifold, image, a graph, tree, a molecule, etc

- Given a probability distribution π we want to sample over some stateapce $\mathbb{X}$

$$\pi(x) = \frac{\gamma(x)}{Z}$$

  - Suppose we can evaluate $\gamma : \mathbb{X} \to \mathbb{R}$ efficiently

SAMPLING: DENSITIES

- Suppose $\mathbb{X}$ is some statespace:
  - Discrete, $\mathbb{R}^d$, a manifold, image, a graph, tree, a molecule, etc
- Given a probability distribution π we want to sample over some stateapce $\mathbb{X}$

$$\pi(x) = \frac{\gamma(x)}{Z}$$

  - Suppose we can evaluate $\gamma : \mathbb{X} \to \mathbb{R}$ efficiently
  - Cannot evaluate the normalising constant Z

$$Z = \int_{\mathbb{X}} \gamma(x)\mathrm{d}x$$

SAMPLING: DENSITIES

- Suppose $\mathbb{X}$ is some statespace:
  - Discrete, $\mathbb{R}^d$, a manifold, image, a graph, tree, a molecule, etc
- Given a probability distribution π we want to sample over some stateapce $\mathbb{X}$

$$\pi(x) = \frac{\gamma(x)}{Z}$$

  - Suppose we can evaluate $\gamma : \mathbb{X} \to \mathbb{R}$ efficiently
  - Cannot evaluate the normalising constant Z

$$Z = \int_{\mathbb{X}} \gamma(x) \mathrm{d}x$$

- We want to be able to say something about the properties of π

SAMPLING: POTENTIALS

SAMPLING: POTENTIALS

- Original formulation of sampling problem came from statistical mechanics

SAMPLING: POTENTIALS

- Original formulation of sampling problem came from statistical mechanics
- The target π is a Bolzmann-Gibbs distribution

$$\pi(x) = \frac{\exp(-\beta U(x))}{Z(\beta)}$$

SAMPLING: POTENTIALS

- Original formulation of sampling problem came from statistical mechanics
- The target π is a Bolzmann-Gibbs distribution

$$\pi(x) = \frac{\exp(-\beta U(x))}{Z(\beta)}$$

  - $U : \mathbb{X} \to \mathbb{R}$ is the potential energy, encoding the physics of a system

$$\gamma(x) = \exp(-\beta U(x))$$

SAMPLING: POTENTIALS

- Original formulation of sampling problem came from statistical mechanics
- The target π is a Bolzmann-Gibbs distribution

$$\pi(x) = \frac{\exp(-\beta U(x))}{Z(\beta)}$$

  - $U : \mathbb{X} \to \mathbb{R}$ is the potential energy, encoding the physics of a system

$$\gamma(x) = \exp(-\beta U(x))$$

  - β is the inverse-temperature of a system

SAMPLING: POTENTIALS

- Original formulation of sampling problem came from statistical mechanics
- The target π is a Bolzmann-Gibbs distribution

$$\pi(x) = \frac{\exp(-\beta U(x))}{Z(\beta)}$$

  - $U : \mathbb{X} \to \mathbb{R}$ is the potential energy, encoding the physics of a system

$$\gamma(x) = \exp(-\beta U(x))$$

  - β is the inverse-temperature of a system
  - Normalising constant is the partition function

$$Z(\beta) = \int_{\mathbb{X}} \exp(-\beta U(x))\mathrm{d}x$$

SAMPLING: POTENTIALS

- Original formulation of sampling problem came from statistical mechanics
- The target π is a Bolzmann-Gibbs distribution

$$\pi(x) = \frac{\exp(-\beta U(x))}{Z(\beta)}$$

  - $U : \mathbb{X} \to \mathbb{R}$ is the potential energy, encoding the physics of a system

$$\gamma(x) = \exp(-\beta U(x))$$

  - β is the inverse-temperature of a system
  - Normalising constant is the partition function

$$Z(\beta) = \int_{\mathbb{X}} \exp(-\beta U(x))\mathrm{d}x$$

- We will often interpolate between density and log-space

GOALS

GOALS

- **Sample**: make inferences from the target

$$X \sim \pi$$

GOALS

- **Sample**: make inferences from the target

$$X \sim \pi$$

- **Probabilities**: probability of a region $A \subset \mathbb{X}$

$$\pi[X \in A] = \int_A \pi(x)\mathrm{d}x$$

GOALS

- **Sample**: make inferences from the target

$$X \sim \pi$$

- **Probabilities**: probability of a region $A \subset \mathbb{X}$

$$\pi[X \in A] = \int_A \pi(x)\mathrm{d}x$$

- **Expectations**: given $f : \mathbb{X} \to \mathbb{R}$

$$\pi[f] = \int_{\mathbb{X}} f(x)\pi(x)\mathrm{d}x$$

GOALS

- **Sample**: make inferences from the target

$$X \sim \pi$$

- **Probabilities**: probability of a region $A \subset \mathbb{X}$

$$\pi[X \in A] = \int_A \pi(x)\mathrm{d}x$$

- **Expectations**: given $f : \mathbb{X} \to \mathbb{R}$

$$\pi[f] = \int_{\mathbb{X}} f(x)\pi(x)\mathrm{d}x$$

- **Normalising constant:**

$$Z = \int_{\mathbb{X}} \gamma(x)\mathrm{d}x$$

EXAMPLE: BAYESIAN STATISTICS

EXAMPLE: BAYESIAN STATISTICS

- Given a prior $p(x)$ and likelihood $L(y \mid x)$

EXAMPLE: BAYESIAN STATISTICS

- Given a prior $p(x)$ and likelihood $L(y | x)$
- In bayesian statististics π is the posterior

$$\pi(x) = \frac{L(y | x)p(x)}{p(y)} = \frac{\gamma(x)}{Z}$$

EXAMPLE: BAYESIAN STATISTICS

- Given a prior $p(x)$ and likelihood $L(y \mid x)$
- In bayesian statististics π is the posterior

$$\pi(x) = \frac{L(y \mid x)p(x)}{p(y)} = \frac{\gamma(x)}{Z}$$

  - $\gamma(x)$ is the un-normalised posterior density

$$\gamma(x) = L(y \mid x)p(x)$$

EXAMPLE: BAYESIAN STATISTICS

- Given a prior $p(x)$ and likelihood $L(y \mid x)$
- In bayesian statistics π is the posterior

$$\pi(x) = \frac{L(y \mid x)p(x)}{p(y)} = \frac{\gamma(x)}{Z}$$

  - $\gamma(x)$ is the un-normalised posterior density

$$\gamma(x) = L(y \mid x)p(x)$$

  - Normalising constant is the model evidence

$$Z = p(y) = \int_{\mathbb{X}} L(y \mid x)p(x)\mathrm{d}x$$

EXAMPLE: BAYESIAN STATISTICS

- Given a prior $p(x)$ and likelihood $L(y \mid x)$
- In bayesian statististics π is the posterior

$$\pi(x) = \frac{L(y \mid x)p(x)}{p(y)} = \frac{\gamma(x)}{Z}$$

  - $\gamma(x)$ is the un-normalised posterior density

$$\gamma(x) = L(y \mid x)p(x)$$

  - Normalising constant is the model evidence

$$Z = p(y) = \int_{\mathbb{X}} L(y \mid x)p(x)\mathrm{d}x$$

- For us, the data is an abstraction

EXAMPLE: MOLECULAR DYNAMICS

EXAMPLE: MOLECULAR DYNAMICS

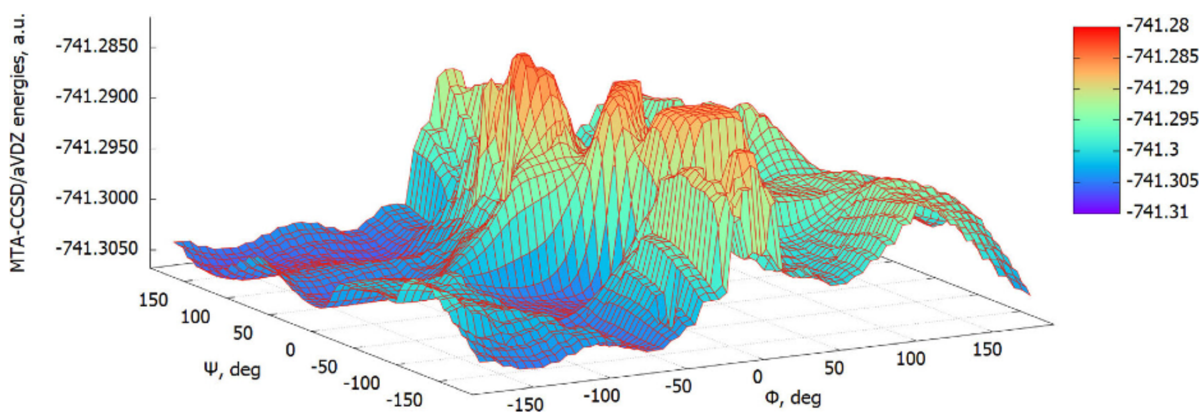
- Statespace represents different configurations of a system of particles

EXAMPLE: MOLECULAR DYNAMICS

- Statespace represents different configurations of a system of particles
- Potential encodes the molecular forces between atoms, molecules, protiens etc

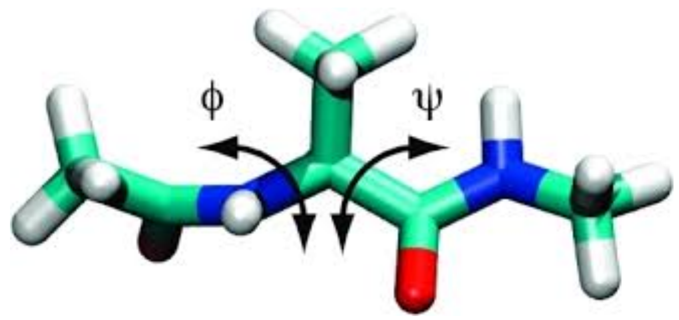
EXAMPLE: MOLECULAR DYNAMICS

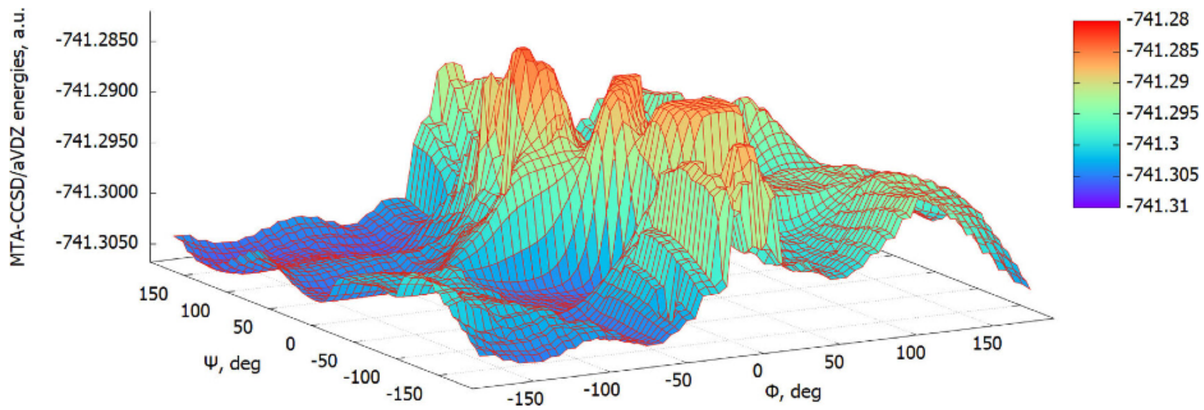
- Statespace represents different configurations of a system of particles
- Potential encodes the molecular forces between atoms, molecules, protiens etc



EXAMPLE: MOLECULAR DYNAMICS

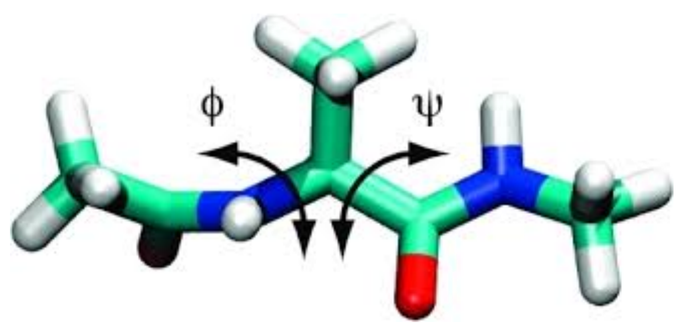
- Statespace represents different configurations of a system of particles
- Potential encodes the molecular forces between atoms, molecules, protiens etc

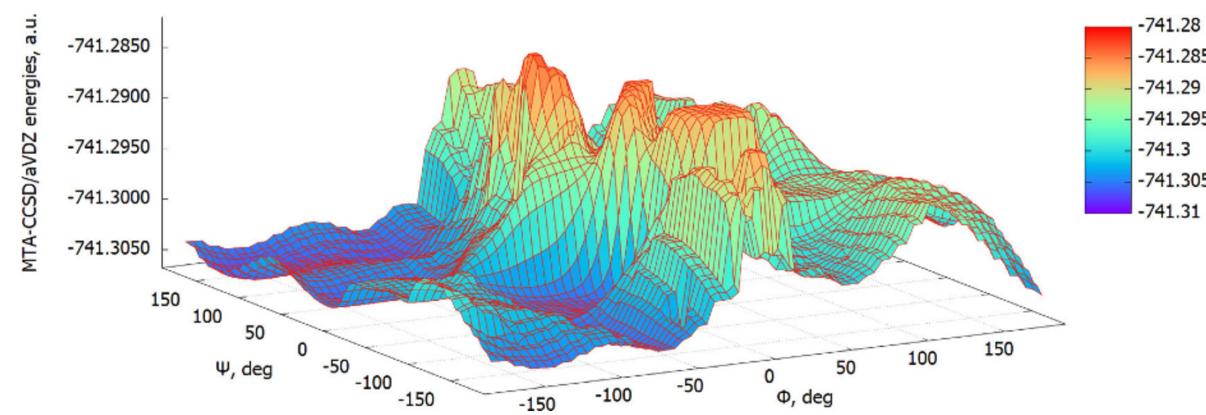




EXAMPLE: MOLECULAR DYNAMICS

- Statespace represents different configurations of a system of particles
- Potential encodes the molecular forces between atoms, molecules, protiens etc

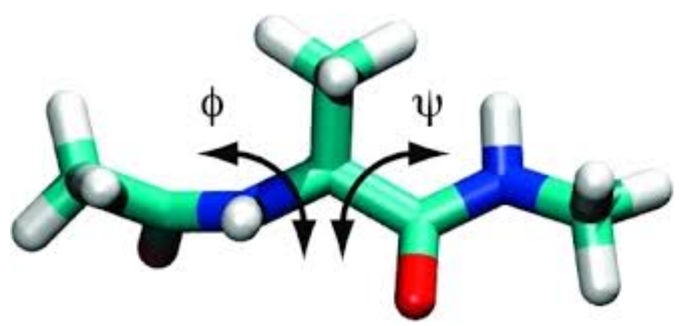


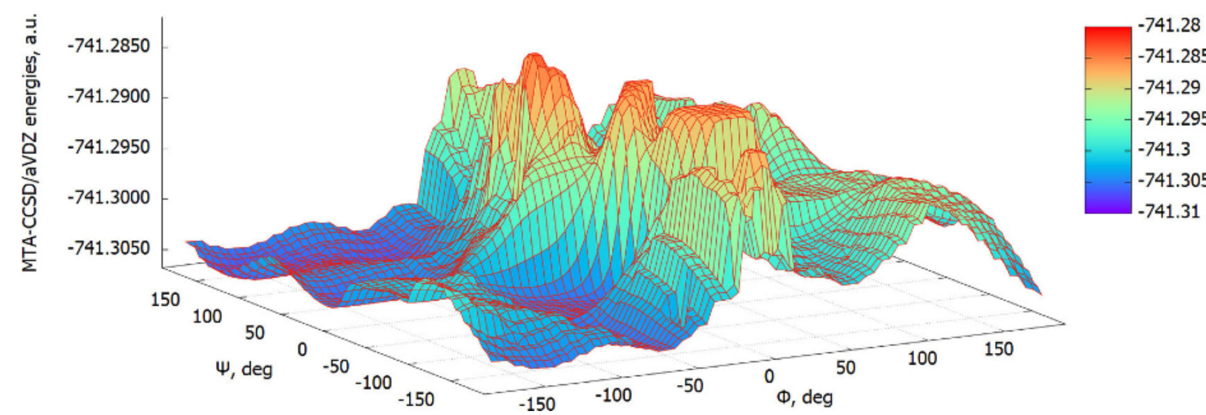


- Molecular dynamics simulates these systems to infer physical properties

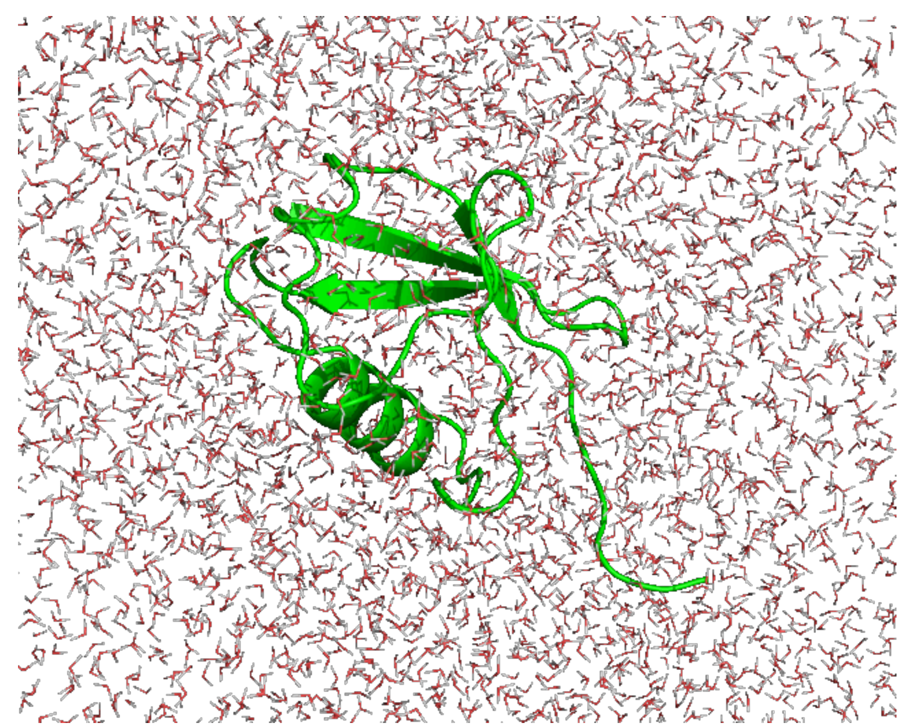
EXAMPLE: MOLECULAR DYNAMICS

- Statespace represents different configurations of a system of particles
- Potential encodes the molecular forces between atoms, molecules, protiens etc



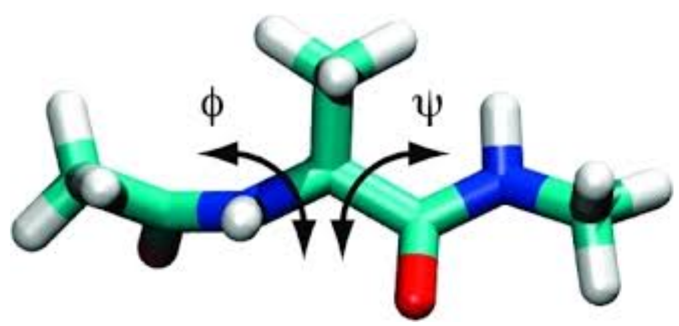


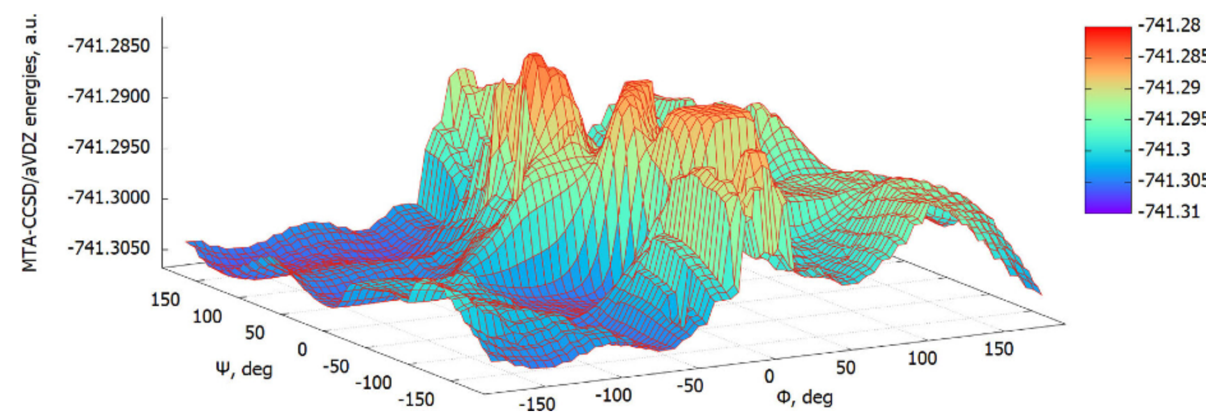
- Molecular dynamics simulates these systems to infer physical properties



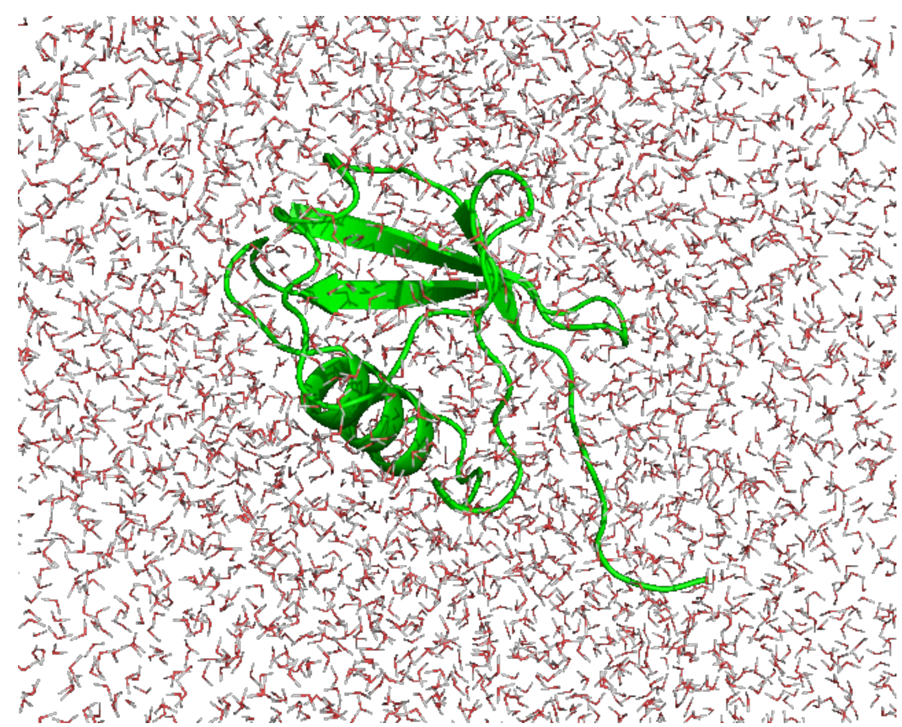
EXAMPLE: MOLECULAR DYNAMICS

- Statespace represents different configurations of a system of particles
- Potential encodes the molecular forces between atoms, molecules, protiens etc



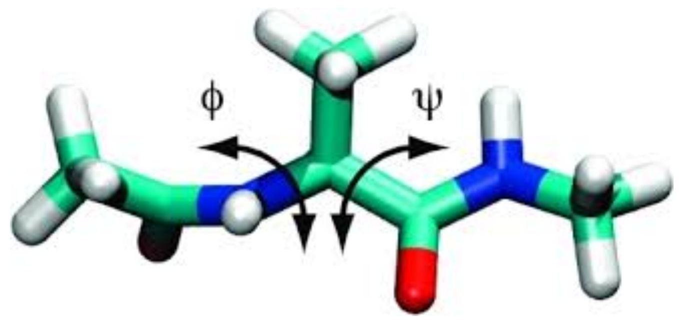


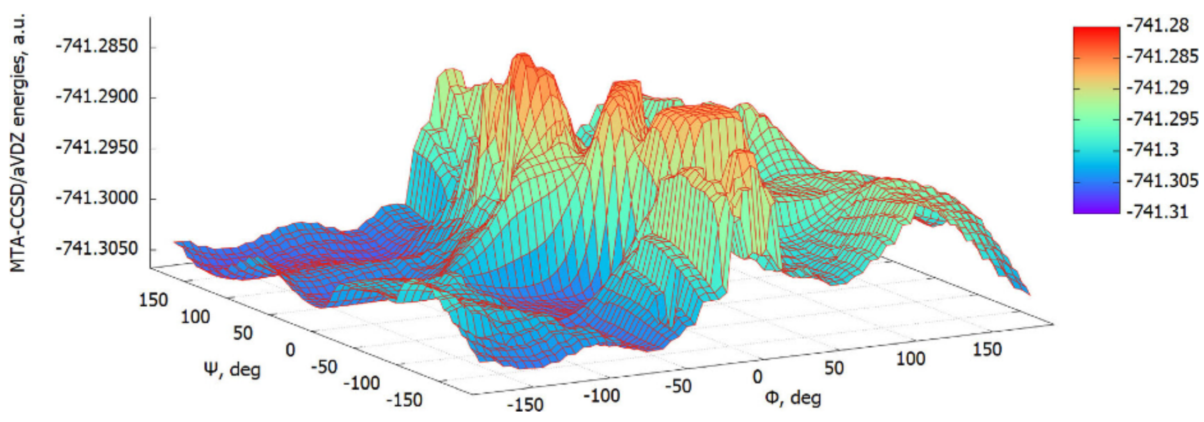
- Molecular dynamics simulates these systems to infer physical properties
- Estimate reactions rates, binding affinity, material properties, etc



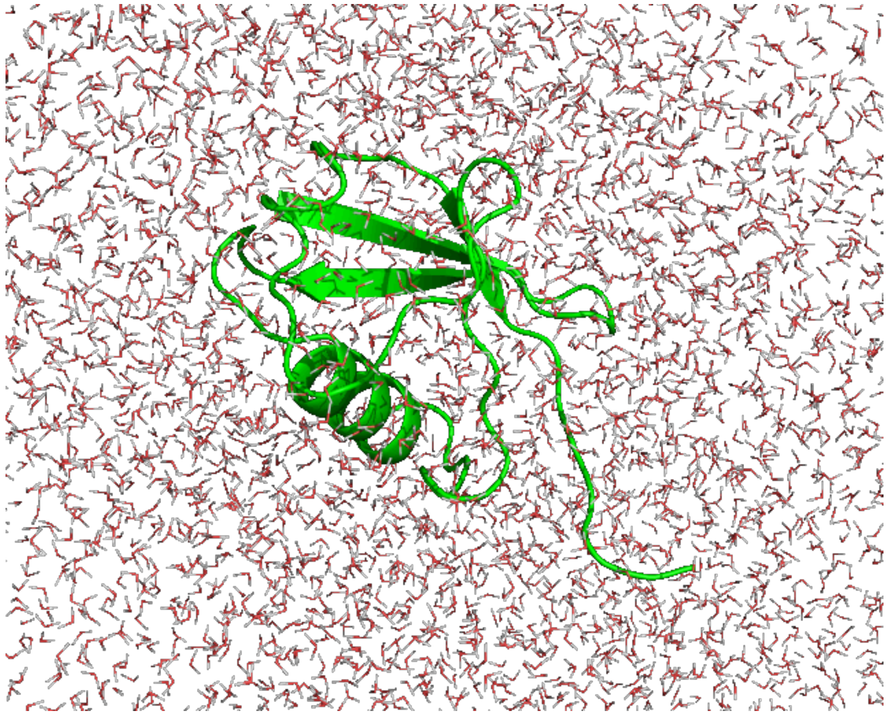
EXAMPLE: MOLECULAR DYNAMICS

- Statespace represents different configurations of a system of particles
- Potential encodes the molecular forces between atoms, molecules, protiens etc





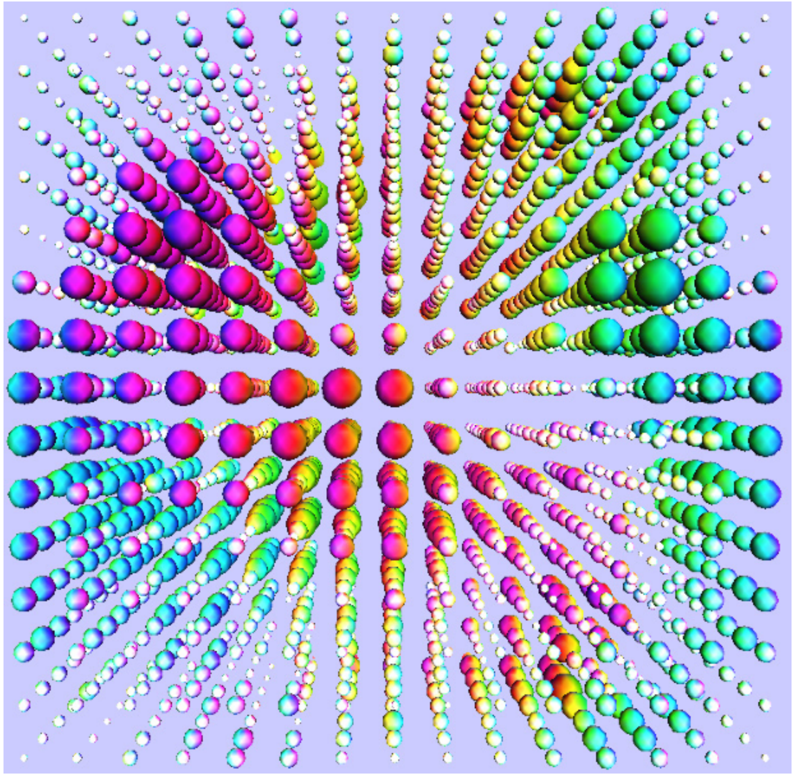
- Molecular dynamics simulates these systems to infer physical properties
- Estimate reactions rates, binding affinity, material properties, etc
- Very important for drug discovery



EXAMPLE: QUANTUM CHROMODYNAMICS

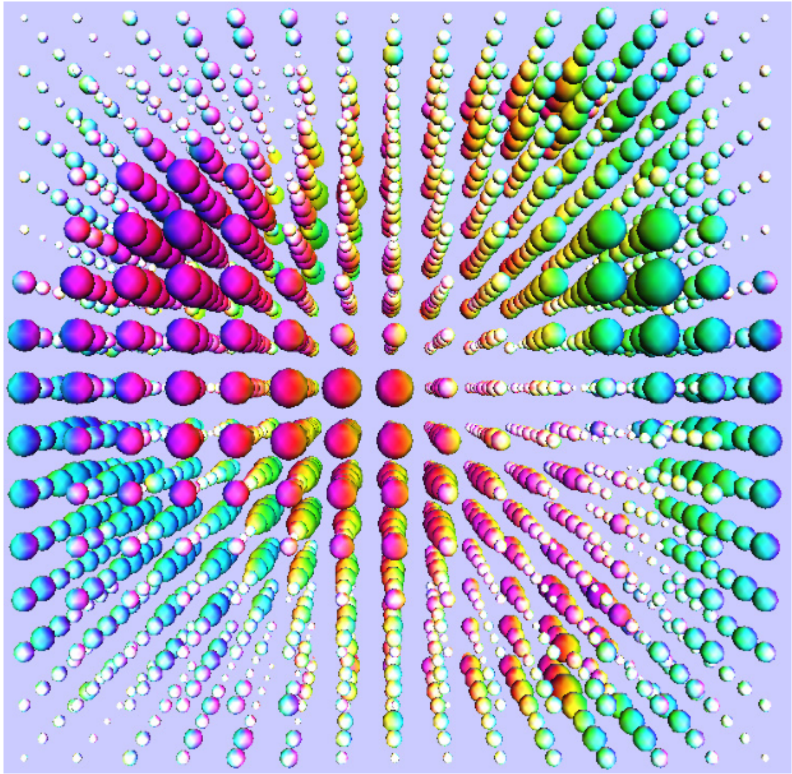
EXAMPLE: QUANTUM CHROMODYNAMICS

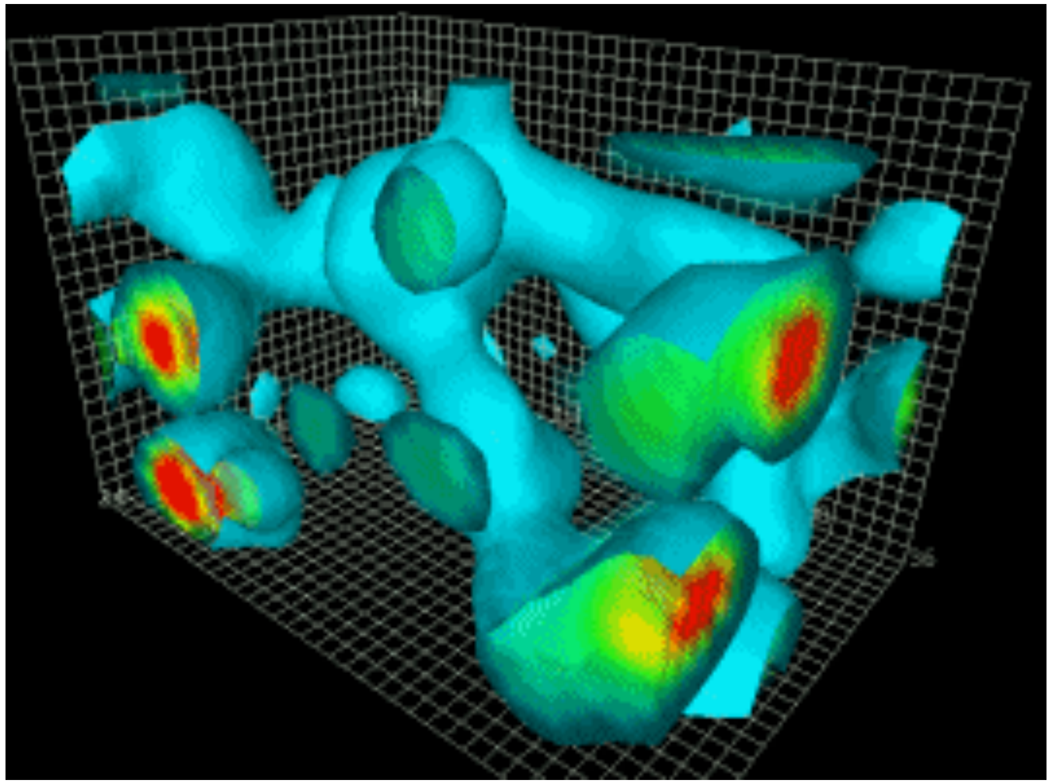
- Statespace is a lattice in 4-dimensions with each site is a Lie group (i.e. gauge field)



EXAMPLE: QUANTUM CHROMODYNAMICS

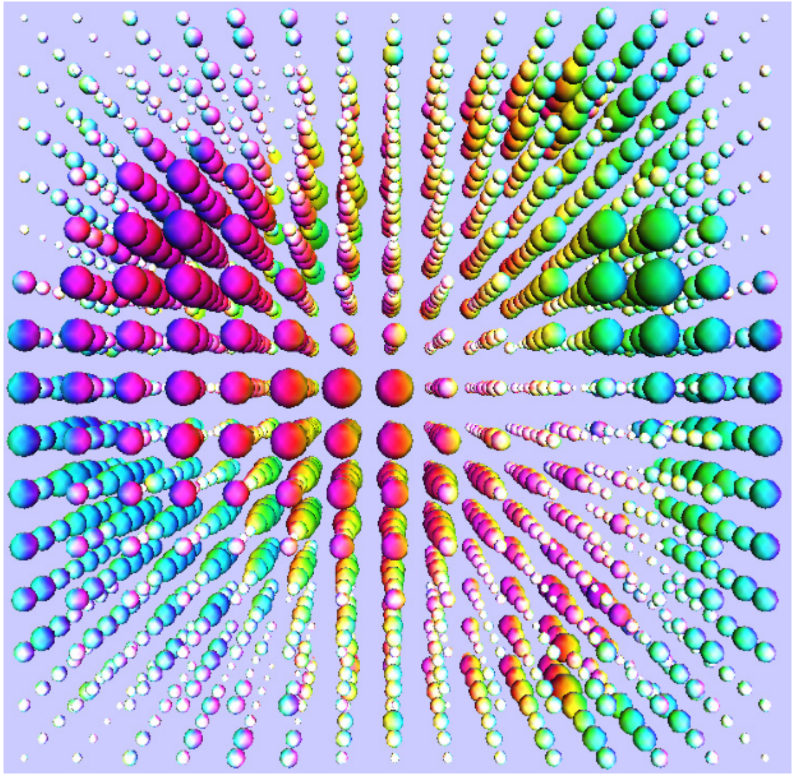
- Statespace is a lattice in 4-dimensions with each site is a Lie group (i.e. gauge field)
- Target density encodes the dynamics governing fluctuations of sub-atomic particles

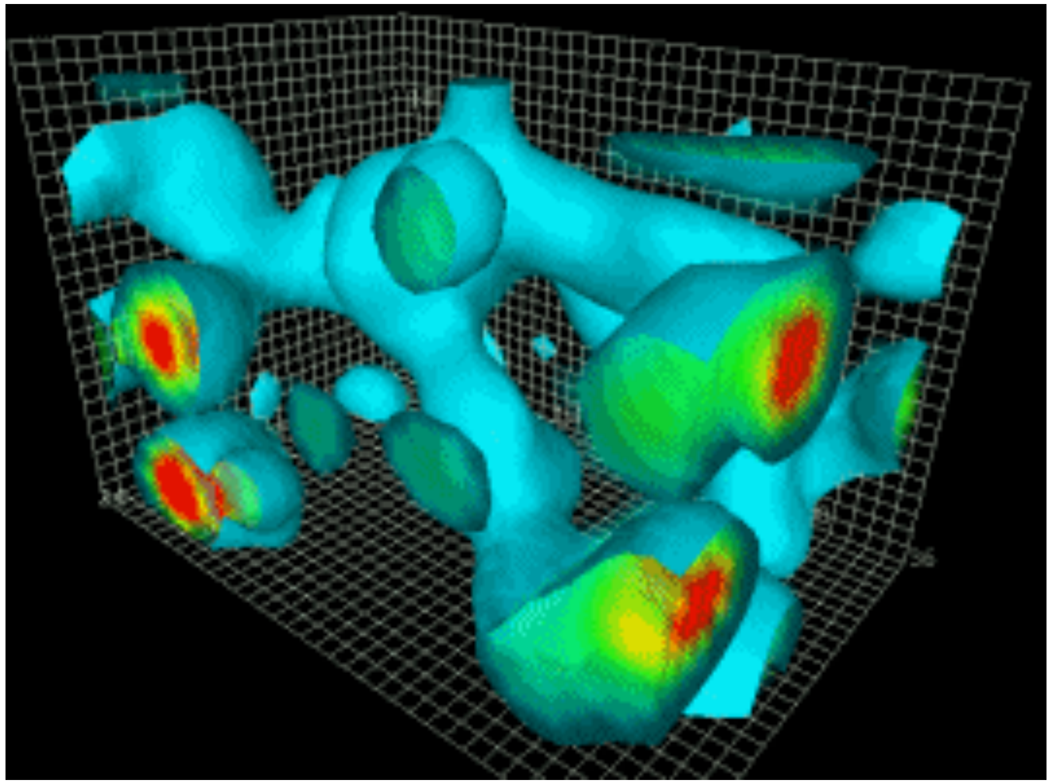




EXAMPLE: QUANTUM CHROMODYNAMICS

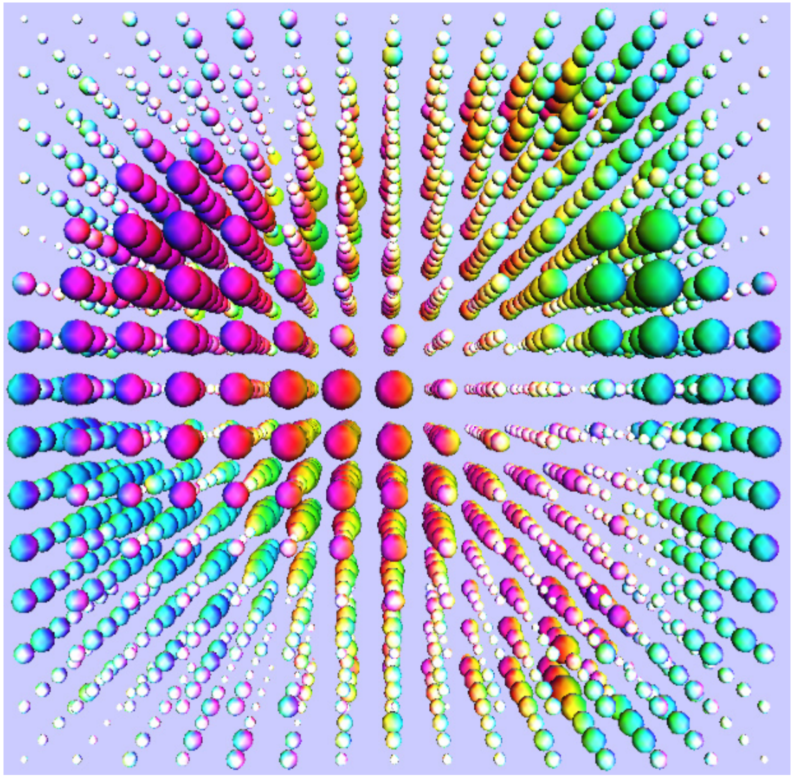
- Statespace is a lattice in 4-dimensions with each site is a Lie group (i.e. gauge field)
- Target density encodes the dynamics governing fluctuations of sub-atomic particles
- Used to model and understand strong nuclear forces/validate standard model

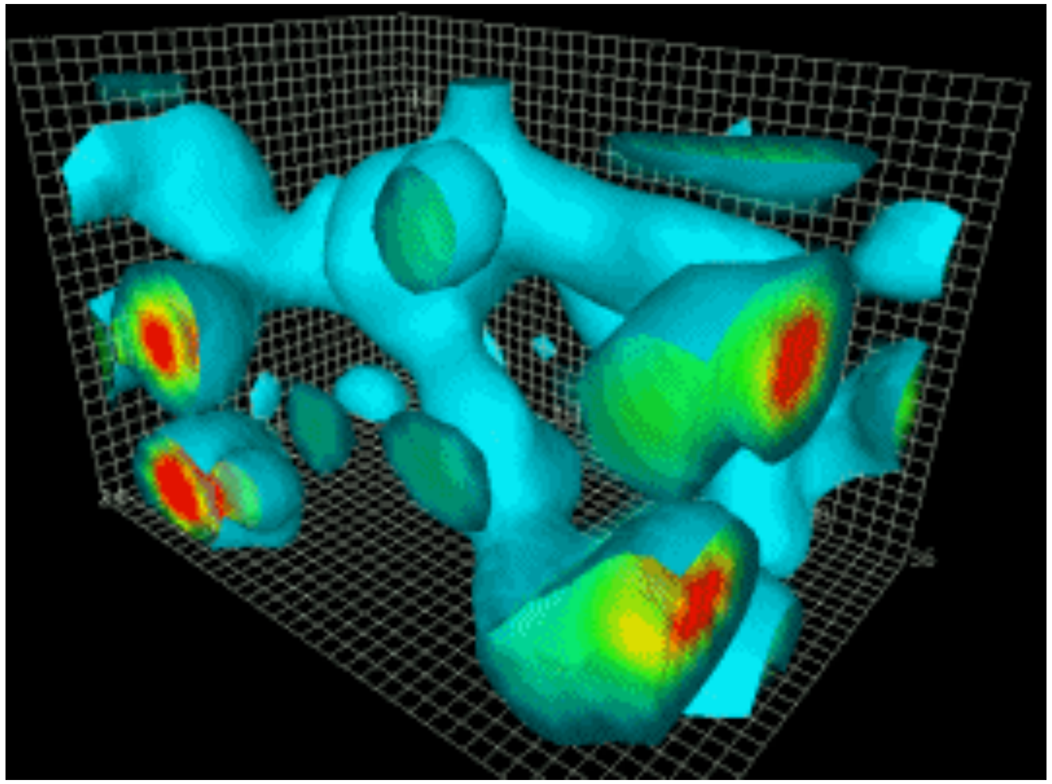




EXAMPLE: QUANTUM CHROMODYNAMICS

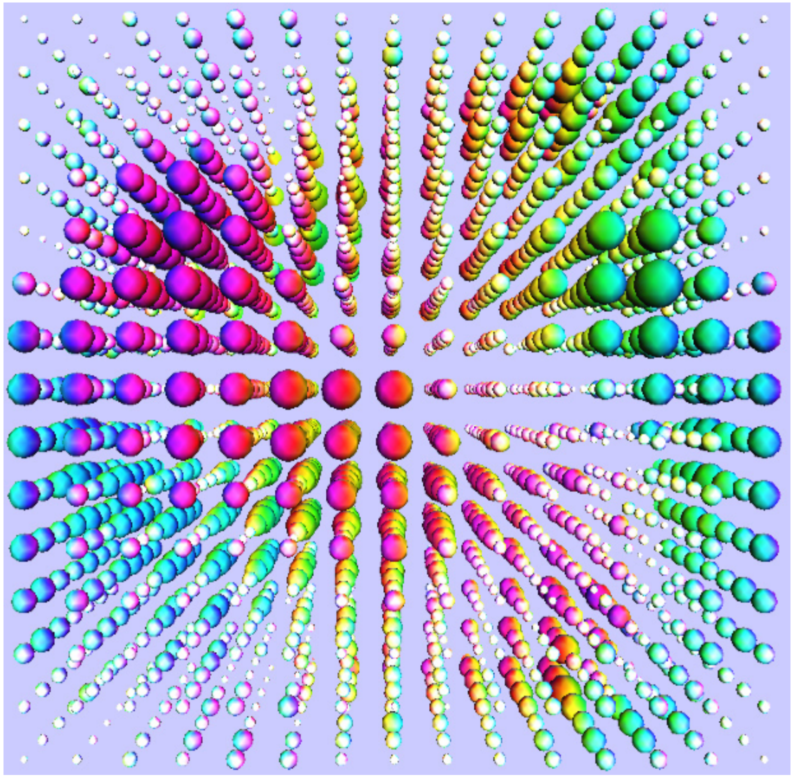
- Statespace is a lattice in 4-dimensions with each site is a Lie group (i.e. gauge field)
- Target density encodes the dynamics governing fluctuations of sub-atomic particles
- Used to model and understand strong nuclear forces/validate standard model
  - e.g calculate mass and spin, charge, of a proton, gluons, etc

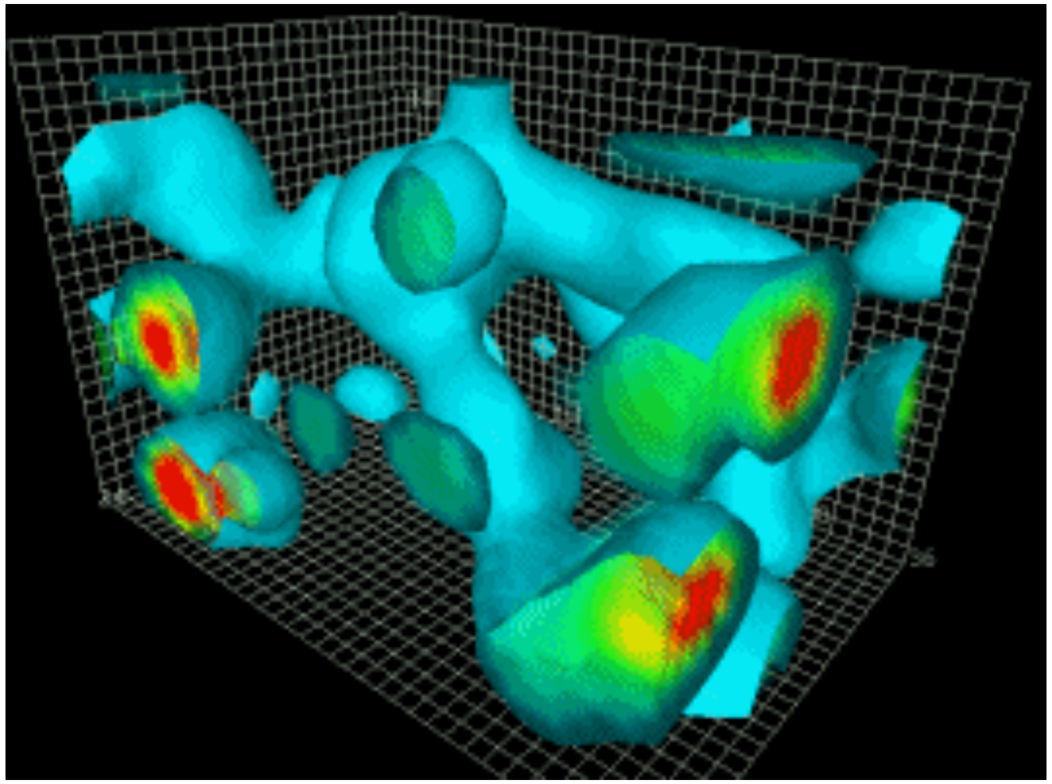




EXAMPLE: QUANTUM CHROMODYNAMICS

- Statespace is a lattice in 4-dimensions with each site is a Lie group (i.e. gauge field)
- Target density encodes the dynamics governing fluctuations of sub-atomic particles
- Used to model and understand strong nuclear forces/validate standard model
  - e.g calculate mass and spin, charge, of a proton, gluons, etc
- Run simulations for months at a time, use ~15% of global compute resources





SAMPLES VS EXPECTATIONS

SAMPLES VS EXPECTATIONS

- Given a sample $X \sim \pi$, we can code features $f(X)$ of X using a statistic $f : \mathbb{X} \to \mathbb{R}$

SAMPLES VS EXPECTATIONS

- Given a sample $X \sim \pi$, we can code features $f(X)$ of X using a statistic $f : \mathbb{X} \to \mathbb{R}$
- Can estimate the average feature using expectations.

$$\pi[f] = \mathbb{E}_{X \sim \pi}[f(X)]$$

SAMPLES VS EXPECTATIONS

- Given a sample $X \sim \pi$, we can code features $f(X)$ of X using a statistic $f : \mathbb{X} \to \mathbb{R}$
- Can estimate the average feature using expectations.

$$\pi[f] = \mathbb{E}_{X \sim \pi}[f(X)]$$

- Expectations, probabilities fully characterise distributions:

SAMPLES VS EXPECTATIONS

- Given a sample $X \sim \pi$, we can code features $f(X)$ of X using a statistic $f : \mathbb{X} \to \mathbb{R}$
- Can estimate the average feature using expectations.

$$\pi[f] = \mathbb{E}_{X \sim \pi}[f(X)]$$

- Expectations, probabilities fully characterise distributions:
  - If μ is some distribution such that $\mu[A] = \pi[A]$ for all $A \subset \mathbb{X}$ then $\mu = \pi$

SAMPLES VS EXPECTATIONS

- Given a sample $X \sim \pi$, we can code features $f(X)$ of X using a statistic $f : \mathbb{X} \to \mathbb{R}$
- Can estimate the average feature using expectations.

$$\pi[f] = \mathbb{E}_{X\sim\pi}[f(X)]$$

- Expectations, probabilities fully characterise distributions:
  - If μ is some distribution such that $\mu[A] = \pi[A]$ for all $A \subset \mathbb{X}$ then $\mu = \pi$
  - If μ is some distribution such that $\mu[f] = \pi[f]$ for all bounded f then $\mu = \pi$

SAMPLES VS EXPECTATIONS

- Given a sample $X \sim \pi$, we can code features $f(X)$ of X using a statistic $f : \mathbb{X} \to \mathbb{R}$
- Can estimate the average feature using expectations.

$$\pi[f] = \mathbb{E}_{X\sim\pi}[f(X)]$$

- Expectations, probabilities fully characterise distributions:
  - If μ is some distribution such that $\mu[A] = \pi[A]$ for all $A \subset \mathbb{X}$ then $\mu = \pi$
  - If μ is some distribution such that $\mu[f] = \pi[f]$ for all bounded f then $\mu = \pi$
- We will say that we can efficiently sample from X if we can provide we can efficiently compute $\pi[f]$ for any f

SAMPLES VS EXPECTATIONS

- Given a sample $X \sim \pi$, we can code features $f(X)$ of X using a statistic $f : \mathbb{X} \to \mathbb{R}$
- Can estimate the average feature using expectations.

$$\pi[f] = \mathbb{E}_{X \sim \pi}[f(X)]$$

- Expectations, probabilities fully characterise distributions:
  - If μ is some distribution such that $\mu[A] = \pi[A]$ for all $A \subset \mathbb{X}$ then $\mu = \pi$
  - If μ is some distribution such that $\mu[f] = \pi[f]$ for all bounded f then $\mu = \pi$
- We will say that we can efficiently sample from X if we can provide we can efficiently compute $\pi[f]$ for any f
- This is basically a course on numerically computing integrals

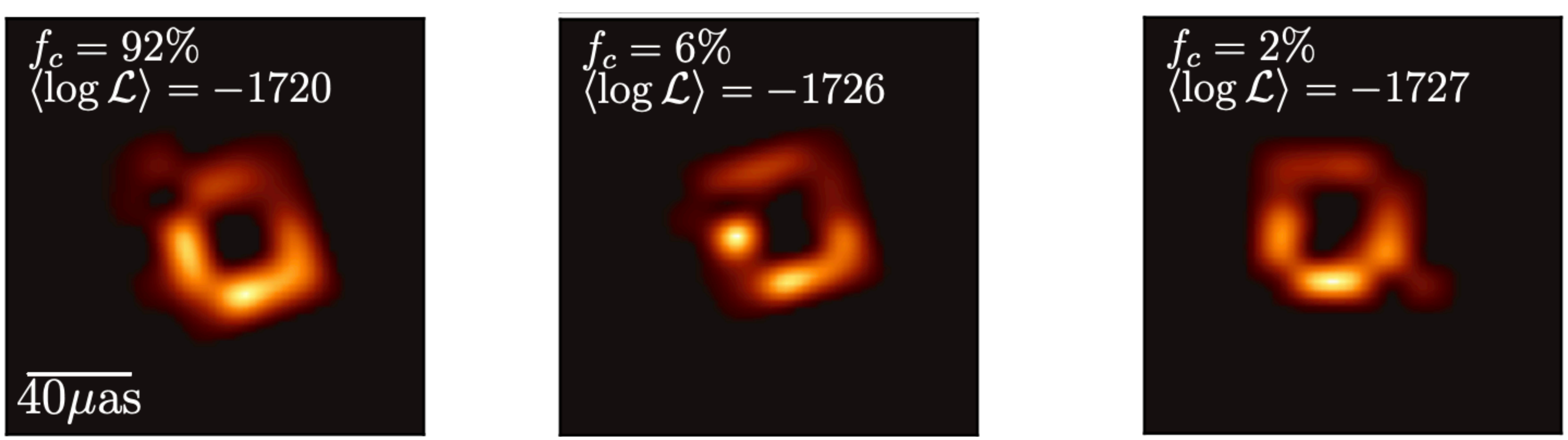
MONTE CARLO METHODS:

MONTE CARLO METHODS:

- **Problem:** computing integrals is hard

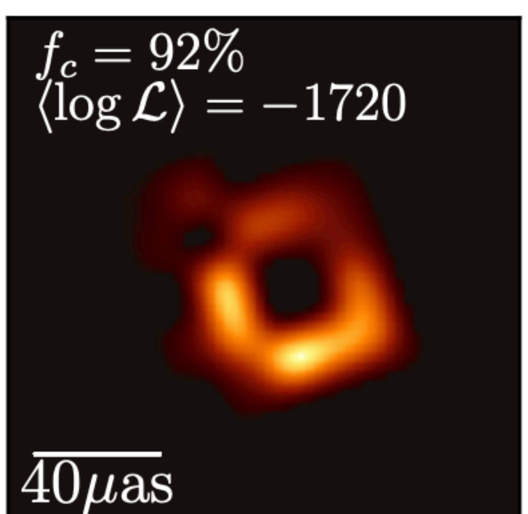
MONTE CARLO METHODS:

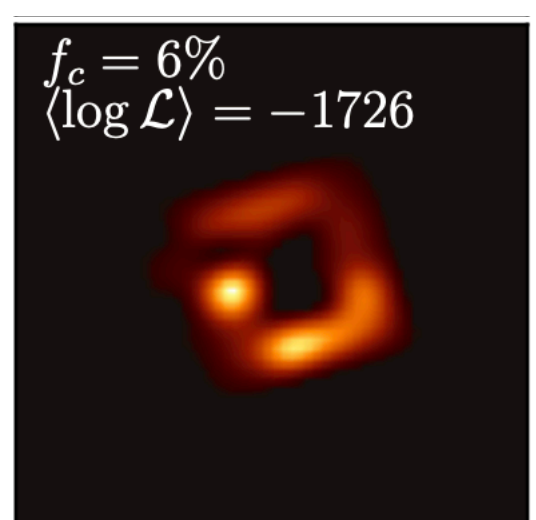
- **Problem:** computing integrals is hard
  - Multi-modality, varying scales, unidentifiable, combinatorial explotions

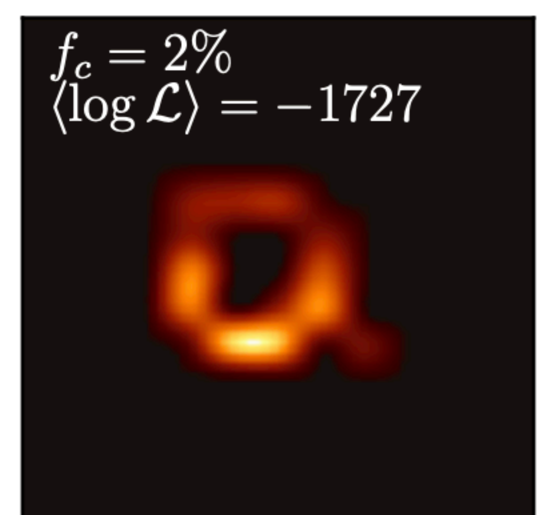


MONTE CARLO METHODS:

- **Problem:** computing integrals is hard
  - Multi-modality, varying scales, unidentifiable, combinatorial explotions
  - Problem only gets worse in high-dimensions

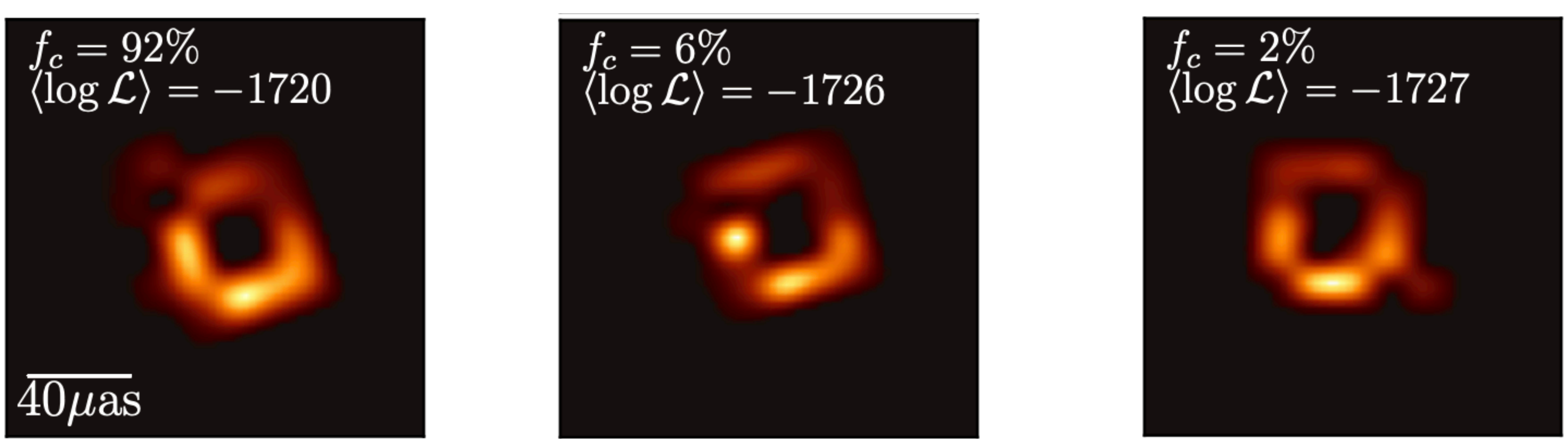






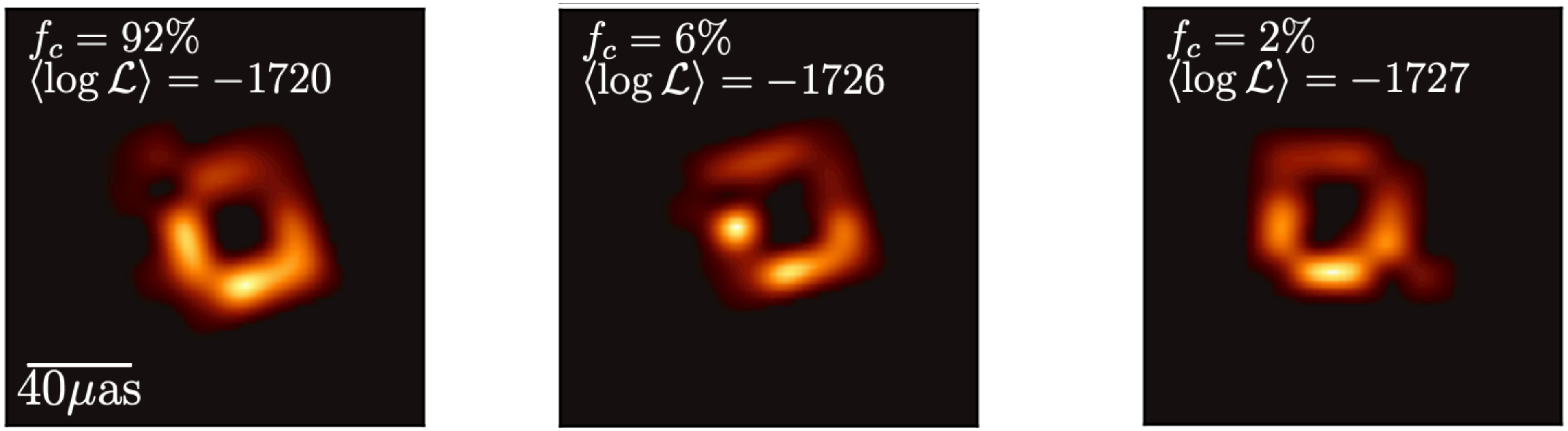
MONTE CARLO METHODS:

- **Problem:** computing integrals is hard
  - Multi-modality, varying scales, unidentifiable, combinatorial explotions
  - Problem only gets worse in high-dimensions
- **Monte Carlo (MC) algorithms:** approximate inference



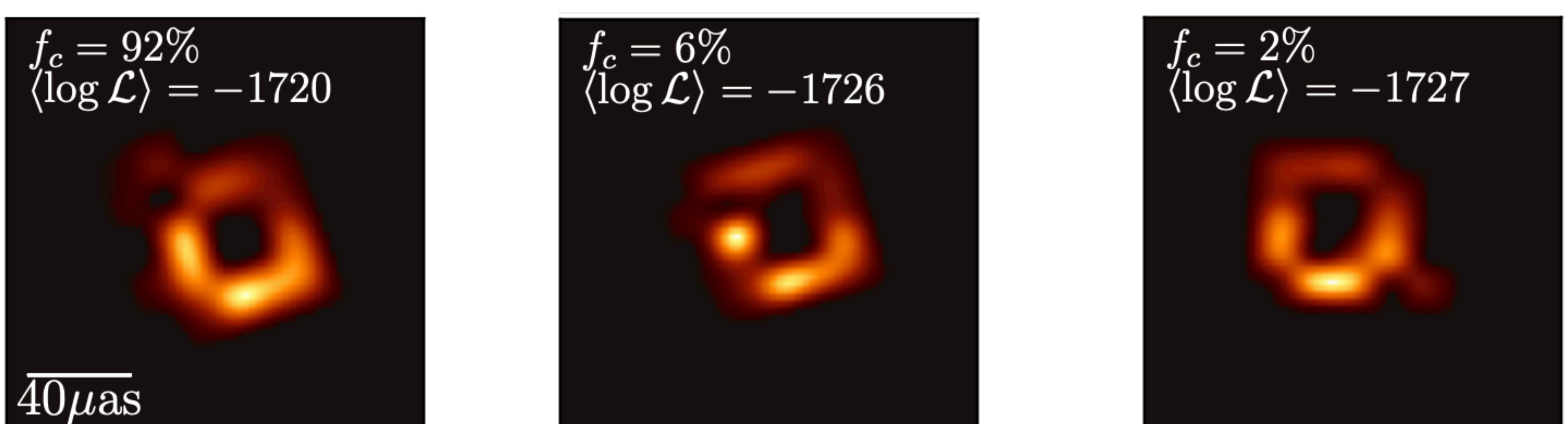
MONTE CARLO METHODS:

- **Problem:** computing integrals is hard
    - Multi-modality, varying scales, unidentifiable, combinatorial explotions
    - Problem only gets worse in high-dimensions
- **Monte Carlo (MC) algorithms:** approximate inference
    - A class of algorithms that gamble accuracy for a fixed compute budget.



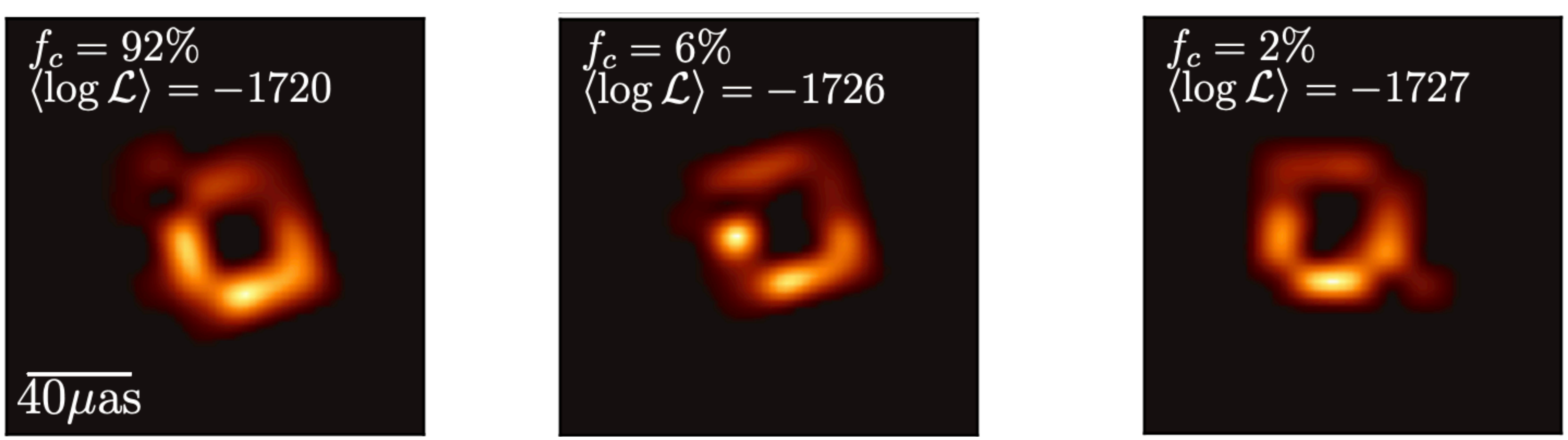
MONTE CARLO METHODS:

- **Problem:** computing integrals is hard
  - Multi-modality, varying scales, unidentifiable, combinatorial explotions
  - Problem only gets worse in high-dimensions
- **Monte Carlo (MC) algorithms:** approximate inference
  - A class of algorithms that gamble accuracy for a fixed compute budget.
  - Accuracy increases with budget



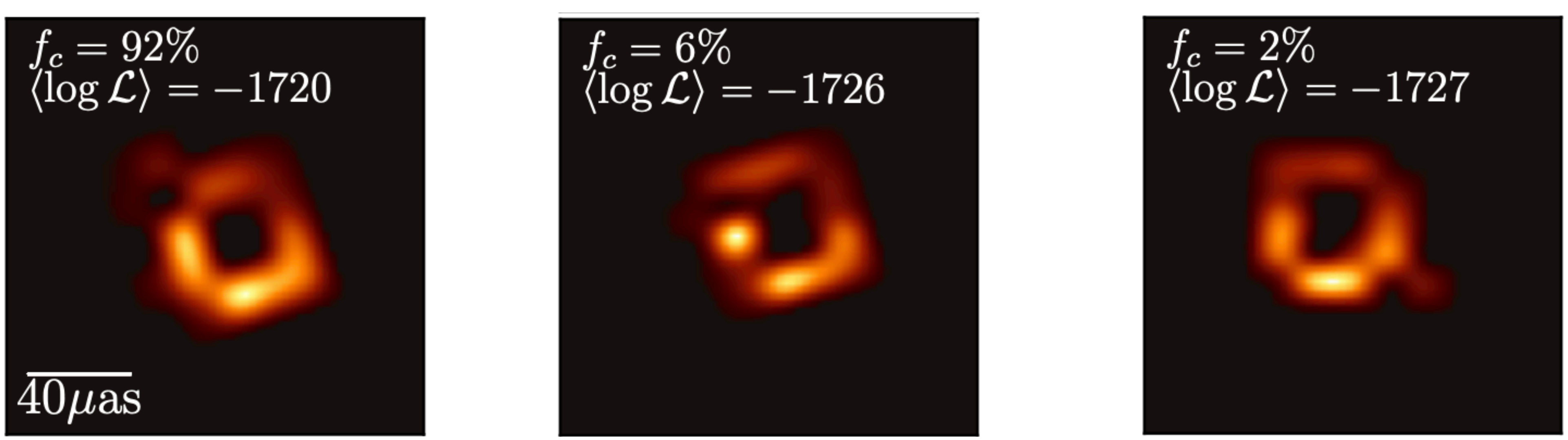
MONTE CARLO METHODS:

- **Problem:** computing integrals is hard
  - Multi-modality, varying scales, unidentifiable, combinatorial explotions
  - Problem only gets worse in high-dimensions
- **Monte Carlo (MC) algorithms:** approximate inference
  - A class of algorithms that gamble accuracy for a fixed compute budget.
  - Accuracy increases with budget
- **Las Vegas algorithms:** perfect inference



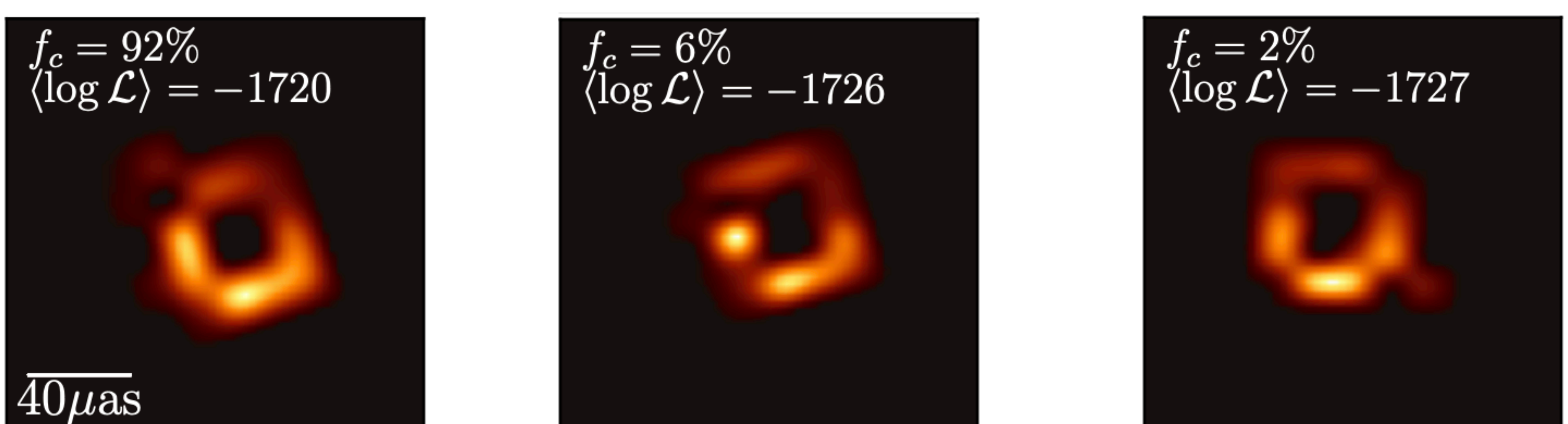
MONTE CARLO METHODS:

- **Problem:** computing integrals is hard
  - Multi-modality, varying scales, unidentifiable, combinatorial explotions
  - Problem only gets worse in high-dimensions
- **Monte Carlo (MC) algorithms:** approximate inference
  - A class of algorithms that gamble accuracy for a fixed compute budget.
  - Accuracy increases with budget
- **Las Vegas algorithms:** perfect inference
  - A class of algorithms that gamble computational budget for perfect accuracy



MONTE CARLO METHODS:

- **Problem:** computing integrals is hard
  - Multi-modality, varying scales, unidentifiable, combinatorial explotions
  - Problem only gets worse in high-dimensions
- **Monte Carlo (MC) algorithms:** approximate inference
  - A class of algorithms that gamble accuracy for a fixed compute budget.
  - Accuracy increases with budget
- **Las Vegas algorithms:** perfect inference
  - A class of algorithms that gamble computational budget for perfect accuracy
  - Compute resources increases with problem complexity



MONTE CARLO ESTIMATOR

MONTE CARLO ESTIMATOR

- Given iid samples from $X_1, \ldots, X_N \sim \pi$, define the Monte Carlo (MC) estimator:

$$\hat{\pi}[f] = \frac{1}{N} \sum_{n=1}^{N} f(X_n)$$

MONTE CARLO ESTIMATOR

- Given iid samples from $X_1, \ldots, X_N \sim \pi$, define the Monte Carlo (MC) estimator:

$$\hat{\pi}[f] = \frac{1}{N} \sum_{n=1}^{N} f(X_n)$$

  - **Unbiased:** $\mathbb{E}[\hat{\pi}[f]]] = \pi[f]$

MONTE CARLO ESTIMATOR

- Given iid samples from $X_1, \ldots, X_N \sim \pi$, define the Monte Carlo (MC) estimator:

$$\hat{\pi}[f] = \frac{1}{N} \sum_{n=1}^{N} f(X_n)$$

  - **Unbiased:** $$\mathbb{E}[\hat{\pi}[f]] = \pi[f]$$
  - **Consistent:** convergence is a.s.

$$\lim_{N \to \infty} \hat{\pi}[f] = \pi[f]$$

MONTE CARLO ESTIMATOR

- Given iid samples from $X_1, \ldots, X_N \sim \pi$, define the Monte Carlo (MC) estimator:

$$\hat{\pi}[f] = \frac{1}{N}\sum_{n=1}^{N} f(X_n)$$

  - **Unbiased:** $$\mathbb{E}[\hat{\pi}[f]]] = \pi[f]$$
  - **Consistent:** convergence is a.s.

$$\lim_{N\to\infty} \hat{\pi}[f] = \pi[f]$$

  - **Variance:**

$$\mathbb{V}\left[\hat{\pi}[f]\right] = \frac{\mathbb{V}_\pi[f]}{N}, \quad \mathbb{V}_\pi[f] = \pi[f^2] - \pi[f]^2$$

MONTE CARLO ESTIMATOR

- Given iid samples from $X_1, \ldots, X_N \sim \pi$, define the Monte Carlo (MC) estimator:

$$\hat{\pi}[f] = \frac{1}{N} \sum_{n=1}^{N} f(X_n)$$

  - **Unbiased:** $$\mathbb{E}[\hat{\pi}[f]] = \pi[f]$$
  - **Consistent:** convergence is a.s.

$$\lim_{N \to \infty} \hat{\pi}[f] = \pi[f]$$

  - **Variance:**

$$\mathbb{V}\left[\hat{\pi}[f]\right] = \frac{\mathbb{V}_\pi[f]}{N}, \quad \mathbb{V}_\pi[f] = \pi[f^2] - \pi[f]^2$$

  - **Central limit theorem:** convergence is weak

$$\sqrt{N}(\hat{\pi}[f] - \pi[f]) \implies N(0, \mathbb{V}_\pi[f])$$

MARKOV CHAIN MONTE CARLO

MARKOV CHAIN MONTE CARLO

- **Problem:** Generating samples is hard…

MARKOV CHAIN MONTE CARLO

- **Problem:** Generating samples is hard…
- **Solution:** Markov chain Monte Carlo (MCMC)

MARKOV CHAIN MONTE CARLO

- **Problem:** Generating samples is hard…
- **Solution:** Markov chain Monte Carlo (MCMC)
  - A class of MC methods that construct a Markov chain X_t that is stationary with respect to target π

MARKOV CHAIN MONTE CARLO

- **Problem:** Generating samples is hard…
- **Solution:** Markov chain Monte Carlo (MCMC)
  - A class of MC methods that construct a Markov chain X_t that is stationary with respect to target π
  - E.g. metropolis-Hastings, MALA, HMC, slice sampler, etc

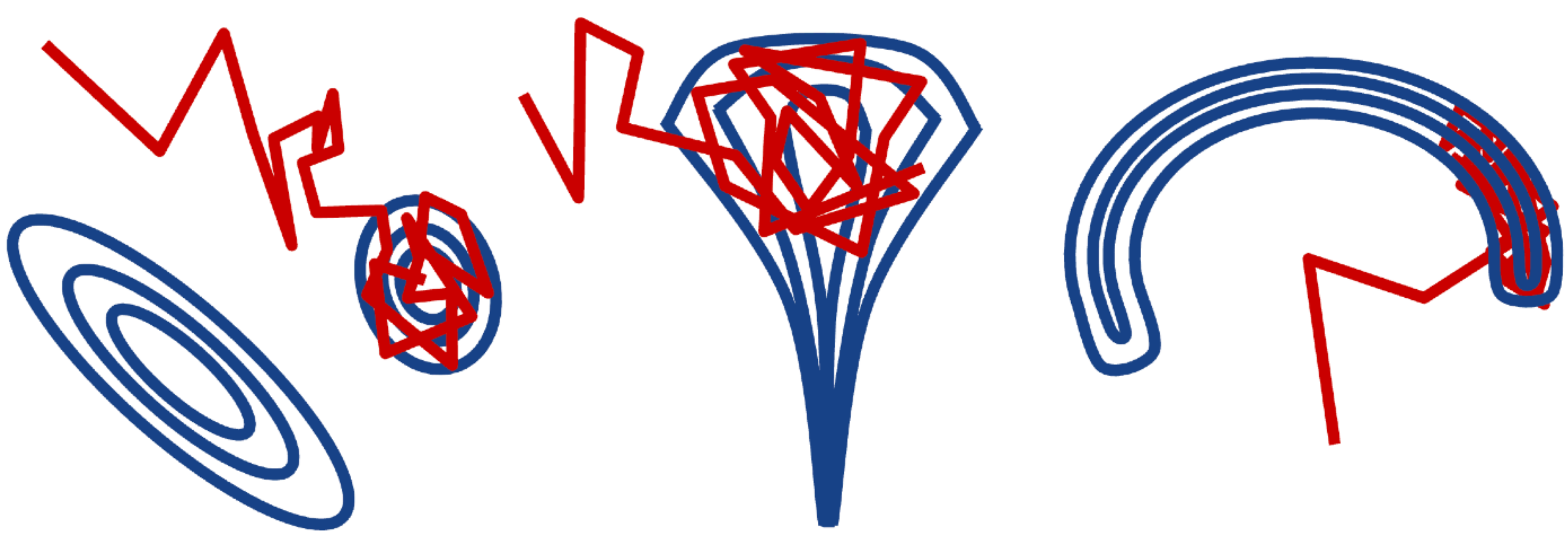
MARKOV CHAIN MONTE CARLO

- **Problem:** Generating samples is hard…
- **Solution:** Markov chain Monte Carlo (MCMC)
  - A class of MC methods that construct a Markov chain X_t that is stationary with respect to target π
  - E.g. metropolis-Hastings, MALA, HMC, slice sampler, etc
  - By the law of large numbers, the average a.s. converges to $\pi[f]$ as $T \to \infty$

$$\pi[f] = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} f(X_t)$$

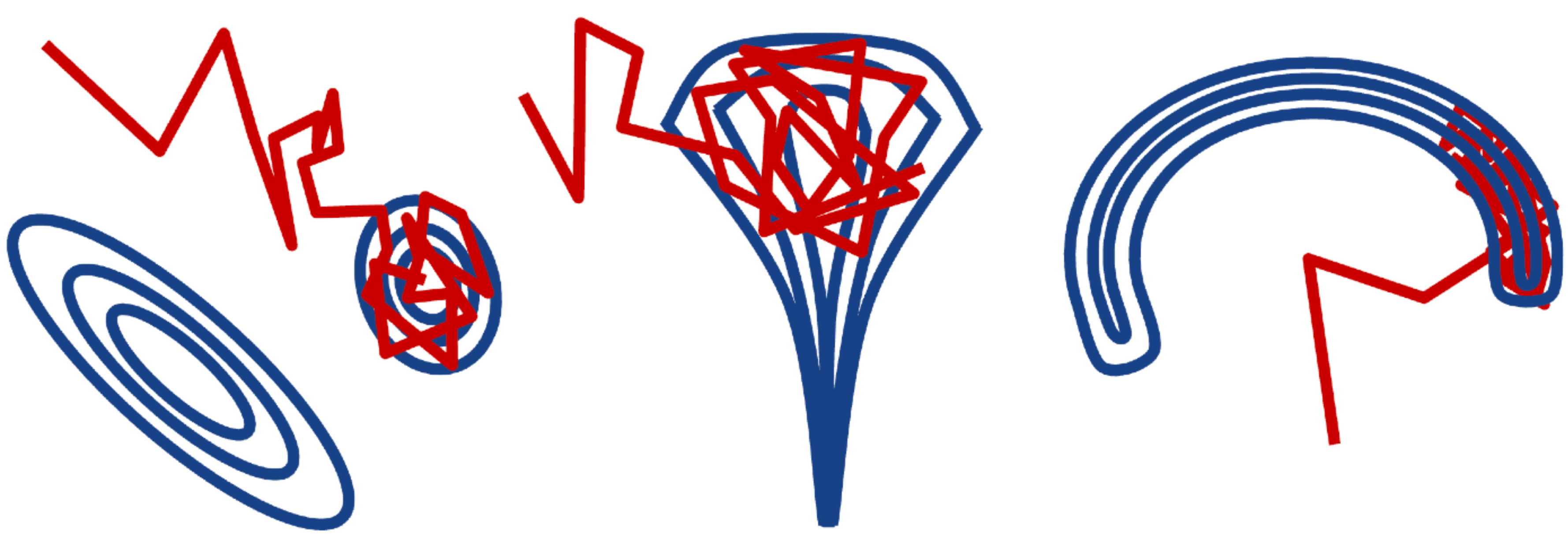
PROBLEMS WITH MCMC

- **Problem:** MCMC kinda sucks…



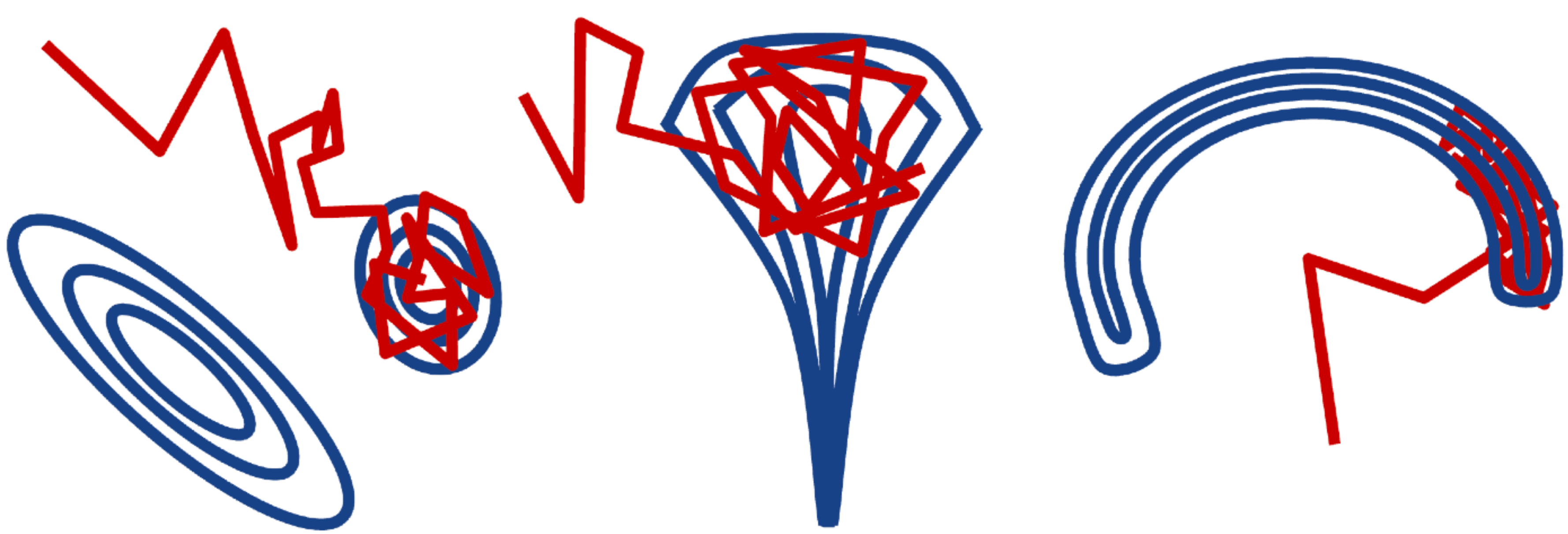
PROBLEMS WITH MCMC

- **Problem:** MCMC kinda sucks…
    - Can take a long time to converge and approximate π



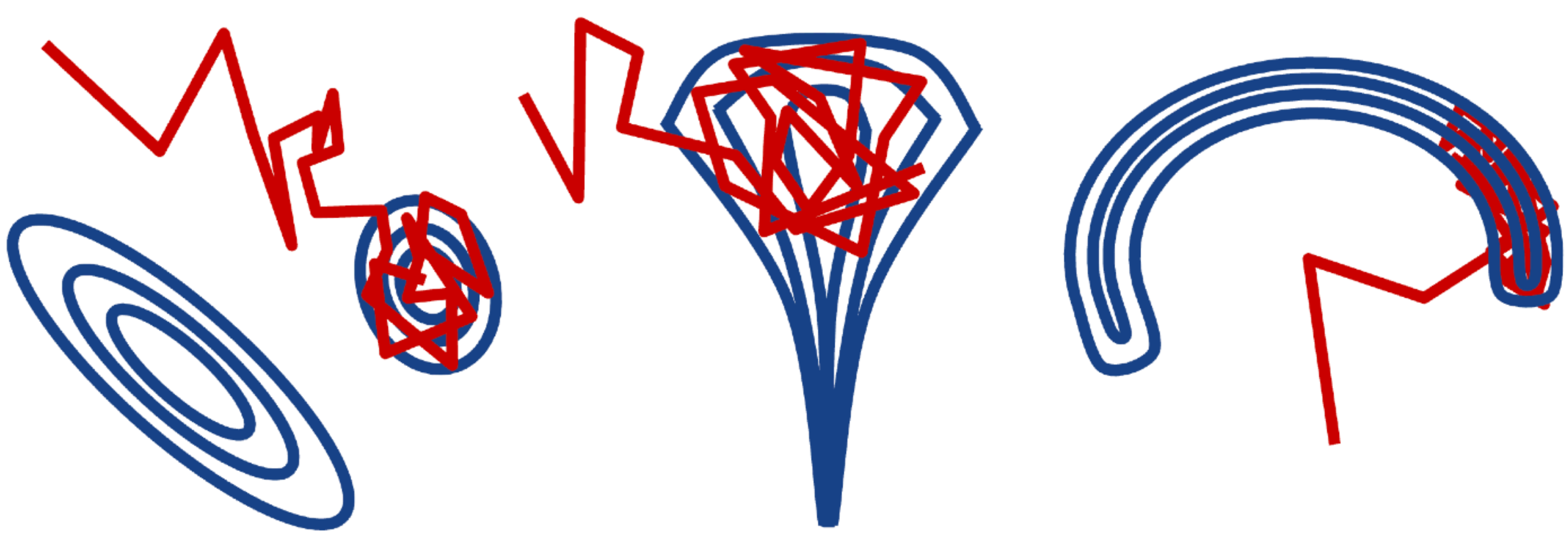
PROBLEMS WITH MCMC

- **Problem:** MCMC kinda sucks…
  - Can take a long time to converge and approximate π
  - After convergence, samples are correlated



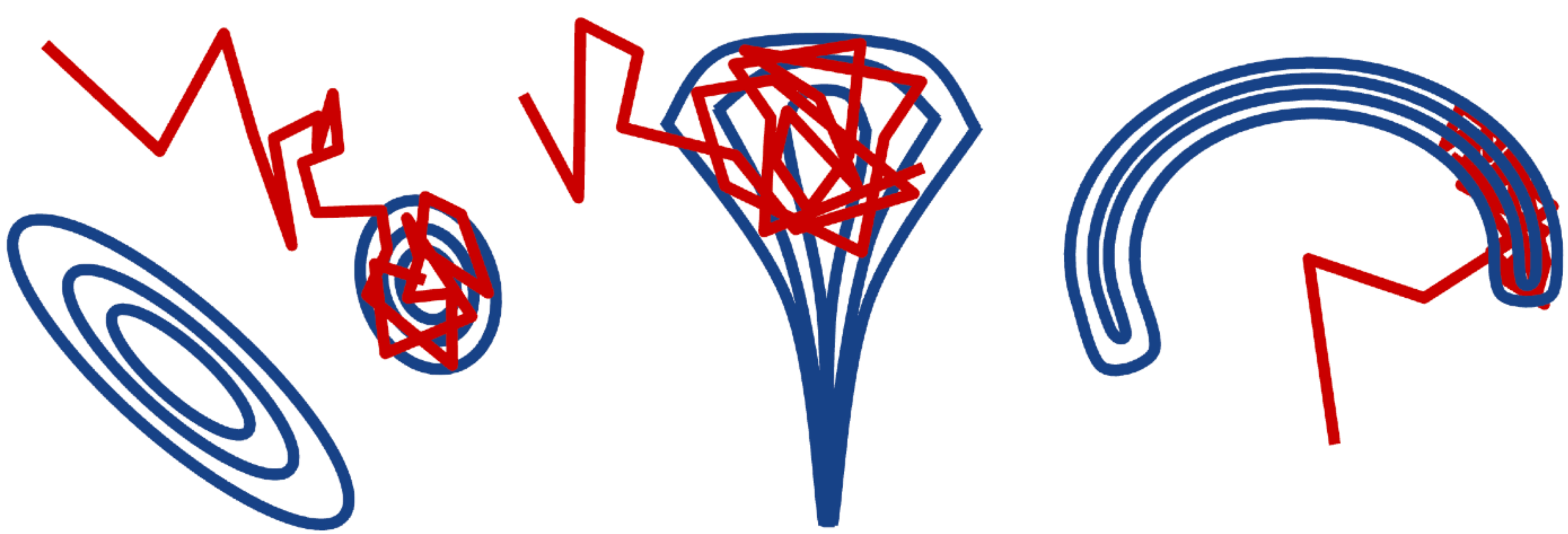
PROBLEMS WITH MCMC

- **Problem:** MCMC kinda sucks…
  - Can take a long time to converge and approximate π
  - After convergence, samples are correlated
  - Sensitive to tuning hyper-parmameters



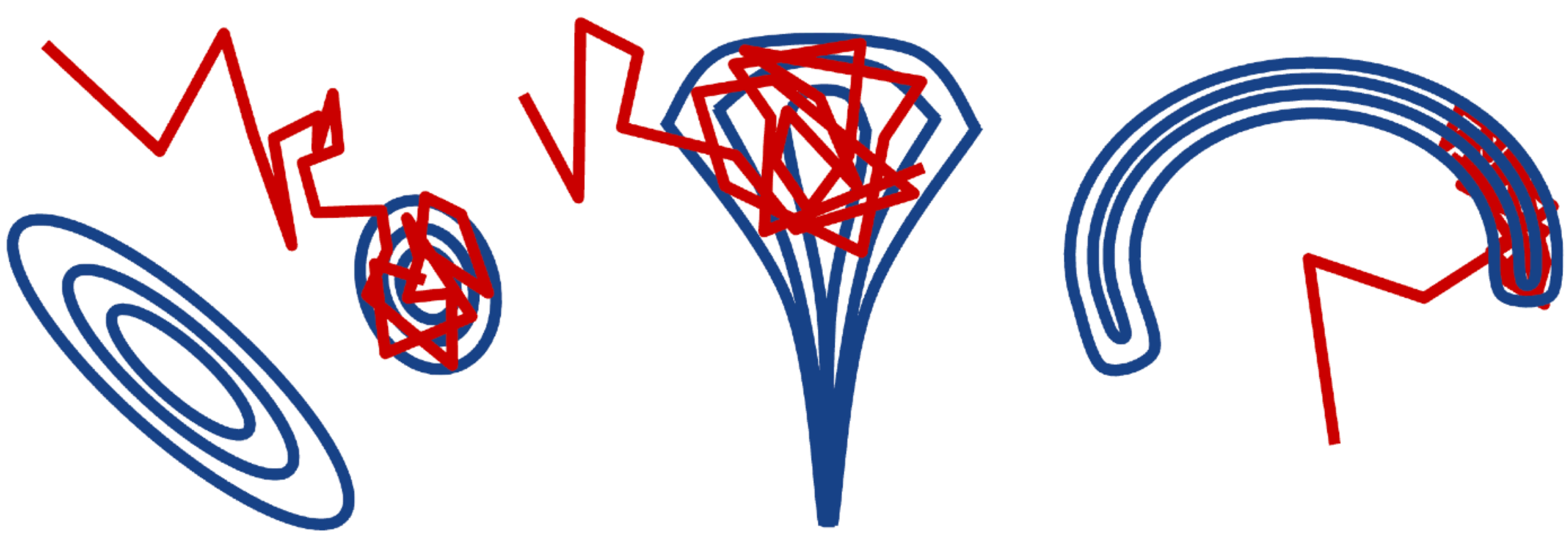
PROBLEMS WITH MCMC

- **Problem:** MCMC kinda sucks…
  - Can take a long time to converge and approximate π
  - After convergence, samples are correlated
  - Sensitive to tuning hyper-parmameters
  - Challening to effectively parallelise



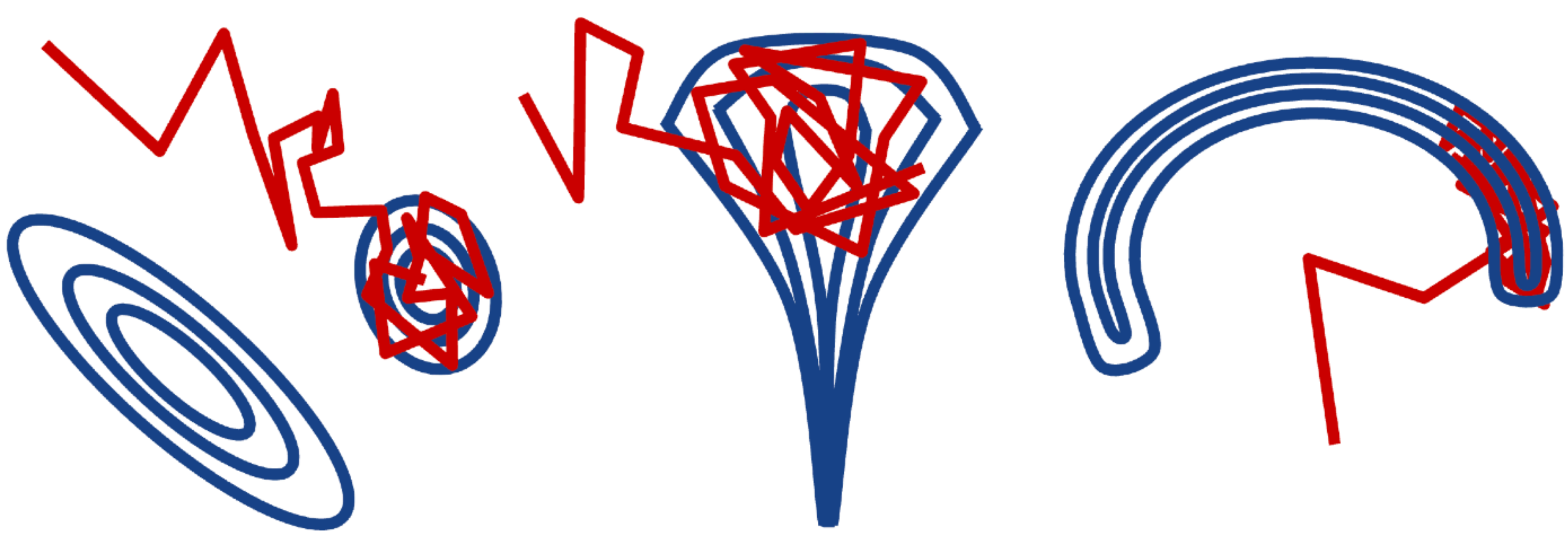
PROBLEMS WITH MCMC

- **Problem:** MCMC kinda sucks…
  - Can take a long time to converge and approximate π
  - After convergence, samples are correlated
  - Sensitive to tuning hyper-parmameters
  - Challening to effectively parallelise
  - Often gets trapped in regions of high probability



PROBLEMS WITH MCMC

- **Problem:** MCMC kinda sucks…
  - Can take a long time to converge and approximate π
  - After convergence, samples are correlated
  - Sensitive to tuning hyper-parmameters
  - Challening to effectively parallelise
  - Often gets trapped in regions of high probability
  - Converge exponentially fast within a mode but exponentiallty slow between modes



NORMALISING CONSTANT

NORMALISING CONSTANT

- **Problem**: MCMC can’t measure the normalising constant Z

NORMALISING CONSTANT

- **Problem**: MCMC can't measure the normalising constant Z
  - Un-normalised density can only compare between two states

NORMALISING CONSTANT

- **Problem**: MCMC can't measure the normalising constant Z
  - Un-normalised density can only compare between two states

- **Solution**: Introduce a reference distribution η (e.g gaussian) that dominates target

NORMALISING CONSTANT

- **Problem**: MCMC can't measure the normalising constant Z
  - Un-normalised density can only compare between two states

- **Solution**: Introduce a reference distribution η (e.g gaussian) that dominates target
  - By the Radon-Nikodym theorem

$$Z = \int_{\mathbb{X}} \gamma(x)\mathrm{d}x = \int_{\mathbb{X}} w(x)\eta(x)\mathrm{d}x = \eta[w], \quad w = \frac{\gamma(x)}{\eta(x)}$$

NORMALISING CONSTANT

- **Problem**: MCMC can't measure the normalising constant Z
  - Un-normalised density can only compare between two states

- **Solution**: Introduce a reference distribution η (e.g gaussian) that dominates target
  - By the Radon-Nikodym theorem

$$Z = \int_{\mathbb{X}} \gamma(x)\mathrm{d}x = \int_{\mathbb{X}} w(x)\eta(x)\mathrm{d}x = \eta[w], \quad w = \frac{\gamma(x)}{\eta(x)}$$

  - Can estimate Z using the Monte Carlo estimator

$$X_1, \ldots, X_N \sim \eta \qquad \hat{Z} = \frac{1}{N}\sum_{n=1}^{N} w(X_n)$$

NORMALISING CONSTANT

- **Unbiased:**

$$\mathbb{E}[\hat{Z}] = \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}[w(X_n)] = Z$$

- **Consistent:** convergence is a.s.

$$\lim_{N\to\infty}\hat{Z} = \lim_{N\to\infty}\frac{1}{N}\sum_{n=1}^{N} w(X_n) = Z$$

- **Variance:**

$$\mathbb{V}\left[\frac{\hat{Z}}{Z}\right] = \frac{1}{N}\mathbb{V}_{X\sim\eta}\left[\frac{\pi(X)}{\eta(X)}\right] = \frac{e^{D_2(\pi\|\eta)} - 1}{N}$$

  - The variance grows exponentially as η deviates from the target

$$D_2(\pi\|\eta) = \log\left(1 + \mathbb{V}_{X\sim\eta}\left[\frac{\pi(X)}{\eta(X)}\right]\right) \geq \mathrm{KL}(\eta\|\pi)$$

INTEGRALS ARE NORMALISING CONSTANTS

INTEGRALS ARE NORMALISING CONSTANTS

- What if I don't care about *every* statistic? I just want to compute $\pi[f]$ for some f

INTEGRALS ARE NORMALISING CONSTANTS

- What if I don't care about *every* statistic? I just want to compute $\pi[f]$ for some f
  - Define a the f-tilted target π_f distribution:

$$\pi_f(\mathrm{d}x) = \frac{\gamma_f(x)}{Z_f}\mathrm{d}x \qquad \gamma_f(x) = f(x)\gamma(x)$$

INTEGRALS ARE NORMALISING CONSTANTS

- What if I don't care about *every* statistic? I just want to compute $\pi[f]$ for some f
    - Define a the f-tilted target π_f distribution:

$$\pi_f(\mathrm{d}x) = \frac{\gamma_f(x)}{Z_f}\mathrm{d}x \qquad\qquad \gamma_f(x) = f(x)\gamma(x)$$

- The normalising constant of the tilted target satisfies:

$$Z_f = \int_{\mathbb{X}} \gamma_f(x)\mathrm{d}x = \int_{\mathbb{X}} f(x)w(x)\eta(x)\mathrm{d}x = \eta[fw],$$

INTEGRALS ARE NORMALISING CONSTANTS

- What if I don't care about _every_ statistic? I just want to compute $\pi[f]$ for some f
  - Define a the f-tilted target π_f distribution:

$$\pi_f(\mathrm{d}x) = \frac{\gamma_f(x)}{Z_f}\mathrm{d}x \qquad\qquad \gamma_f(x) = f(x)\gamma(x)$$

- The normalising constant of the tilted target satisfies:

$$Z_f = \int_{\mathbb{X}} \gamma_f(x)\mathrm{d}x = \int_{\mathbb{X}} f(x)w(x)\eta(x)\mathrm{d}x = \eta[fw],$$

- Normalising constants are equivalent to computing a single expectation:

INTEGRALS ARE NORMALISING CONSTANTS

- What if I don't care about *every* statistic? I just want to compute $\pi[f]$ for some f
  - Define a the f-tilted target π_f distribution:

$$\pi_f(\mathrm{d}x) = \frac{\gamma_f(x)}{Z_f}\mathrm{d}x \qquad\qquad \gamma_f(x) = f(x)\gamma(x)$$

- The normalising constant of the tilted target satisfies:

$$Z_f = \int_{\mathbb{X}} \gamma_f(x)\mathrm{d}x = \int_{\mathbb{X}} f(x)w(x)\eta(x)\mathrm{d}x = \eta[fw],$$

- Normalising constants are equivalent to computing a single expectation:

$$\pi[f] = \int_{\mathbb{X}} f(x)\pi(x)\mathrm{d}x$$

INTEGRALS ARE NORMALISING CONSTANTS

- What if I don't care about *every* statistic? I just want to compute $\pi[f]$ for some f
  - Define a the f-tilted target π_f distribution:

$$\pi_f(\mathrm{d}x) = \frac{\gamma_f(x)}{Z_f}\mathrm{d}x \qquad\qquad \gamma_f(x) = f(x)\gamma(x)$$

- The normalising constant of the tilted target satisfies:

$$Z_f = \int_{\mathbb{X}} \gamma_f(x)\mathrm{d}x = \int_{\mathbb{X}} f(x)w(x)\eta(x)\mathrm{d}x = \eta[fw],$$

- Normalising constants are equivalent to computing a single expectation:

$$\pi[f] = \int_{\mathbb{X}} f(x)\pi(x)\mathrm{d}x \quad = \frac{\int_{\mathbb{X}} f(x)\gamma(x)\mathrm{d}x}{Z}$$

INTEGRALS ARE NORMALISING CONSTANTS

- What if I don't care about *every* statistic? I just want to compute $\pi[f]$ for some f
    - Define a the f-tilted target π_f distribution:

$$\pi_f(\mathrm{d}x) = \frac{\gamma_f(x)}{Z_f}\mathrm{d}x \qquad \gamma_f(x) = f(x)\gamma(x)$$

- The normalising constant of the tilted target satisfies:

$$Z_f = \int_{\mathbb{X}} \gamma_f(x)\mathrm{d}x = \int_{\mathbb{X}} f(x)w(x)\eta(x)\mathrm{d}x = \eta[fw],$$

- Normalising constants are equivalent to computing a single expectation:

$$\pi[f] = \int_{\mathbb{X}} f(x)\pi(x)\mathrm{d}x = \frac{\int_{\mathbb{X}} f(x)\gamma(x)\mathrm{d}x}{Z} = \frac{Z_f}{Z}$$

INTEGRALS ARE NORMALISING CONSTANTS

- What if I don't care about *every* statistic? I just want to compute $\pi[f]$ for some f
  - Define a the f-tilted target π_f distribution:

$$\pi_f(\mathrm{d}x) = \frac{\gamma_f(x)}{Z_f}\mathrm{d}x \qquad\qquad \gamma_f(x) = f(x)\gamma(x)$$

- The normalising constant of the tilted target satisfies:

$$Z_f = \int_{\mathbb{X}} \gamma_f(x)\mathrm{d}x = \int_{\mathbb{X}} f(x)w(x)\eta(x)\mathrm{d}x = \eta[fw],$$

- Normalising constants are equivalent to computing a single expectation:

$$\pi[f] = \int_{\mathbb{X}} f(x)\pi(x)\mathrm{d}x = \frac{\int_{\mathbb{X}} f(x)\gamma(x)\mathrm{d}x}{Z} = \frac{Z_f}{Z} = \frac{\eta[fw]}{\eta[w]}$$

MONTE CARLO AND COMPUTATION

MONTE CARLO AND COMPUTATION

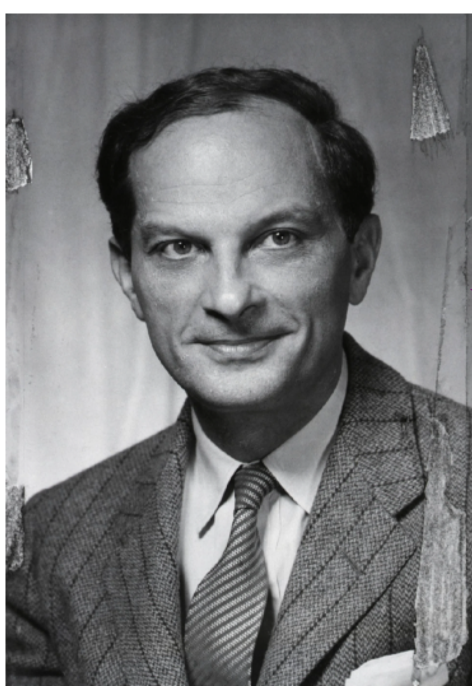
- The history of Monte Carlo and computation are intimately related

MONTE CARLO AND COMPUTATION

- The history of Monte Carlo and computation are intimately related
- Monte Carlo was first used during the Manhattan Project with the ENIAC computer

MONTE CARLO AND COMPUTATION

- The history of Monte Carlo and computation are intimately related
- Monte Carlo was first used during the Manhattan Project with the ENIAC computer
    - Stanislaw Ulam used in 1946 to estimate the probability of winning solitaire



HEY JOHNNY,
BRO LOOK AT THIS 💀!

CAP!

NO CAP THIS
HITS DIFFERENT FR

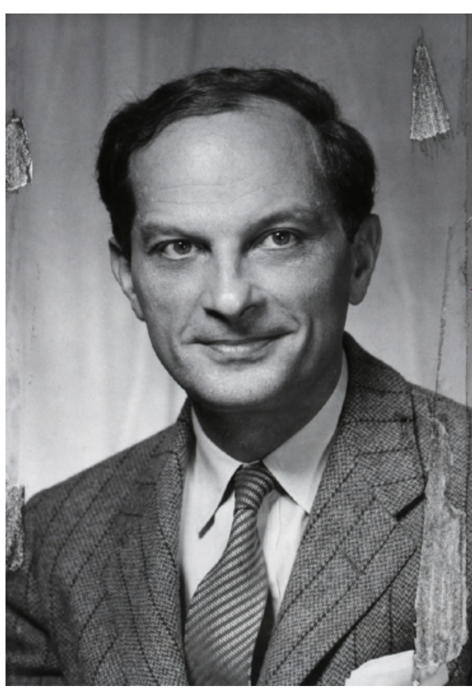
THAT'S
STRAIGHT SIGMA

Stanislaw Ulam

John Von Neumann

MONTE CARLO AND COMPUTATION

- The history of Monte Carlo and computation are intimately related
- Monte Carlo was first used during the Manhattan Project with the ENIAC computer
  - Stanislaw Ulam used in 1946 to estimate the probability of winning solitaire
  - Neumann applied it to the bomb



Stanislaw Ulam

HEY JOHNNY,
BRO LOOK AT THIS 💀!

NO CAP THIS
HITS DIFFERENT FR

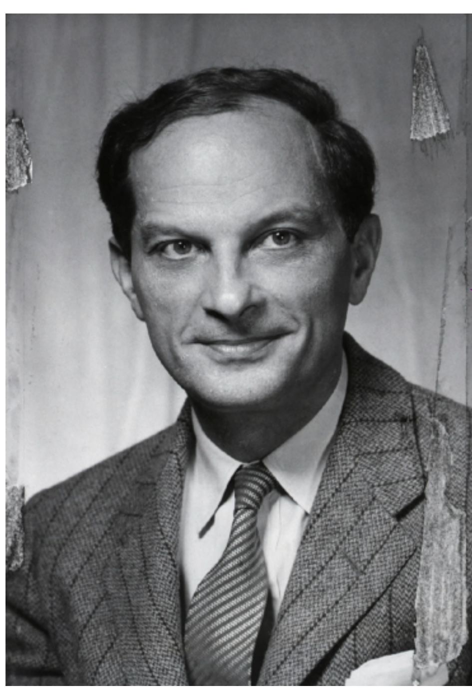
CAP!

THAT'S
STRAIGHT SIGMA

John Von Neumann

MONTE CARLO AND COMPUTATION

- The history of Monte Carlo and computation are intimately related
- Monte Carlo was first used during the Manhattan Project with the ENIAC computer
  - Stanislaw Ulam used in 1946 to estimate the probability of winning solitaire
  - Neumann applied it to the bomb
  - First time computation was used as a replacement for math



Stanislaw Ulam

HEY JOHNNY,
BRO LOOK AT THIS 💀!

CAP!

NO CAP THIS
HITS DIFFERENT FR

THAT'S
STRAIGHT SIGMA

John Von Neumann

MONTE CARLO AND COMPUTATION

MONTE CARLO AND COMPUTATION

- There is an arms race between hardware, models and algorithms

MONTE CARLO AND COMPUTATION

- There is an arms race between hardware, models and algorithms
  - As hardware improves previous hard problems become sub-routines

MONTE CARLO AND COMPUTATION

- There is an arms race between hardware, models and algorithms
  - As hardware improves previous hard problems become sub-routines
  - We always find a way to push the limits of hardware to increase model complexity

MONTE CARLO AND COMPUTATION

- There is an arms race between hardware, models and algorithms
  - As hardware improves previous hard problems become sub-routines
  - We always find a way to push the limits of hardware to increase model complexity
  - Adoption in Bayesian inference tied to computational advances with MC methods

MONTE CARLO AND COMPUTATION

- There is an arms race between hardware, models and algorithms
  - As hardware improves previous hard problems become sub-routines
  - We always find a way to push the limits of hardware to increase model complexity
  - Adoption in Bayesian inference tied to computational advances with MC methods
- Our creativity is limited by the tools we can use

MONTE CARLO AND COMPUTATION

- There is an arms race between hardware, models and algorithms
  - As hardware improves previous hard problems become sub-routines
  - We always find a way to push the limits of hardware to increase model complexity
  - Adoption in Bayesian inference tied to computational advances with MC methods
- Our creativity is limited by the tools we can use
- **Goal**: design algorithms that scale with advancements in hardware and model complexity

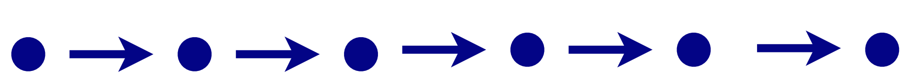
MONTE CARLO AND COMPUTATION

- There is an arms race between hardware, models and algorithms
  - As hardware improves previous hard problems become sub-routines
  - We always find a way to push the limits of hardware to increase model complexity
  - Adoption in Bayesian inference tied to computational advances with MC methods
- Our creativity is limited by the tools we can use
- **Goal**: design algorithms that scale with advancements in hardware and model complexity
  - Modern hardware advancements have come with increased parallism

MONTE CARLO AND COMPUTATION

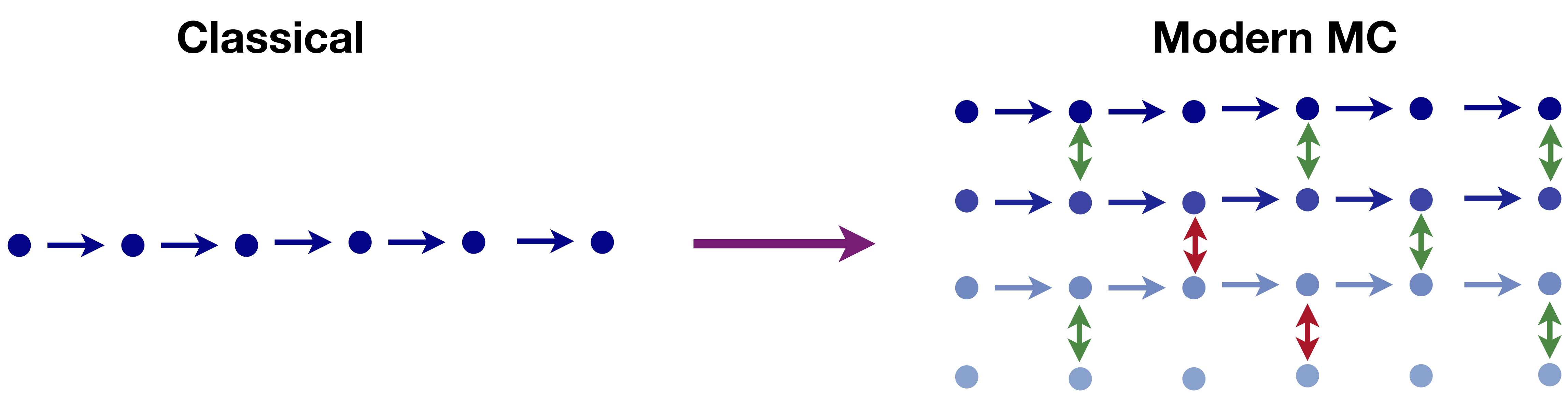
- There is an arms race between hardware, models and algorithms
  - As hardware improves previous hard problems become sub-routines
  - We always find a way to push the limits of hardware to increase model complexity
  - Adoption in Bayesian inference tied to computational advances with MC methods
- Our creativity is limited by the tools we can use
- **Goal**: design algorithms that scale with advancements in hardware and model complexity
  - Modern hardware advancements have come with increased parallism

Classical



MONTE CARLO AND COMPUTATION

- There is an arms race between hardware, models and algorithms
  - As hardware improves previous hard problems become sub-routines
  - We always find a way to push the limits of hardware to increase model complexity
  - Adoption in Bayesian inference tied to computational advances with MC methods
- Our creativity is limited by the tools we can use
- **Goal**: design algorithms that scale with advancements in hardware and model complexity
  - Modern hardware advancements have come with increased parallism



GOAL OF THIS COURSE

GOAL OF THIS COURSE

- As a practitioner, you want to drive the car and don't want to worry about your engine

GOAL OF THIS COURSE

- As a practitioner, you want to drive the car and don’t want to worry about your engine
- This goal of this course is to make you a responsible driver and do basic mantainance

GOAL OF THIS COURSE

- As a practitioner, you want to drive the car and don’t want to worry about your engine
- This goal of this course is to make you a responsible driver and do basic mantainance

GOAL OF THIS COURSE

- As a practitioner, you want to drive the car and don't want to worry about your engine
- This goal of this course is to make you a responsible driver and do basic mantainance
    - Develop best practices for using the sampling algorithm for your own problems

GOAL OF THIS COURSE

- As a practitioner, you want to drive the car and don’t want to worry about your engine
- This goal of this course is to make you a responsible driver and do basic mantainance
  - Develop best practices for using the sampling algorithm for your own problems
  - Give you a peak under the hood to understand why these algorithms work

GOAL OF THIS COURSE

- As a practitioner, you want to drive the car and don't want to worry about your engine
- This goal of this course is to make you a responsible driver and do basic mantainance
  - Develop best practices for using the sampling algorithm for your own problems
  - Give you a peak under the hood to understand why these algorithms work
  - Develop practical guidelines to diagnose and interpret outputs

GOAL OF THIS COURSE

- As a practitioner, you want to drive the car and don't want to worry about your engine
- This goal of this course is to make you a responsible driver and do basic mantainance
  - Develop best practices for using the sampling algorithm for your own problems
  - Give you a peak under the hood to understand why these algorithms work
  - Develop practical guidelines to diagnose and interpret outputs
  - Give you the terminology and foundations to read papers and not be scared

GOAL OF THIS COURSE

- As a practitioner, you want to drive the car and don’t want to worry about your engine
- This goal of this course is to make you a responsible driver and do basic mantainance
  - Develop best practices for using the sampling algorithm for your own problems
  - Give you a peak under the hood to understand why these algorithms work
  - Develop practical guidelines to diagnose and interpret outputs
  - Give you the terminology and foundations to read papers and not be scared

NOT THE GOAL OF THIS COURSE

NOT THE GOAL OF THIS COURSE

- This course is not to make you a engineer

NOT THE GOAL OF THIS COURSE

- This course is not to make you a engineer

NOT THE GOAL OF THIS COURSE

- This course is not to make you a engineer
  - This is not a theory course!

NOT THE GOAL OF THIS COURSE

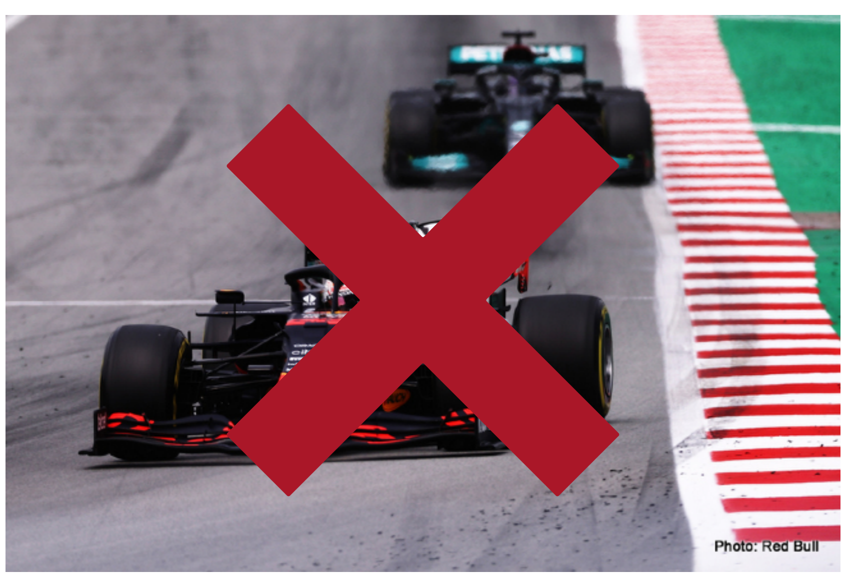
- This course is not to make you a engineer
  - This is not a theory course!
  - We will use theory to motivate methodology

NOT THE GOAL OF THIS COURSE

- This course is not to make you a engineer
  - This is not a theory course!
  - We will use theory to motivate methodology
- This course is not to make you a racecar driver:

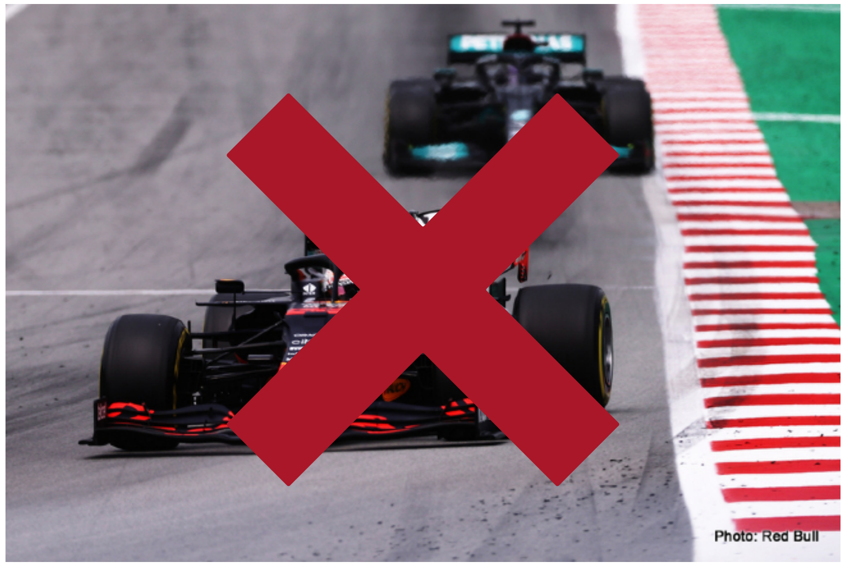
NOT THE GOAL OF THIS COURSE

- This course is not to make you a engineer
  - This is not a theory course!
  - We will use theory to motivate methodology
- This course is not to make you a racecar driver:



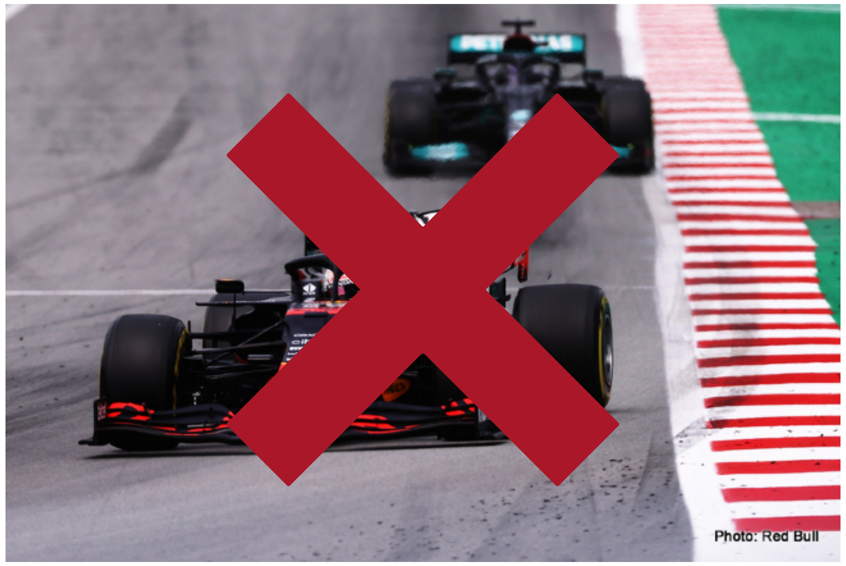
NOT THE GOAL OF THIS COURSE

- This course is not to make you a engineer
  - This is not a theory course!
  - We will use theory to motivate methodology
- This course is not to make you a racecar driver:
  - This is not a course on probabilisitic programming or high performance computing



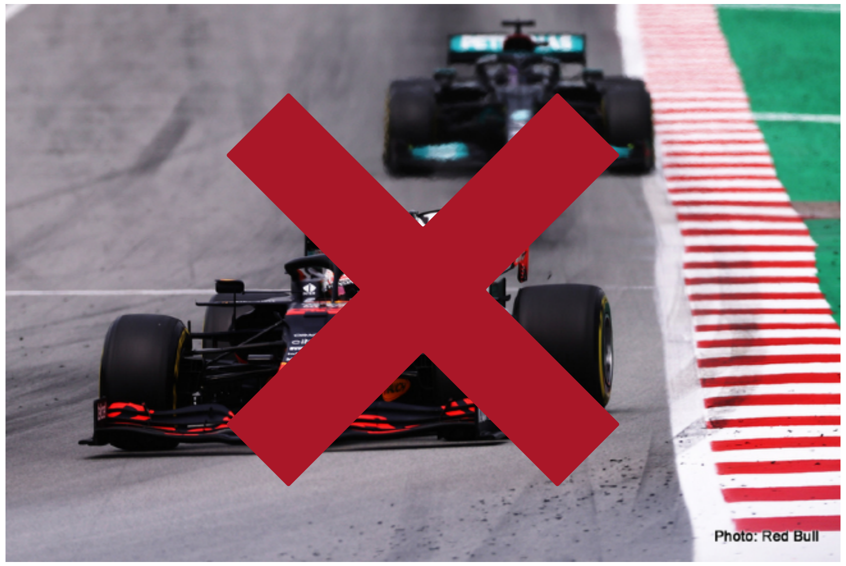
NOT THE GOAL OF THIS COURSE

- This course is not to make you a engineer
  - This is not a theory course!
  - We will use theory to motivate methodology
- This course is not to make you a racecar driver:
  - This is not a course on probabilisitic programming or high performance computing
  - Hardware, packages, languages, models, etc change



NOT THE GOAL OF THIS COURSE

- This course is not to make you a engineer
  - This is not a theory course!
  - We will use theory to motivate methodology
- This course is not to make you a racecar driver:
  - This is not a course on probabilisitic programming or high performance computing
  - Hardware, packages, languages, models, etc change
  - Implemations in practice have their own challenges that are not our focus



COURSE OUTLINE

- **Part 1: Foundations**
  - MCMC theory
  - Local inference algorithms
  - Annealing
- **Part 2: Annealing algorithms**
  - Parallel annealing
  - Sequential annealing
- **Part 3: Free energy methods**
  - Acceleration methods
  - Enhanced sampling
  - Optional topics