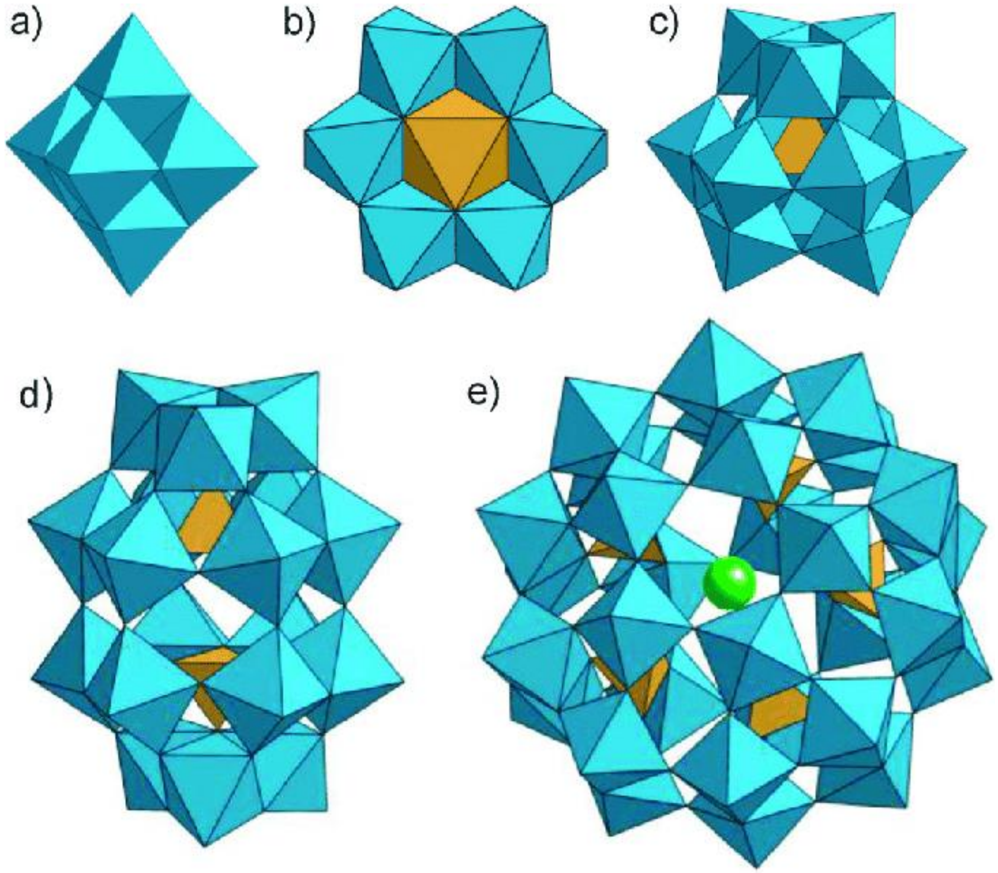
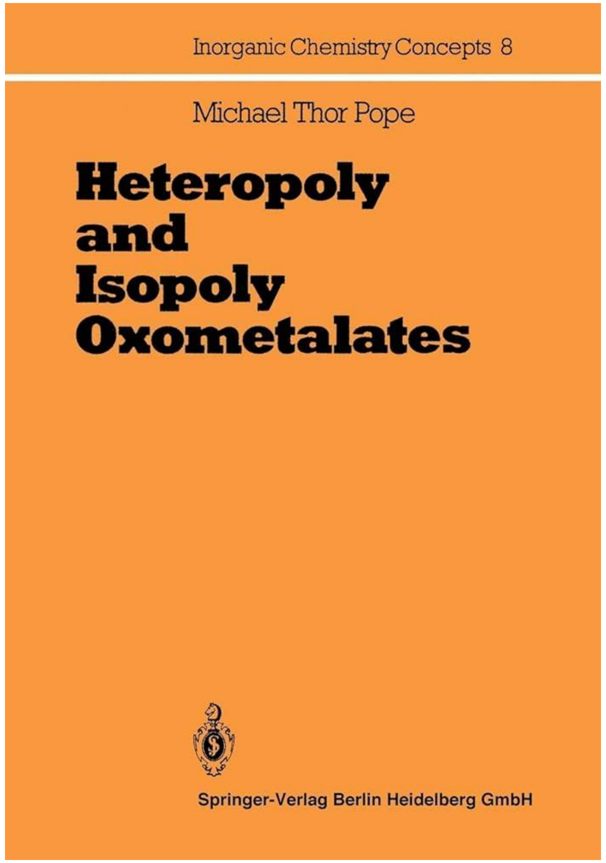
Day 4: Decisions, Beliefs, and Causality

Instructor: Sergei V. Kalinin

Synthesis of polyoxometallates



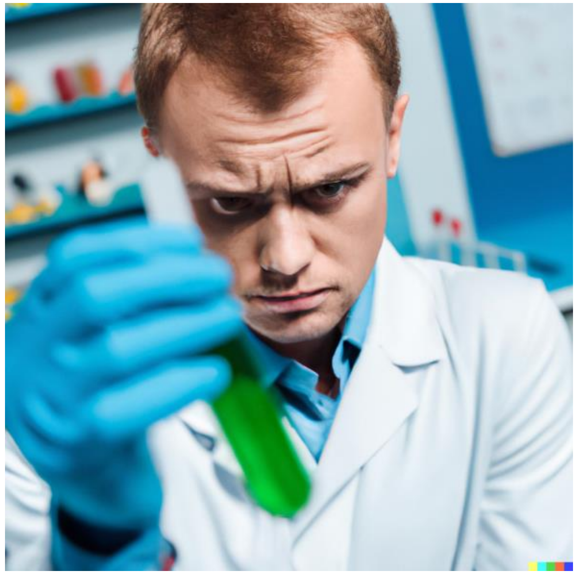


X. López, J. Carbó, C. Bo, J. Poblet, *Structure, properties and reactivity of polyoxometalates: A theoretical perspective*, Chemical Society reviews 71, 7537 (2012)

Synthesis of polyoxometallates

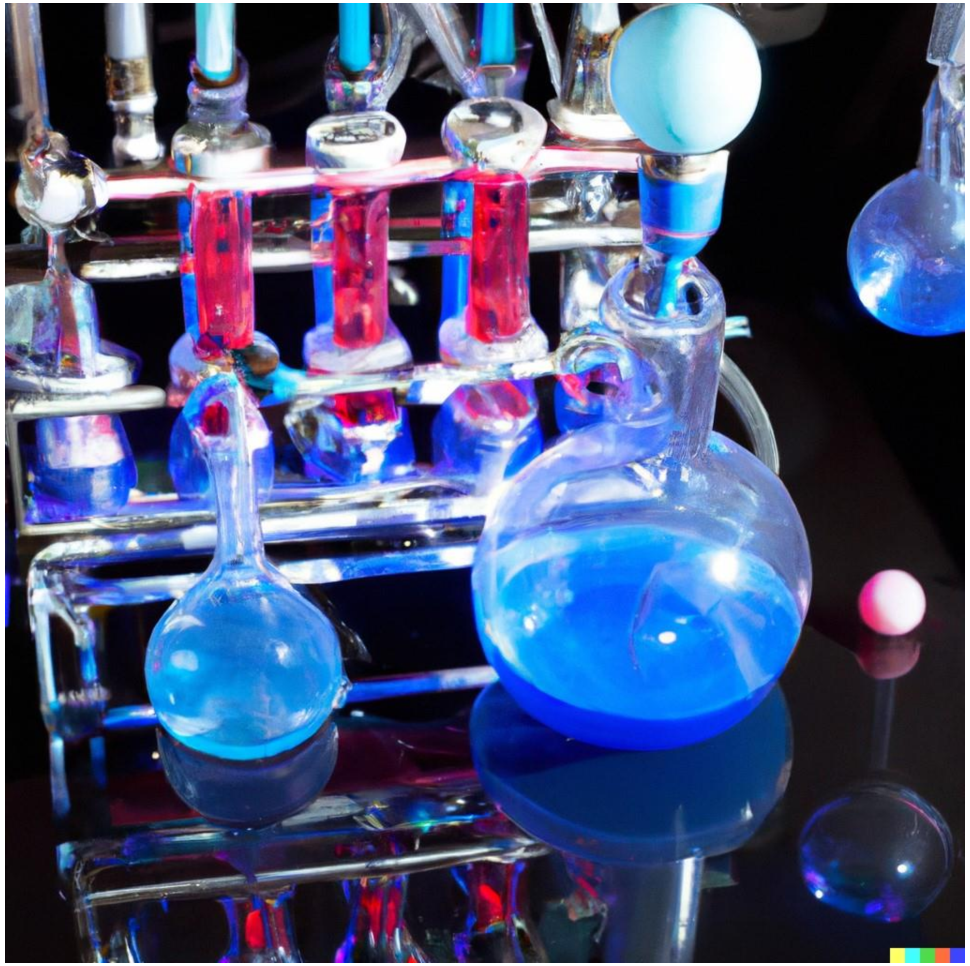
Good day

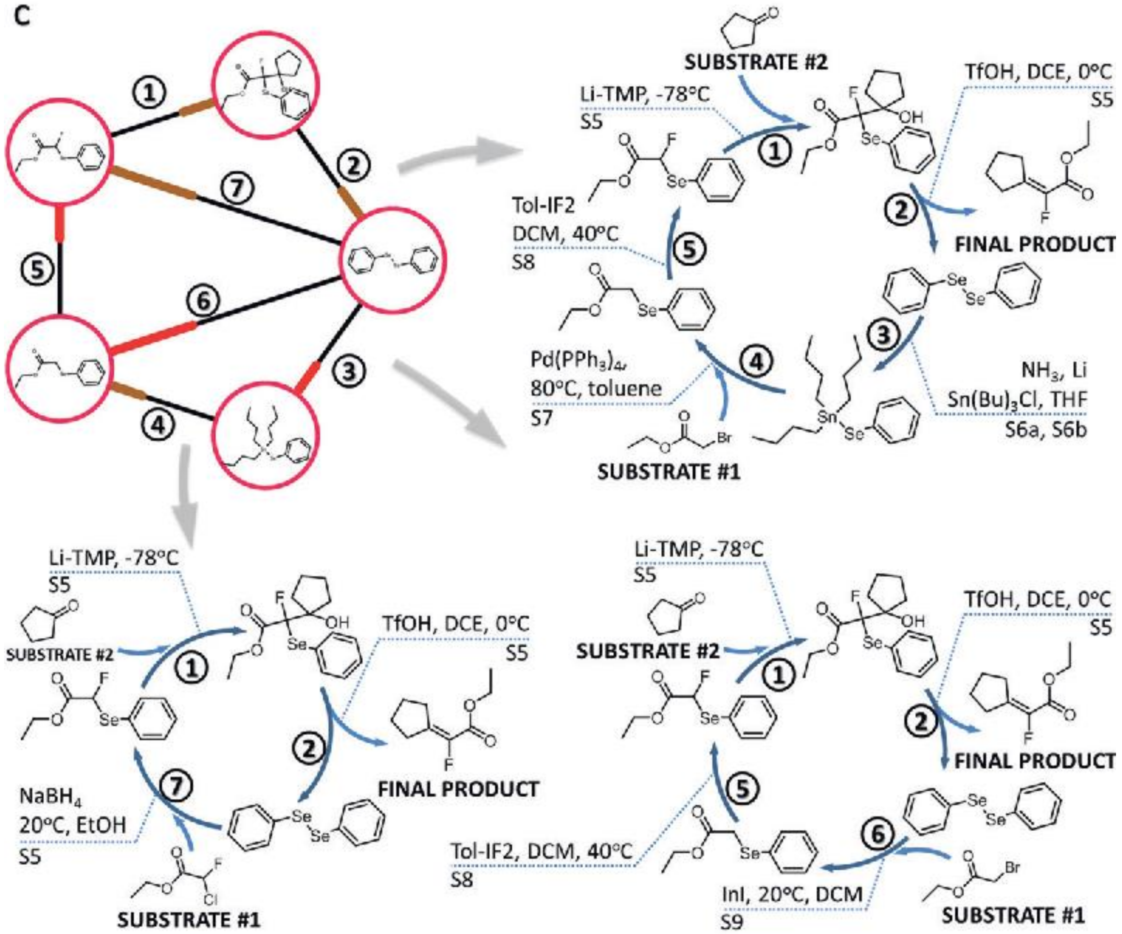
Bad day



- Synthesis may or may not work!
- Depending on difficult to quantify factors
- It’s a coin toss, a.k.a. Bernoulli distribution

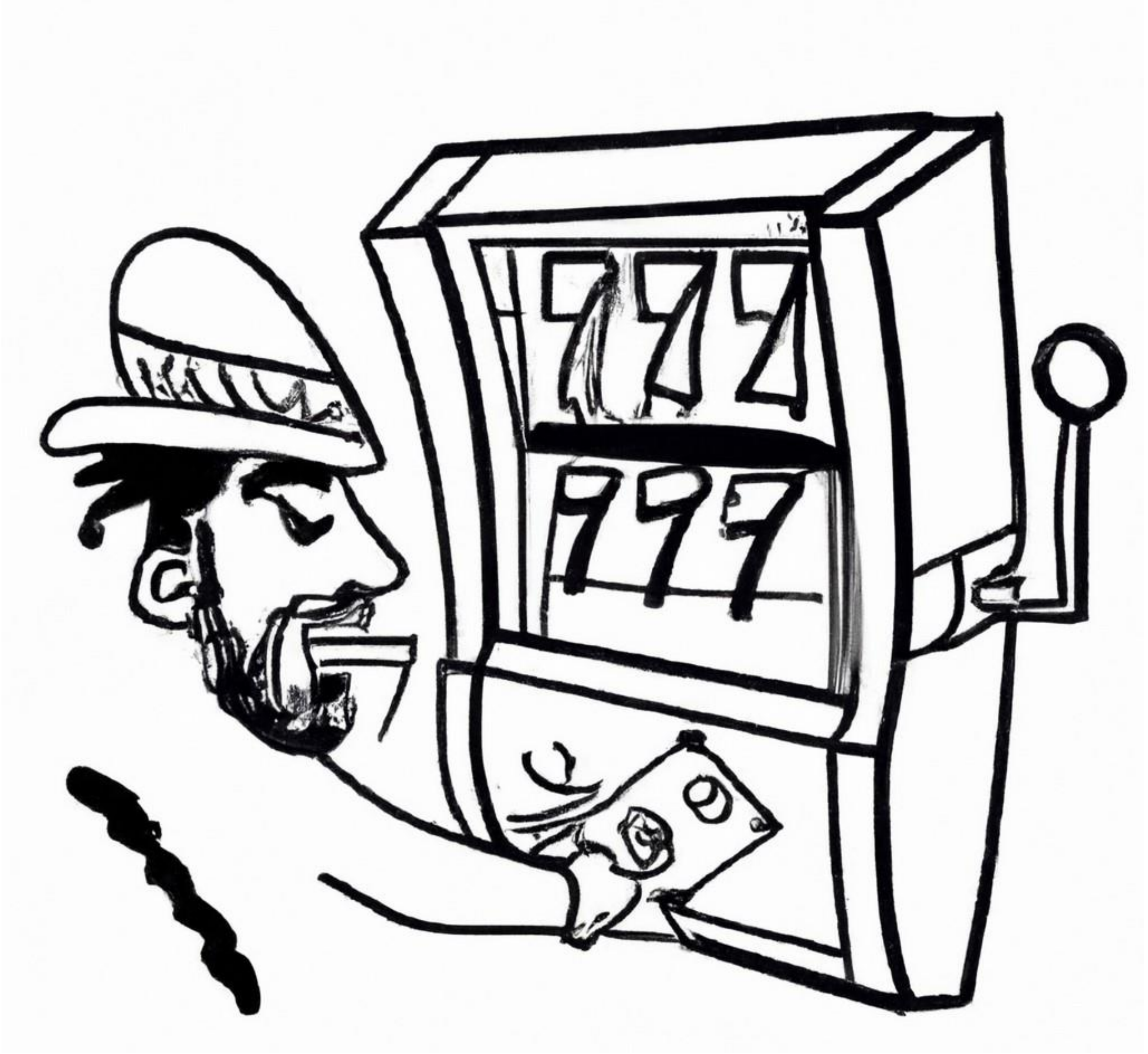
Organic synthesis





M. Bajczyk, P. Dittwald, A. Wołoś, S. Szymkuć, B. Grzybowski, *Discovery and Enumeration of Organic-Chemical and Biomimetic Reaction Cycles within the Network of Chemistry*, DOI:10.1002/anie.201712052

Bandit problem



- Imagine that we have a number of slot machines with different probabilities of win…
- Or different web-sites to places ads on…
- Or groups of patients for specific medical protocol….
- Or team members to synthesize certain material…
- Or reaction pathways to choose

How do we optimize this problem and maximize our reward?

A/B Testing

- The most common exploration strategy is **A/B testing**, a method to determine which one of the two alternatives (of online products, pages, ads etc.) performs better.
- The users are randomly split into two groups to try different alternatives. At the end of the testing period, the results are compared to choose the best alternative, which is then used in production for the rest of the problem horizon.
- This approach can be applied for more then two alternatives - **A/B/n testing**.

From Enes Bilgin, Mastering Reinforcement Learning with Python, Packt 2020

Definitions: reward and value

$$Q_n \triangleq \frac{R_1 + R_2 + \cdots + R_{n-1}}{n-1}$$

- First, we denote the **reward** (i.e. 1 for click, 0 for no click) received after selecting the action *a* for the n^{th} time by R_i.

- Q_n estimates the expected value of the reward that this action yields, R, after n-1 observations.

- Q_n is also called the action value of a. Here, Q_n estimates of the action value after selecting this action n - 1 times.

From Enes Bilgin, Mastering Reinforcement Learning with Python, Packt 2020

Updating value

$$Q_{n+1} = \frac{R_1 + R_2 + \cdots + R_n}{n} = Q_n + \frac{1}{n} \cdot (R_n - Q_n)$$

- Q_n is the estimate for the action value of a before we take it for the n^{th} time.
- When we observe the reward R_n, it gives us another signal for the action value.
- We adjust our current estimate, Q_n, in the direction of the **error** that we calculate based on the latest observed reward, R_n, with a **step size** $1/n$ and obtain a new estimate Q_{n+1}
- For convenience, $Q_0 = 0$ (**But - Human heuristics**!)

From Enes Bilgin, Mastering Reinforcement Learning with Python, Packt 2020

Updating value: generalization

$$Q_{n+1}(a) = Q_n(a) + \alpha(R_n(\alpha) - Q_n(a))$$

- The rate at which we adjust our estimate will get smaller as we make more observations
- The step size must be smaller than 1 for the estimate to converge (and larger than 0 for a proper update).
- Using a fixed α will make the weights of the older observations to decrease exponentially as we take action a more and more.

From Enes Bilgin, Mastering Reinforcement Learning with Python, Packt 2020

Limitations of A/B testing

- **A/B/n testing is inefficient as it does not modify the experiment dynamically by learning from the observations.** It fails to benefit from the early observations in the test by writing off/promoting an alternative even if it is obviously under- or outperforming the others.

- **It is unable to correct a decision once it's made.** There is no way to correct the decision for the rest of the deployment horizon.

- **It is unable to adapt to the changes in a dynamic environment.** If the underlying reward distributions change over time, plain A/B/n testing has no way of detecting such changes after the selection is fixed.

- **The length of the test period is a hyperparameter to tune, affecting the efficiency of the test.** If this period is chosen to be shorter than needed, an incorrect alternative could be declared the best because of the noise in the observations. If the test period is chosen to be too long, too much money gets wasted in exploration.

- **A/B/n testing is simple.** Despite all these shortcomings, it is intuitive and easy to implement, therefore widely used in practice

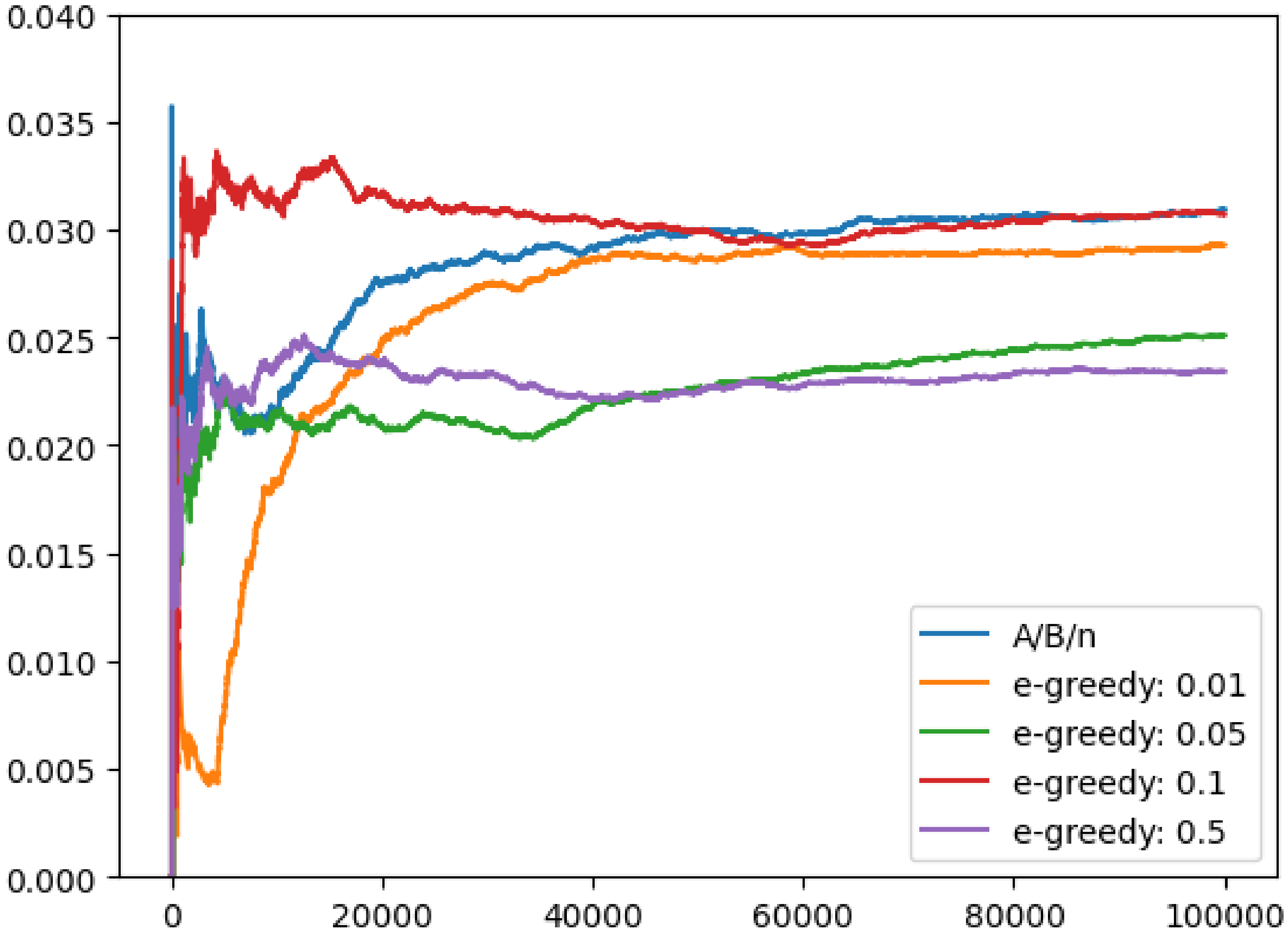
From Enes Bilgin, Mastering Reinforcement Learning with Python, Packt 2020

ε-greedy policies

- Most of the time, greedily taking the action that is the best according to the rewards observed that far in the experiment (i.e. with 1-ε probability)
- Once in a while (i.e. with ε probability) take a random action regardless of the action performances.
- Here ε is a number between 0 and 1, usually closer to zero (e.g. 0.1) to "exploit" in most decisions.
- Obviously, the number of alternatives has to be fairly small
- Parameter ε can change during the experiment

From Enes Bilgin, Mastering Reinforcement Learning with Python, Packt 2020

ε-greedy policies



A/B vs. ε-greedy policies

- ε-greedy actions and A/B/n tests are similarly inefficient and static in allocating the exploration budget. The ε-greedy approach, too, fails to write off actions that are clearly bad and continues to allocate the same exploration budget to each alternative. Similarly, if a particular action is under-explored/over-explored at any point, the exploration budget is not adjusted accordingly.

- With ε-greedy actions, exploration is continuous, unlike in A/B/n testing. This means if the environment is not stationary, the ε-greedy approach has the potential to pick up the changes and modify its selection of the best alternative.

- The ε-greedy actions approach could be made more efficient by dynamically changing the ε. For example, one could start with a high ε to explore more at the beginning and gradually decrease it to exploit more later. This way, there is still continuous exploration, but not as much as at the beginning when there was no knowledge of the environment.

From Enes Bilgin, Mastering Reinforcement Learning with Python, Packt 2020

A/B vs. ε-greedy policies

The ε-greedy actions approach could be made more dynamic by increasing the importance of the more recent observations:

$$Q_{n+1}(a) = Q_n(a) + \alpha(R_n(a) - Q_n(a))$$

- **Modifying the ε-greedy actions approach introduces new hyperparameters, which need to be tuned.**
- Both gradually diminishing ε and using exponential smoothing for Q, come with additional hyperparameters, and it may not be obvious what values to set these to.
- Incorrect selection of these hyperparameters may lead to worse results than what the standard version would yield.

From Enes Bilgin, Mastering Reinforcement Learning with Python, Packt 2020

Upper Confidence Bound (UCB)

$$A_t \triangleq \underset{a}{\arg\max} \left[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}} \right]$$

- At each step, we select the action that has the highest potential for reward.

- The potential of the action is calculated as the sum of the action value estimate and a measure of the uncertainty of this estimate. This sum is what we call the upper confidence bound.

- **Overall:** the action is selected either because our estimate for the action value is high, or the action has not been explored enough (i.e. as many times as the other ones) and there is high uncertainty about its value, or both.

From Enes Bilgin, Mastering Reinforcement Learning with Python, Packt 2020

Digging deeper: UCB

$$A_t \triangleq \arg\max_a \left[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}}\right]$$

- $Q_t(a)$ and $N_t(a)$ have the same meanings as before. This formula looks at the variable values, which may have been updated a while ago, at the time of decision making a , whereas the earlier formula described how to update them.
- In this equation, the square root term is a measure of the uncertainty for the estimate of the action value of a .
- The more we select a , the less we are uncertain about its value, and so is the $N_t(a)$ term in the denominator.
- As the time passes, however, the uncertainty grows due to the $\ln t$ term (which makes sense especially if the environment is not stationary), and more exploration is encouraged.
- The emphasis on uncertainty during decision making is controlled by a hyperparameter, c . This obviously requires tuning and a bad selection could diminish the value in the method.

From Enes Bilgin, Mastering Reinforcement Learning with Python, Packt 2020

Advantages and disadvantages of UCB

- **UCB is a set-and-forget approach.** It systematically and dynamically allocates the budget to alternatives that need exploration. If there are changes in the environment, for example, if the reward structure changes because one of the ads gets more popular for some reason, the method will adapt its selection of the actions accordingly.

- **UCB can be further optimized for dynamic environments, potentially at the expense of introducing additional hyperparameters.** The formula we provided for UCB is a common one, but it can be improved, for example, by using exponential smoothing. There are also more effective estimations of the uncertainty component in literature. These modifications, though, could potentially make the method more complicated.

- **UCB could be hard to tune.** It is somewhat easier make the call and say "I want to explore 10% of the time, and exploit for the rest" for the ε-greedy approach than saying "I want my uncertainty to be 0.729" for the UCB approach, especially if you are trying these methods on a brand-new problem. When not tuned, an UCB implementation could give unexpectedly bad results.

From Enes Bilgin, Mastering Reinforcement Learning with Python, Packt 2020

Definitions:

- **Objective:** overall goal that we aim to achieve. Not available during or immediately after experiment.

- **Reward:** the measure of success available at the end of experiment

- **Value:** expected reward. Difference between reward and value is a feedback signal for multiple types of active learning

- **Action:** how can ML agent interact with the system

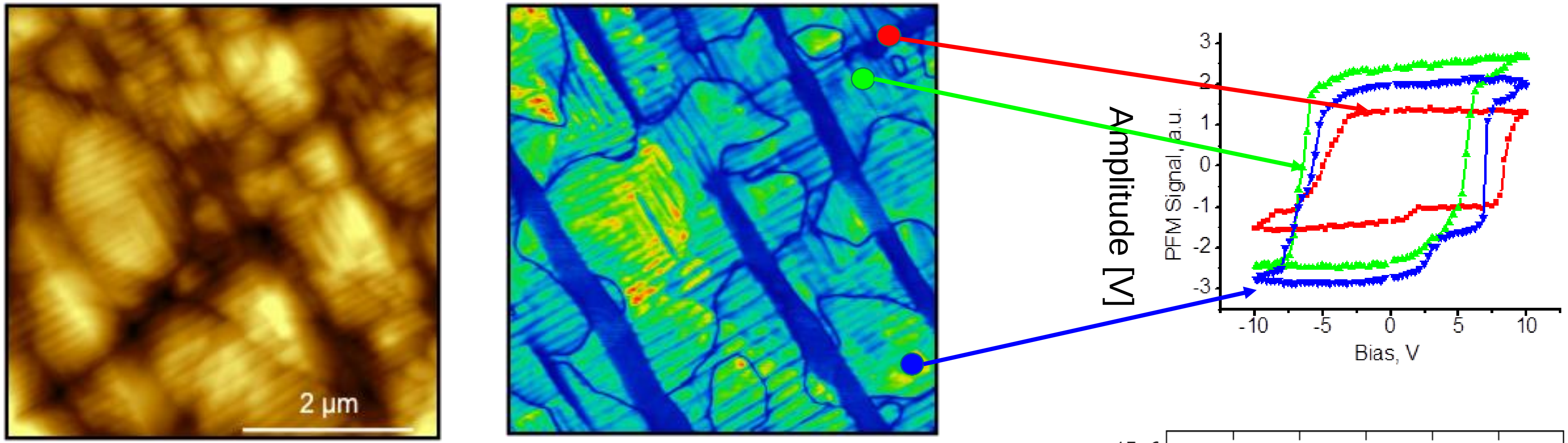
- **State:** information about the system available to ML agent

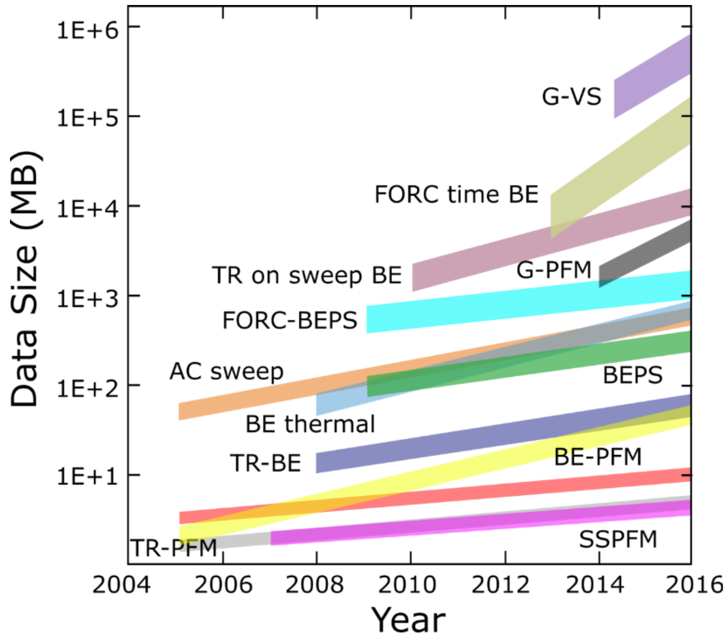
- **Policy:** rulebook that defines actions given the observed state

Bandits

- **Objective:** get rich!
- **Reward:** pay-off from specific hand/click-rate of ad/effectiveness of drug
- **Value:** expected reward
- **Action:** playing a hand/placing ad/administering drug
- **State:** no state
- **Policy:** gameplan given the values of specific actions

Rewards and policies in microscopy world





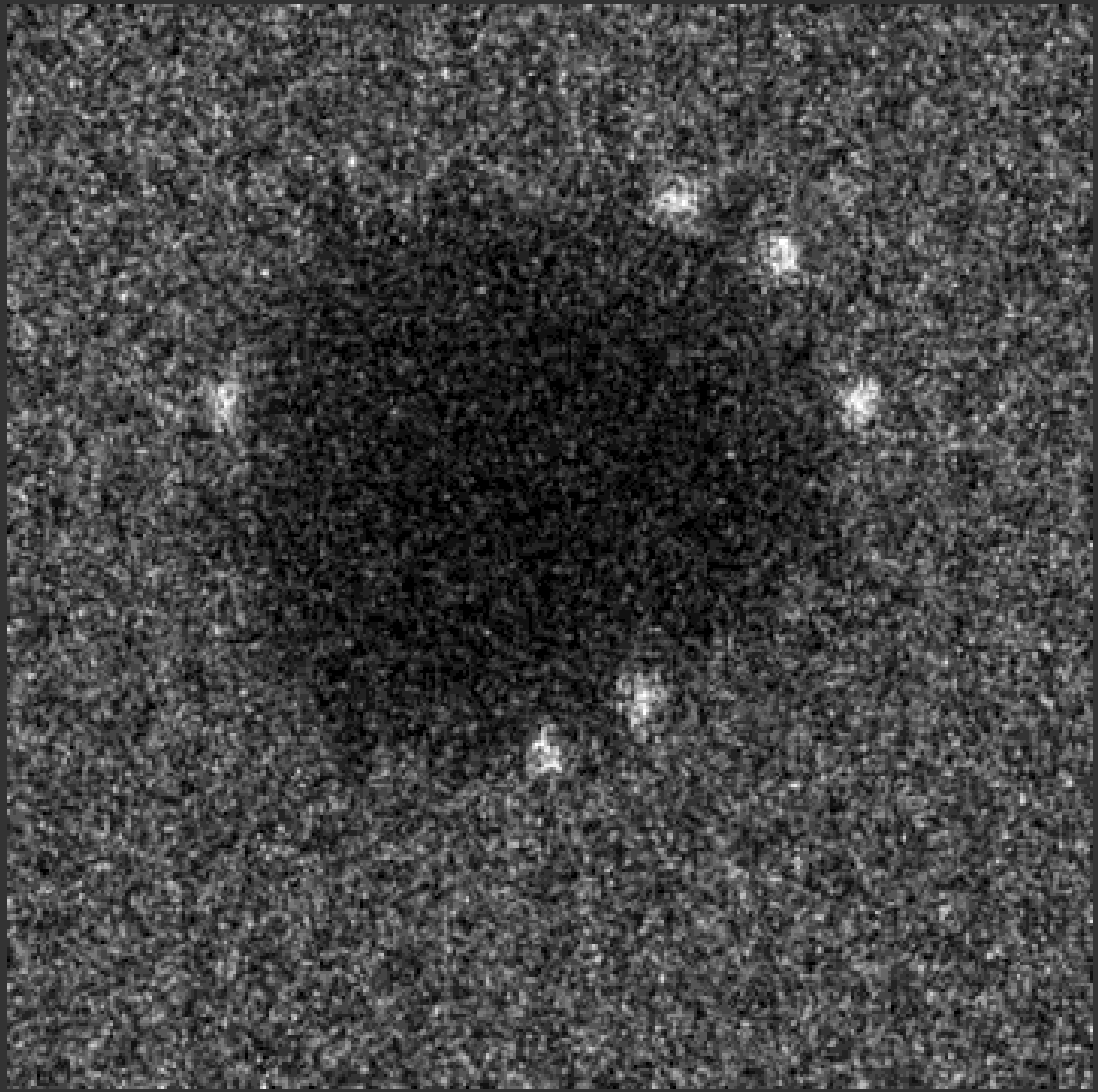
- Interesting functionalities are expected at the certain elements of domain structure
- We can guess some; we have to discover others

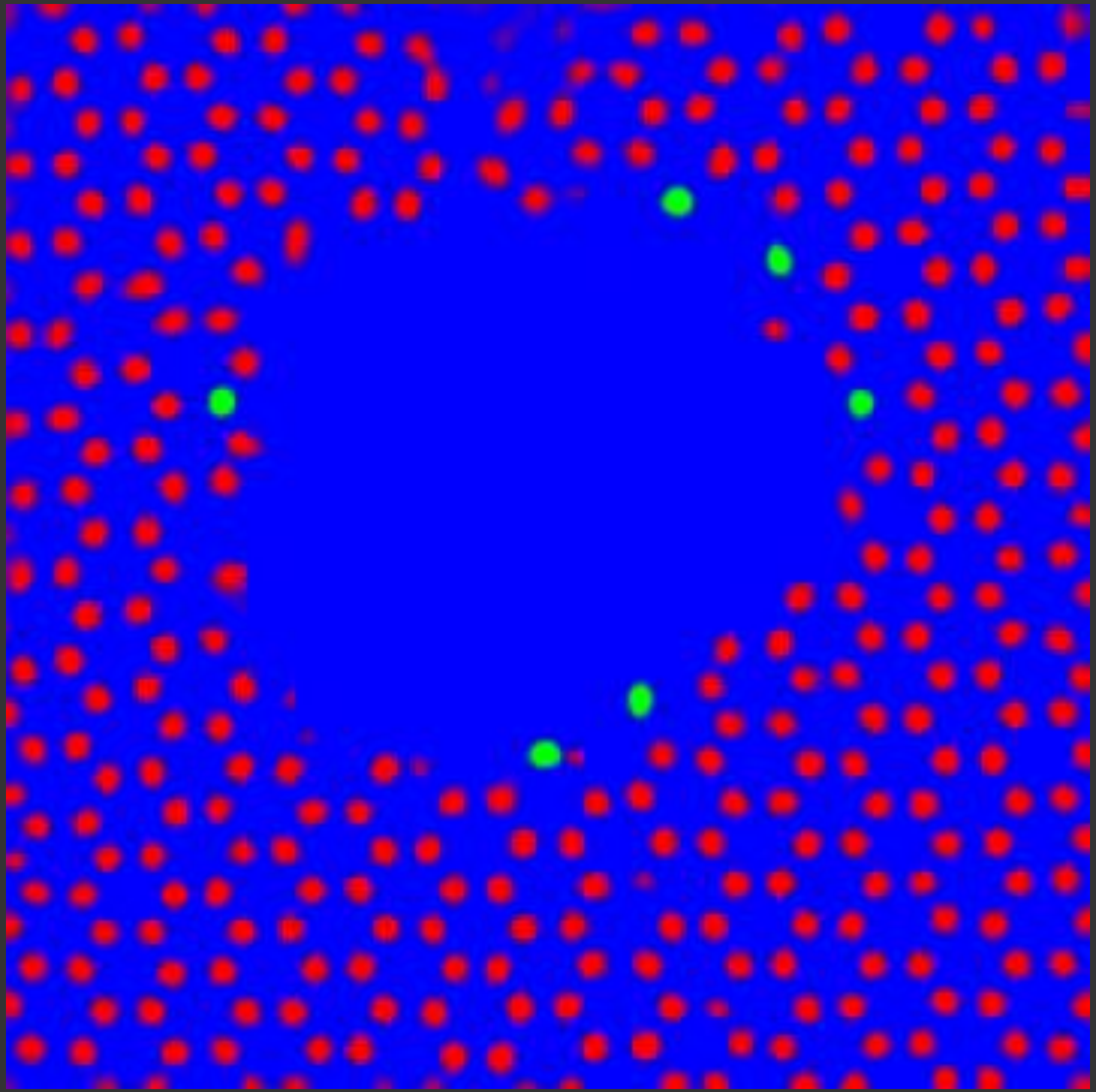
- **Experimental objectives → ML Rewards**
  - Microscope optimization
  - Properties of a priori known regions of interest
  - Discovery of regions with interesting properties
  - Physical theory falsification

Fixed policy experiments

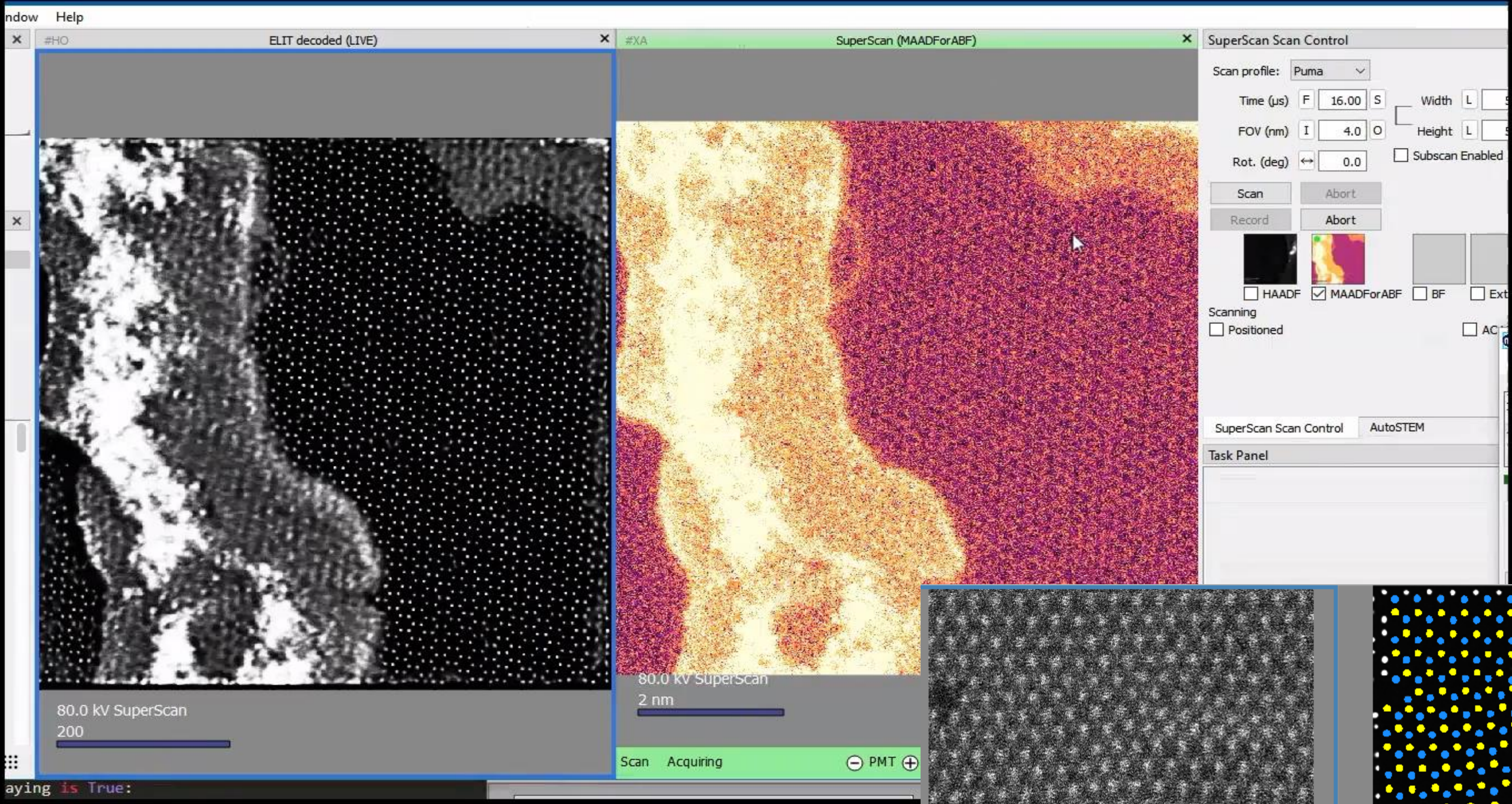
Deep learning works like a charm for:

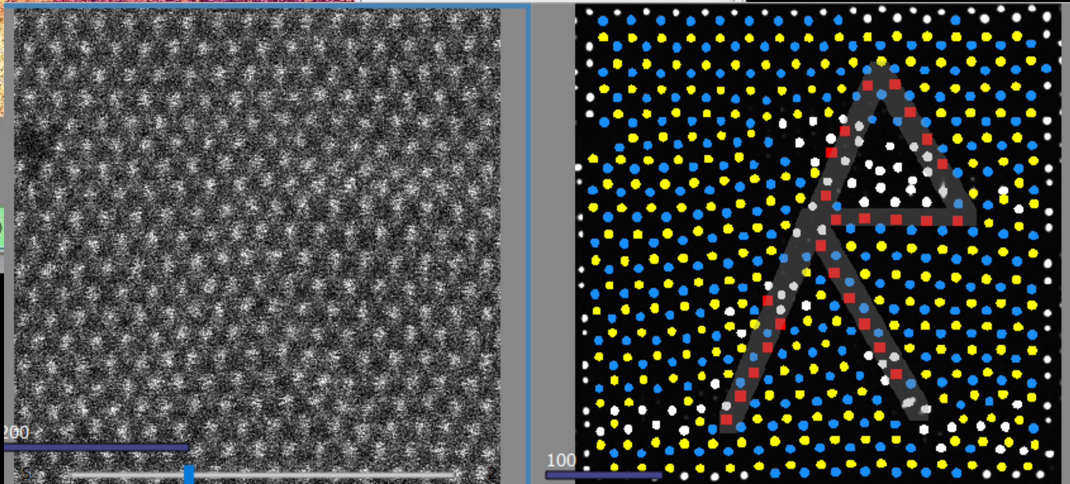
- Drift correction
- Denoising
- Data processing/dimensionality reduction
- Feature finding (physics is in the training set)





AE with Fixed Policies



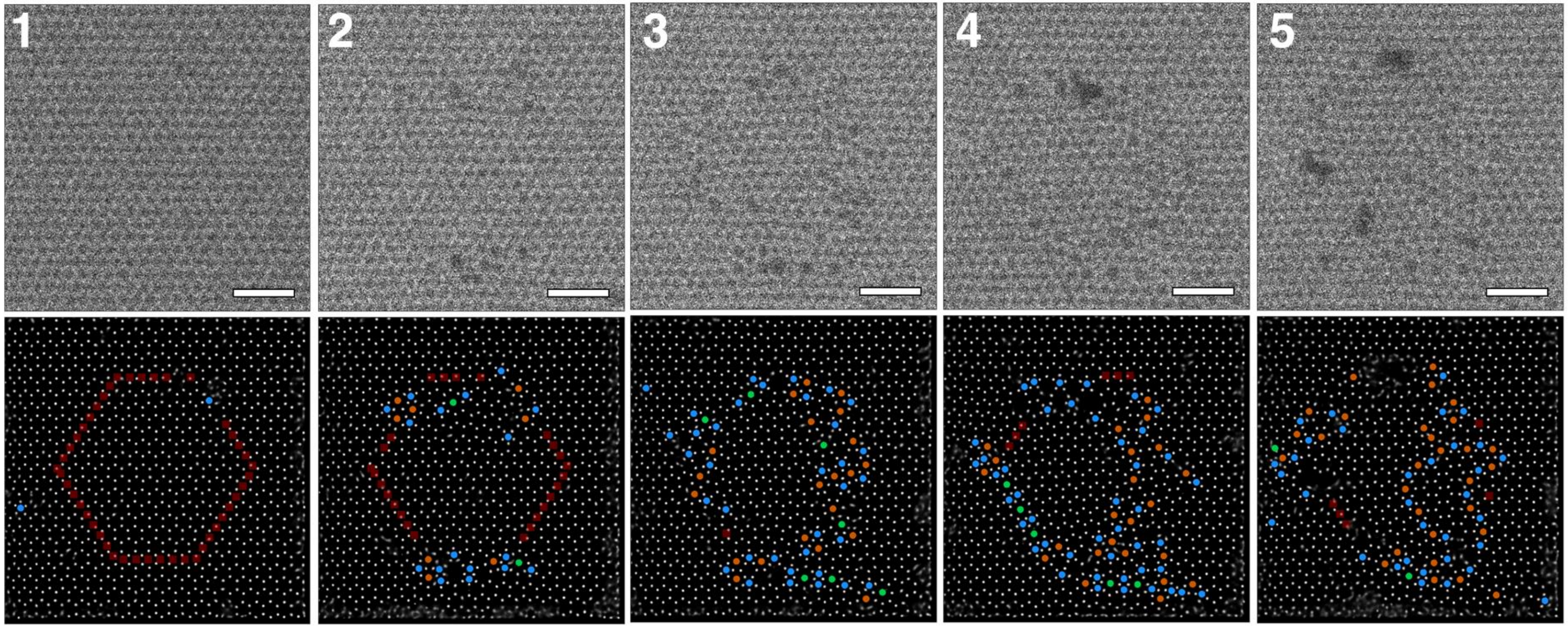


Automated control

Implementation: Kevin Roccapriore, Ayana Ghosh, Sergei V. Kalinin & Maxim Ziatdinov, 2021

Defect patterning in graphene

- Locate atomic coordinates (and defects)
- Direct beam in predetermined path to generate defects (vacancies) – unclear if they form / remain in place
- Repeat scan path, but avoid formed defects
- **Hexagon pattern**



What were we doing?

- **Objective(s):**
  - Understanding electronic and vibrational properties of defects
  - Building structures on the atomic level for biological sequencing and quantum sensing
  - … and so on. We will find out later!

- **Reward:**
  - The number of discovered defects (not used as feedback)
  - Atom moved in desired location

- **Value:** expected reward

- **Action:** position electron beam at given location, take EELS spectrum

- **State:** image

- **Policy:** fixed action table (if detect defect, take EELS)

File Edit View Insert Cell Kernel Widgets Help

Trusted gpuenv

Run Code

```



move_(-volt*2-(offsetvx), 0, 0-offsetvy, 0, move_speed)
```


```
scanning line #56
```

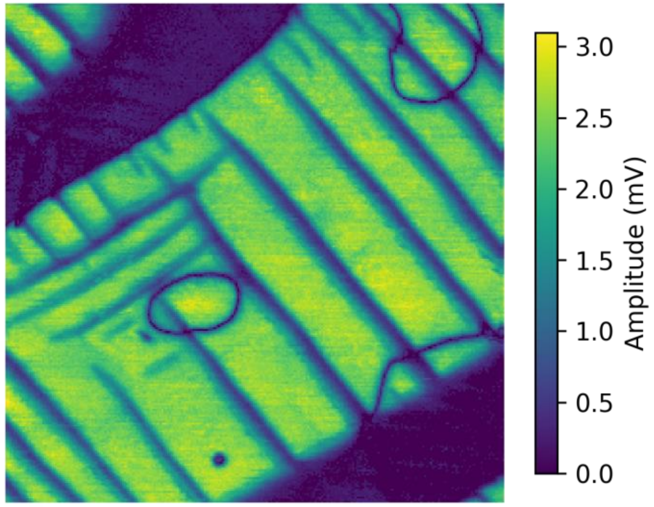
In [ ]:

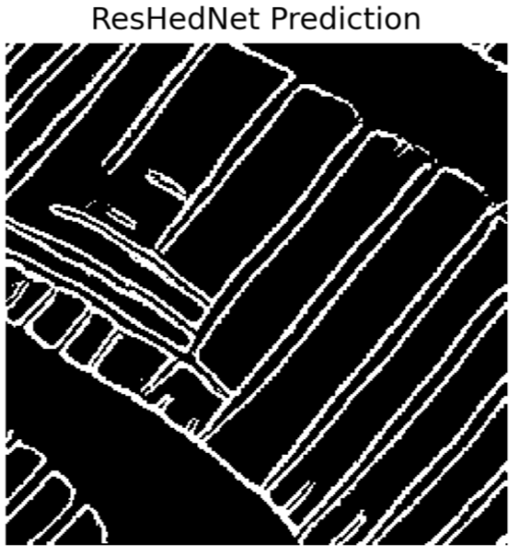
```

```

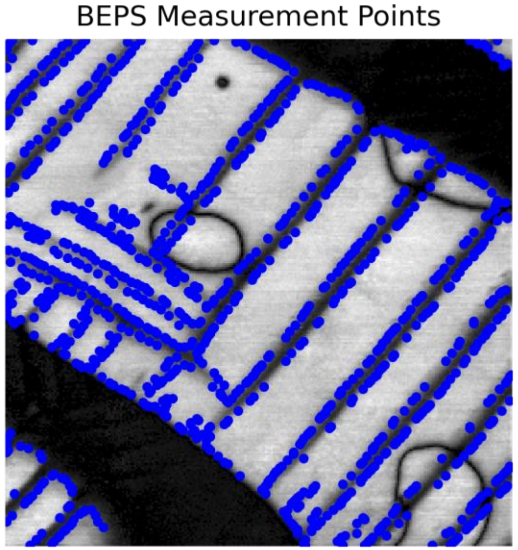
In [ ]:

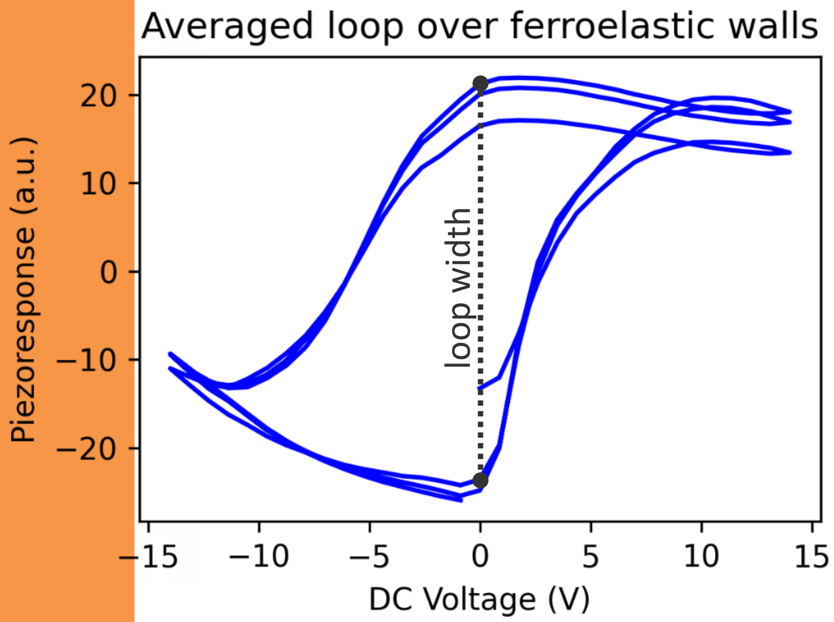
```

```

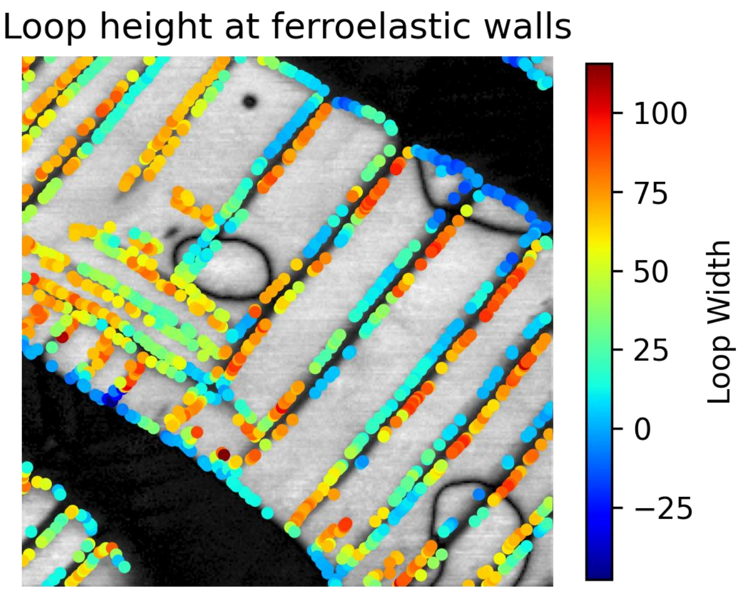
Mapping Activity of Domain Walls

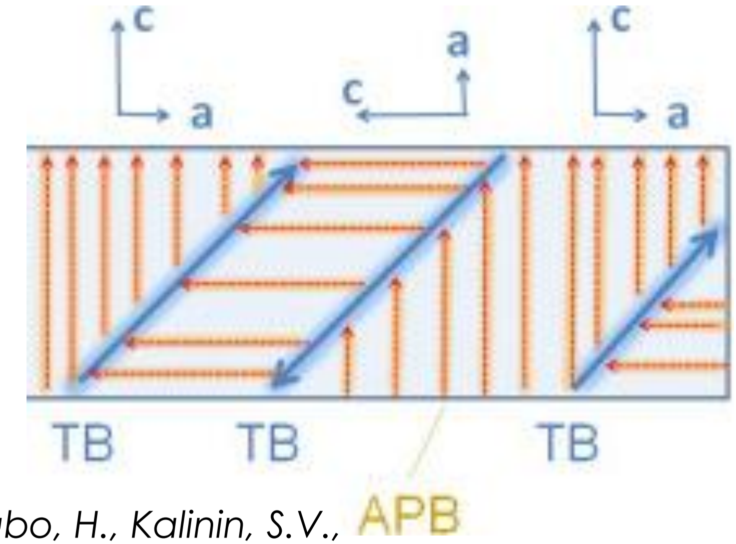












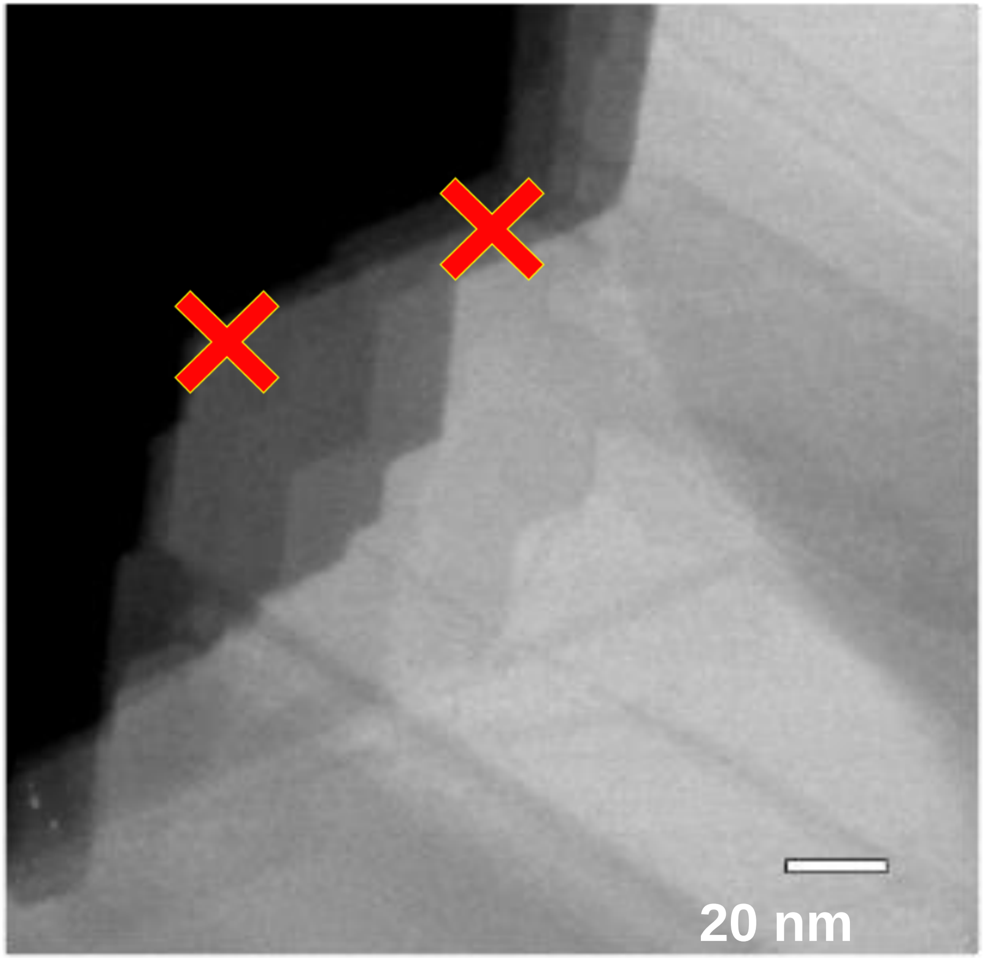
Liu, Y., Kelley, K.P., Funakubo, H., Kalinin, S.V., Ziatdinov, M., Arxiv, submitted

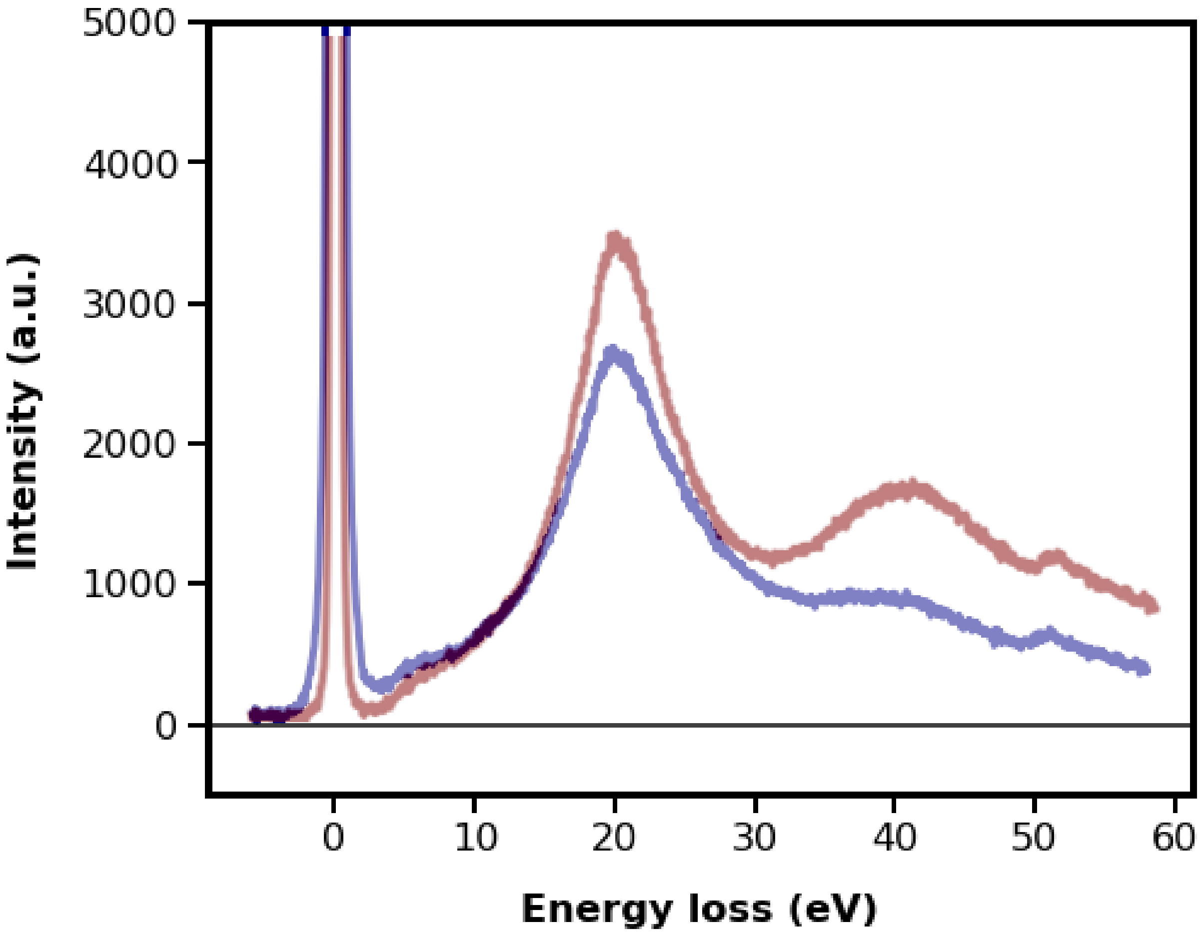
What were we doing?

- **Objective(s):**
  - Understanding the role of domain walls on polarization switching
  - Discover what makes these materials good piezoelectrics
  - … and so on. For fundamental research, vey often impact is clear later!
- **Reward:**
  - The number of explored domain walls (not used as feedback)
- **Value:** expected reward
- **Action:** position SPM probe at a given location, take PFM spectrum
- **State:** image
- **Policy:** fixed action table (if detect wall, take spectrum)

Myopic policy experiments (basically, bandits)

From Static to Active Learning

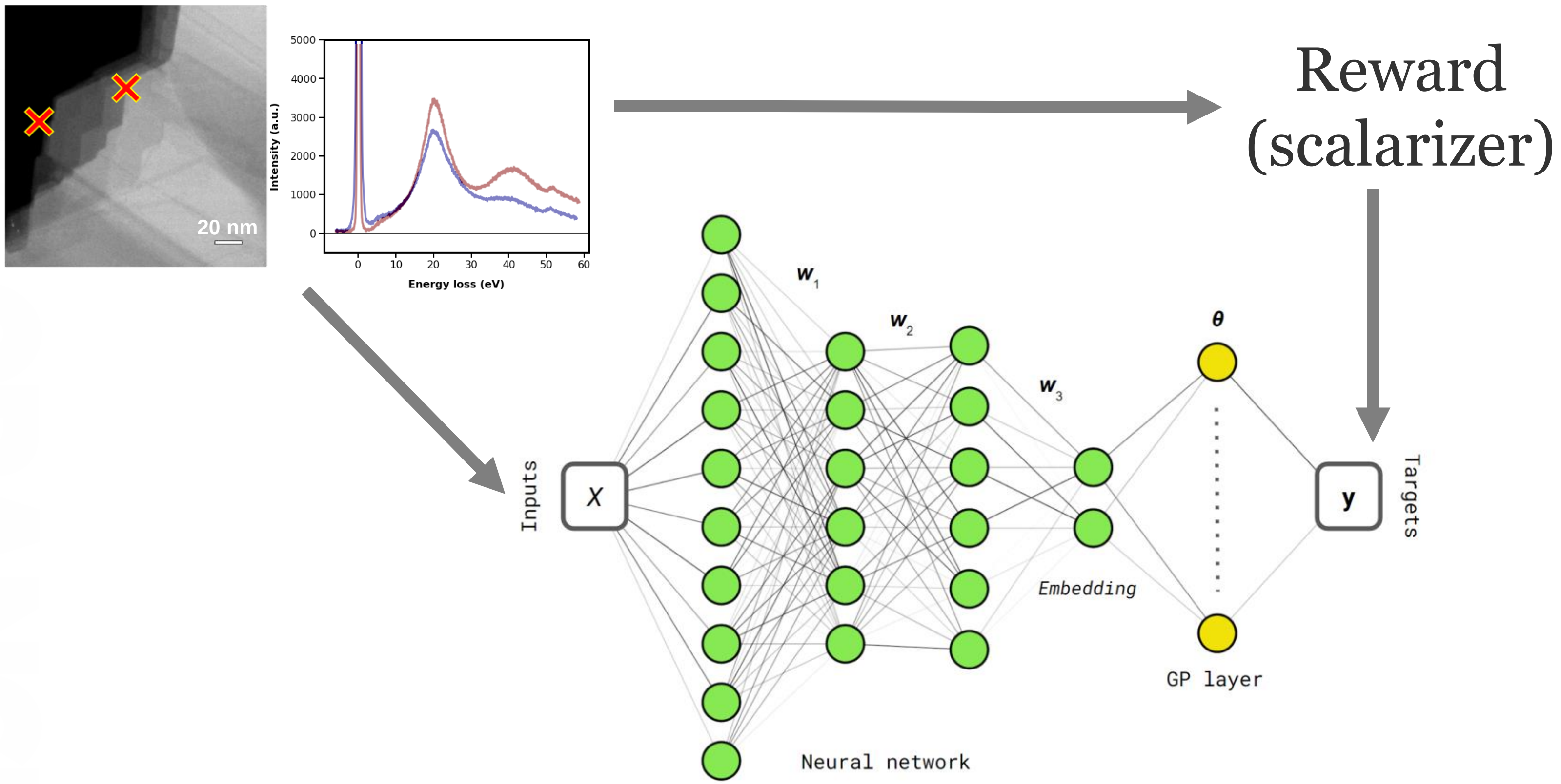




1. What if we have full access to structural information
2. And want to choose locations for (EELS, 4D STEM, CL, EDX) measurements
3. So as to **learn** relationship between structure and spectrum fastest
4. Or **discover** which microstructural elements give rise to specific **desired** spectral features?

Deep Kernel Learning

- All image patches are available in the beginning of the experiment
- We measure spectra one by one
- And are interested in some specific aspect of spectra
- We aim to learn the relationship between structure and this aspect



Allows navigation of the system to search for physics

Specify physics criteria

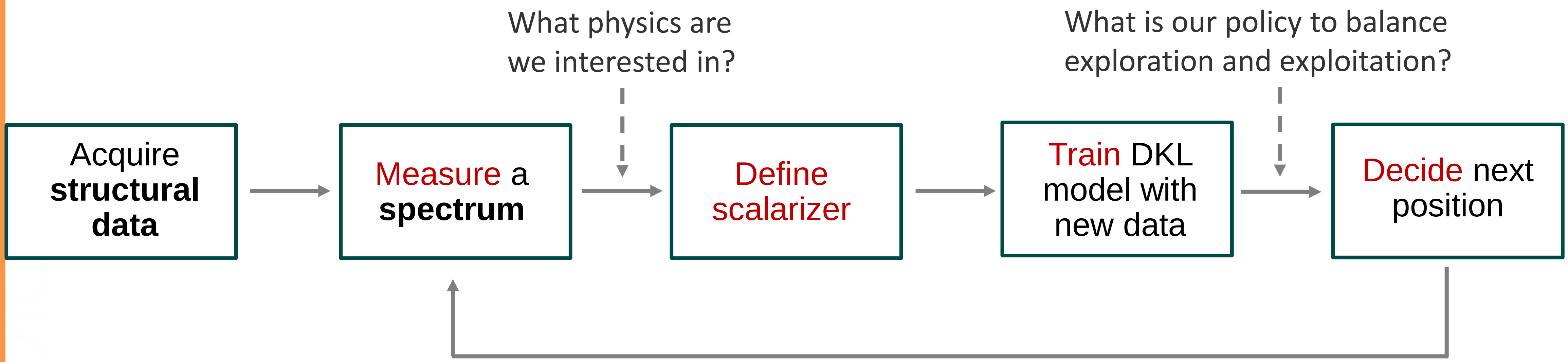
Acquire **structural data**

Measure a **spectrum**

Train DKL model with new data

Decide next position (optimize physics criteria)

Deep Kernel Learning based BO



Key concepts:

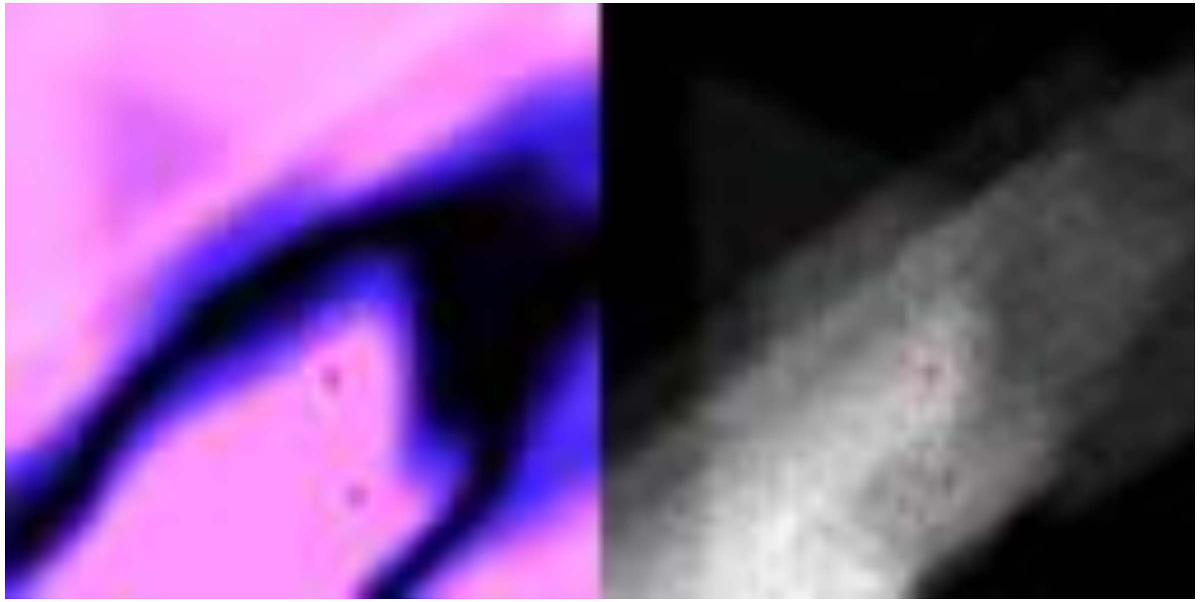
- **Scalarizer:** (any) function that transforms spectrum into measure of interest. Can be integration over interval, parameters of a peak fit, ration of peaks, or more complex analysis

- **Experimental trace:** collection of image patches and associated spectra acquired during experiment. Note that we collect spectra, not only scalarizers

Discovering Regions with Interesting Physics

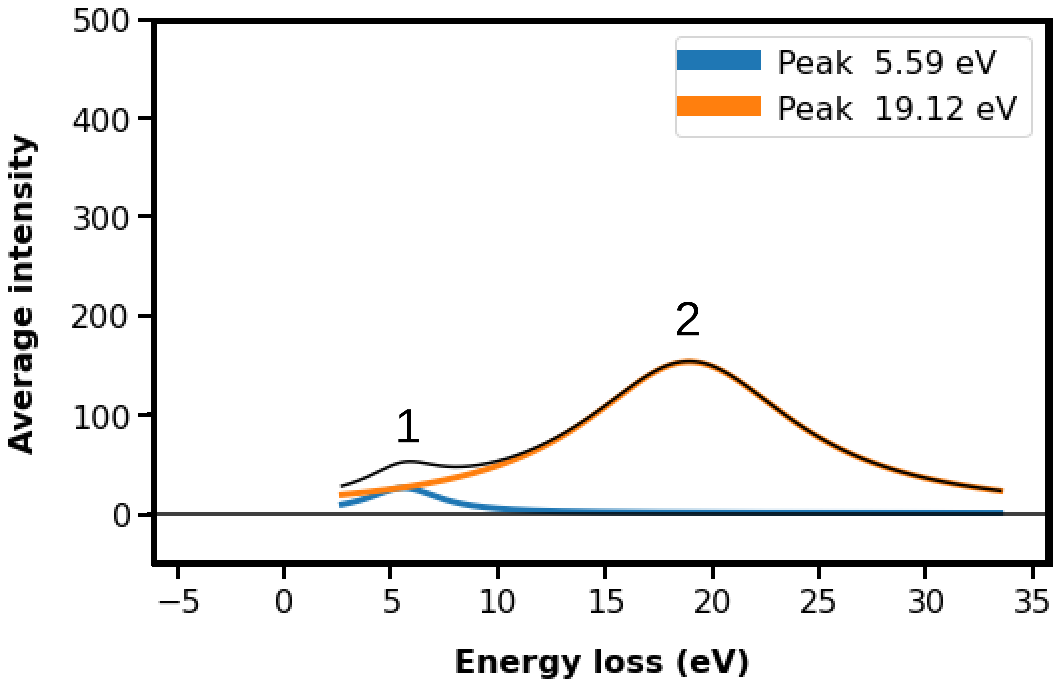
- **Discovering physics in a “new” material $MnPS_3$**
- **Curve fitting** to help enforce physical processes

“Acquisition function” HAADF-STEM



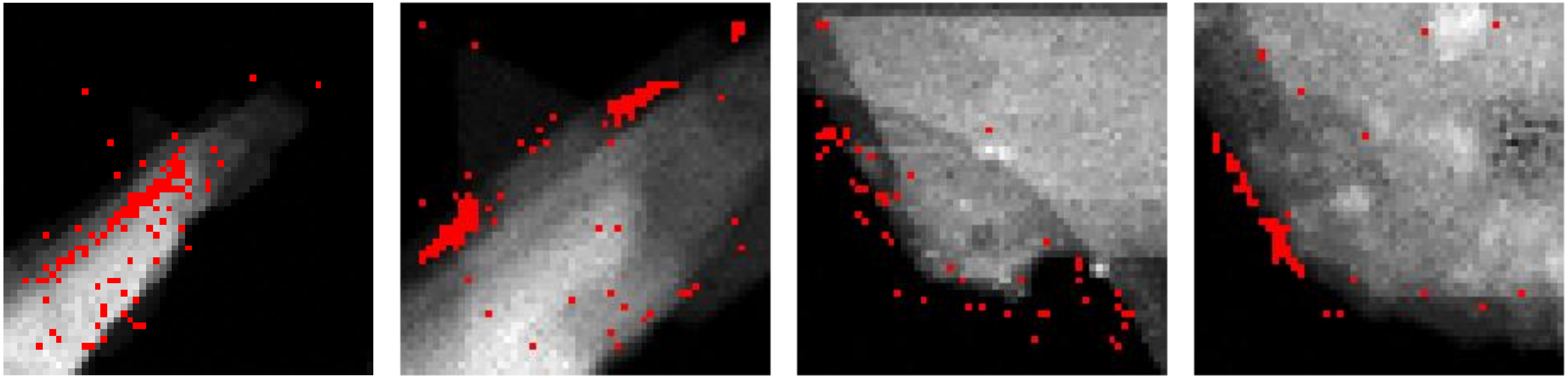
Physics search criteria:

$$Ratio = Peak\,1\,/\,peak\,2$$



More Examples of Physics Discovery

- Very similar behavior when searching for the same criteria!
- Success!

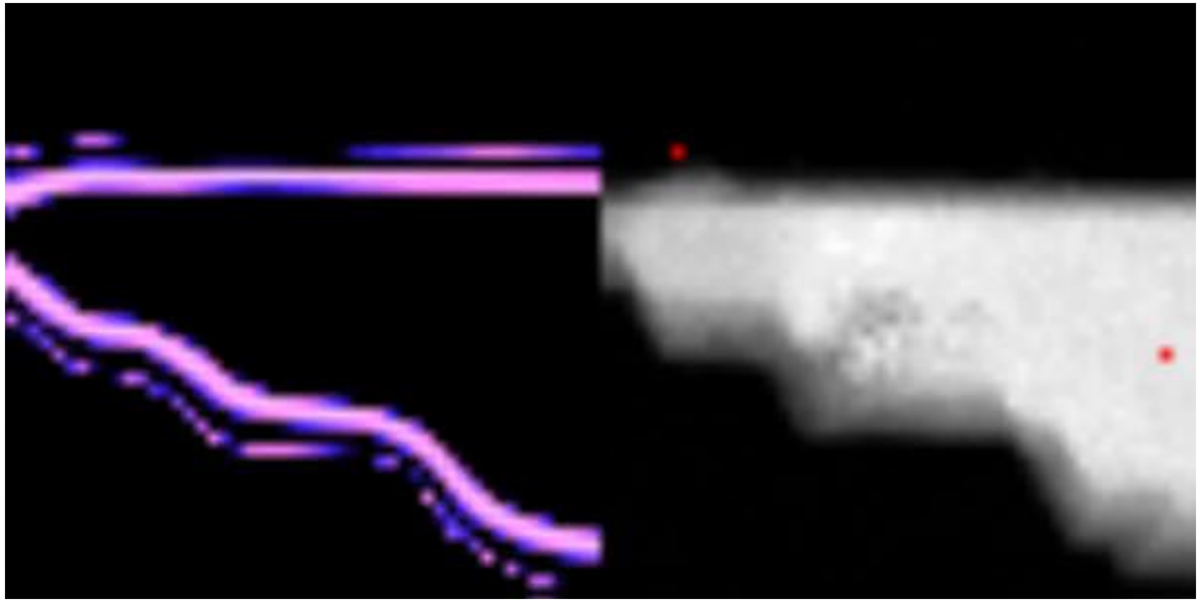


Discovery pathway depends on the reward structure (scalarizer that defines signature of physics we want to discover)!

Changing the scalarizer

- (Same region) **Simple physics search:** peak max in selected region

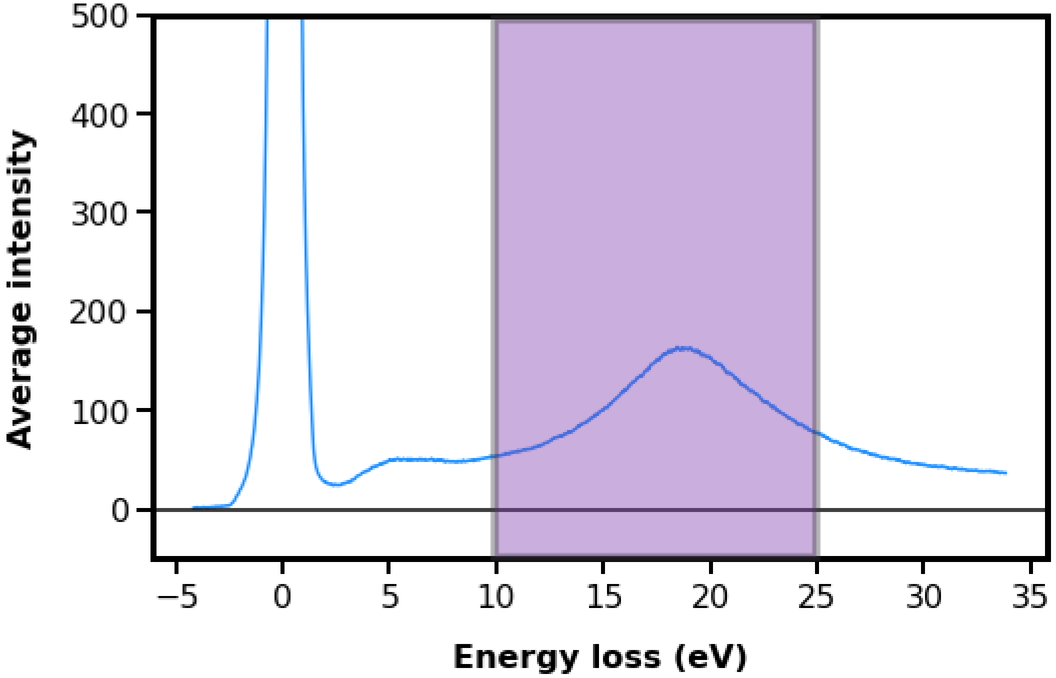
“Acquisition function” HAADF-STEM + points visited



Physics search criteria:

$$\boldsymbol{Maximize(f)}$$

(Specific peak intensity)



What were we doing?

- **Objective(s):**
  - Understanding the emergence of the nanoplasmonic behaviors
  - For understanding physics, optical interfaces to quantum devices, etc.
  - … and so on. For fundamental research, vey often impact is clear later!

- **Reward:**
  - Minimizing uncertainty in structure-property relationships (can we predict expensive EELS from cheap structure)
  - Discovering structures that maximize certain aspect of nanophotonic behavior (have maximal intensity of certain peak, peak area ratio, etc.)

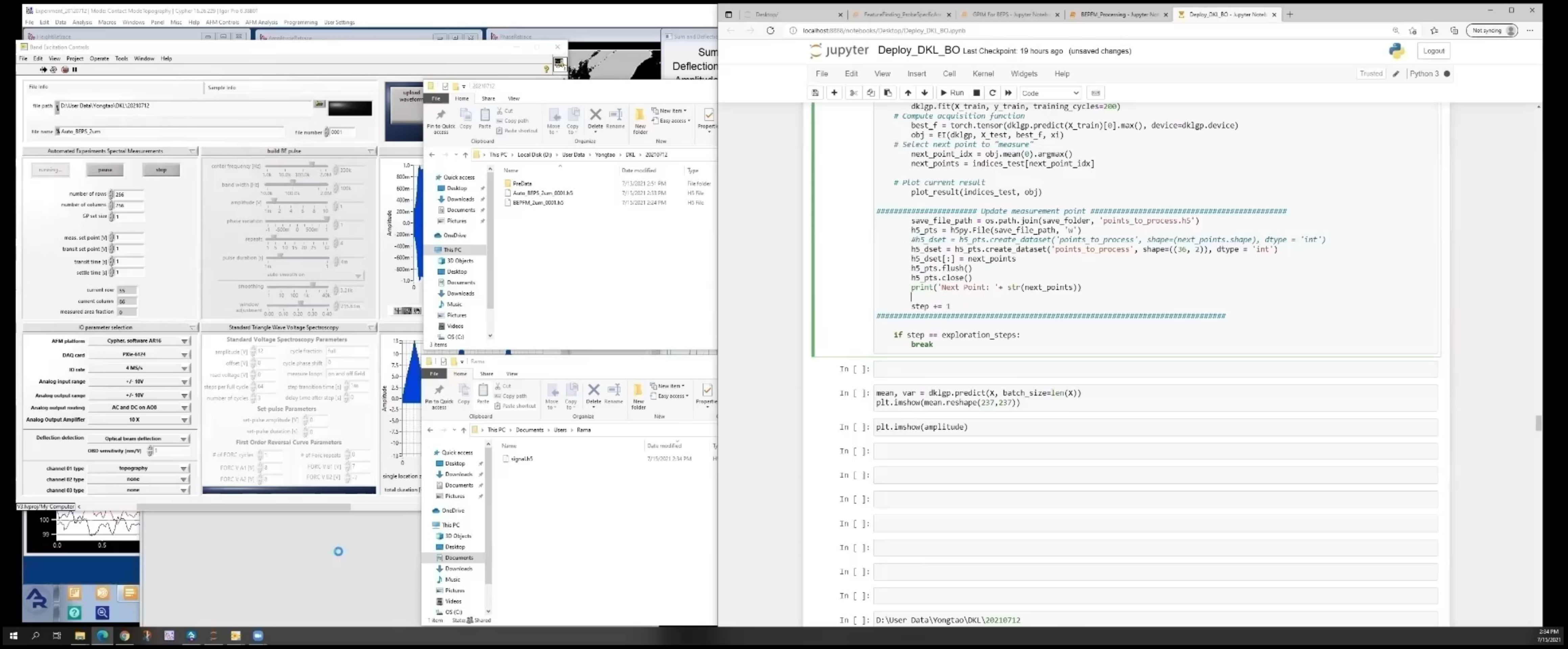
- **Value:** expected reward. Here – predicted scalarizer

- **Action:** position STEM probe at a given location, take EELS spectrum

- **State:** image patch

- **Policy:** myopic optimization (actually, upper confidence bound) with defined exploration-exploitation balance

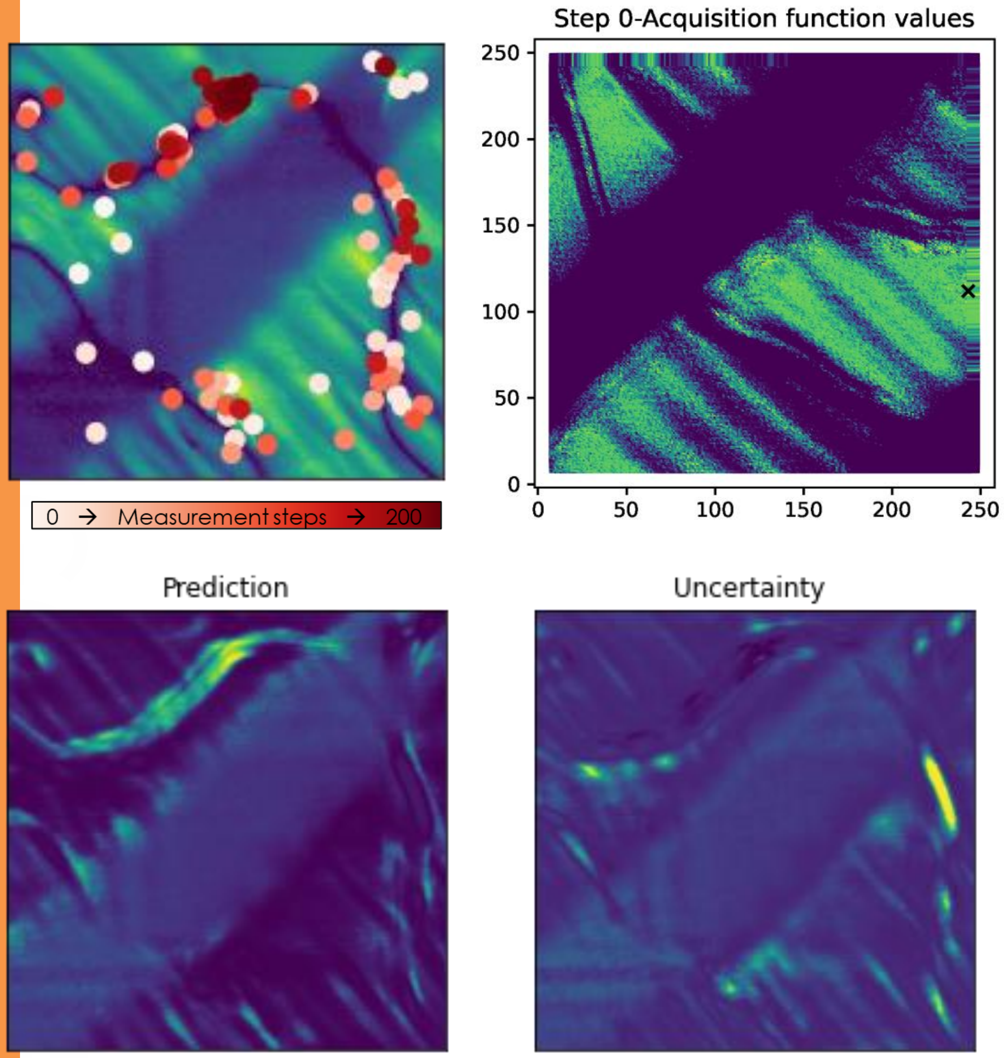
Deep Kernel Learning AE SPM



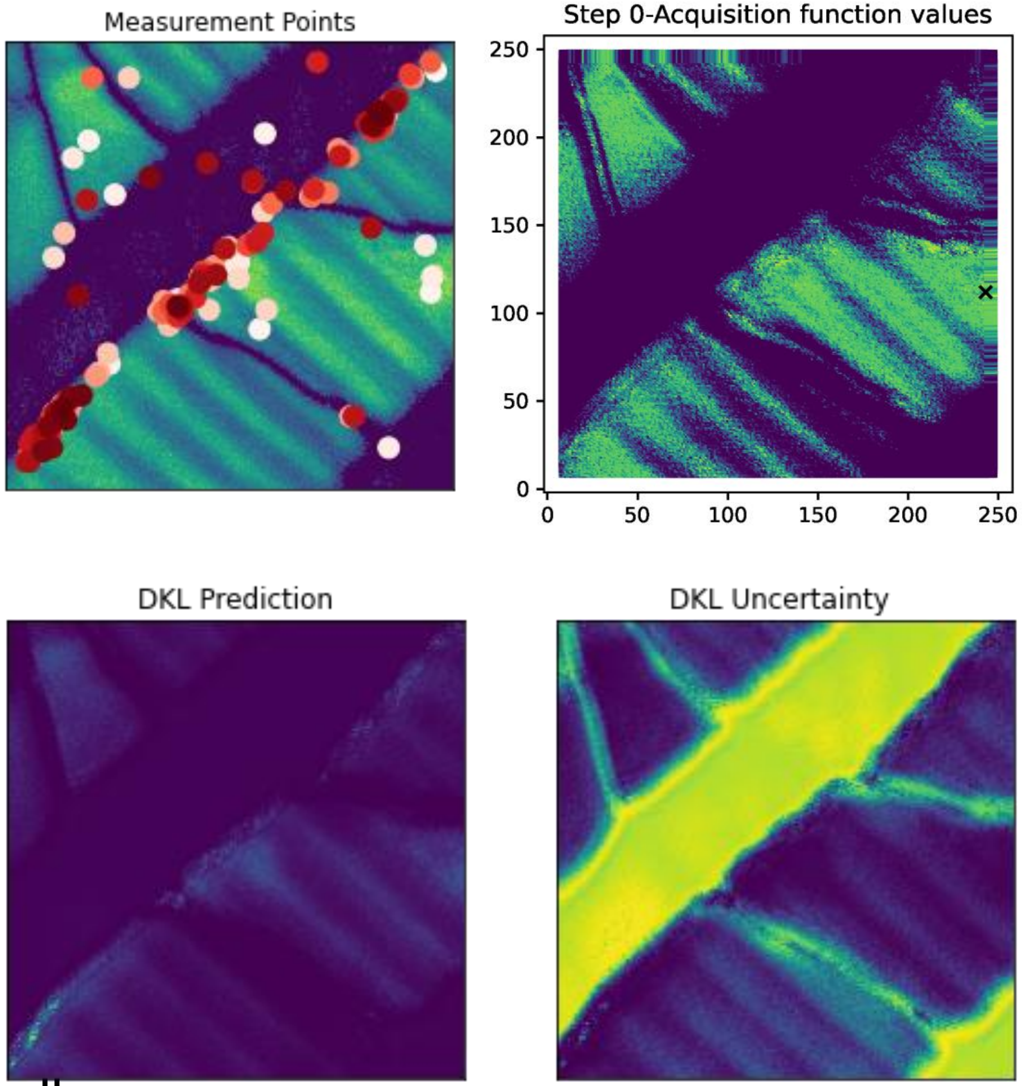
DKL SPM

THE UNIVERSITY of TENNESSEE KNOXVILLE

Guided by: On field loop area



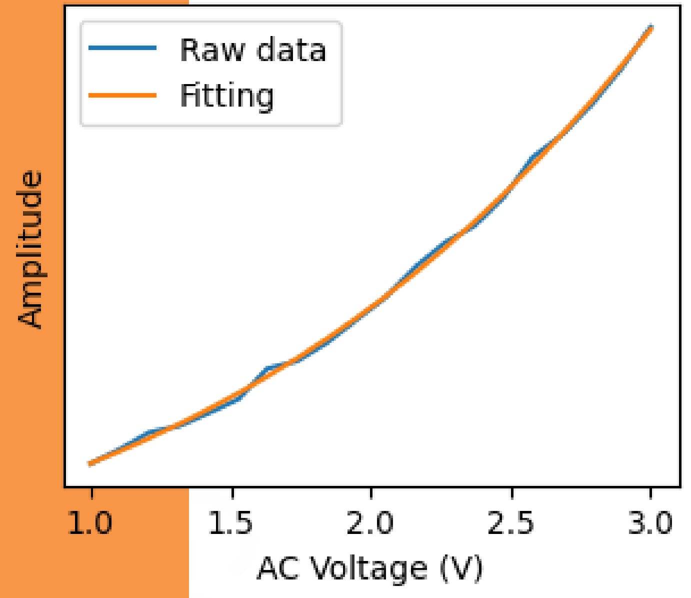
Guided by: Off field loop area



- ➢ Large loop opening corresponding 180º domain walls
- ➢ This behavior can be attributed to the large polarization mobility of 180º walls

Liu, Yongtao, et al, Nature Machine Intelligence 4, 4 (2022): 341-350.

Exploring Non-Linearity



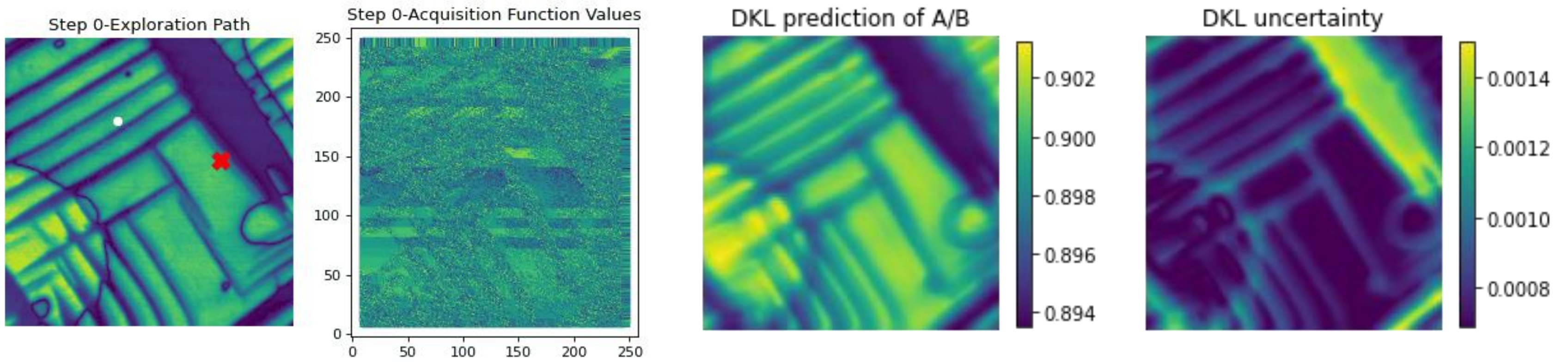
V_{AC} sweep curve at each location was fitted as $y = Ax^3 + Bx^2 + Cx$

A, B, C, and A/B were us as the target function to guide DKL-V_{AC} measurement.

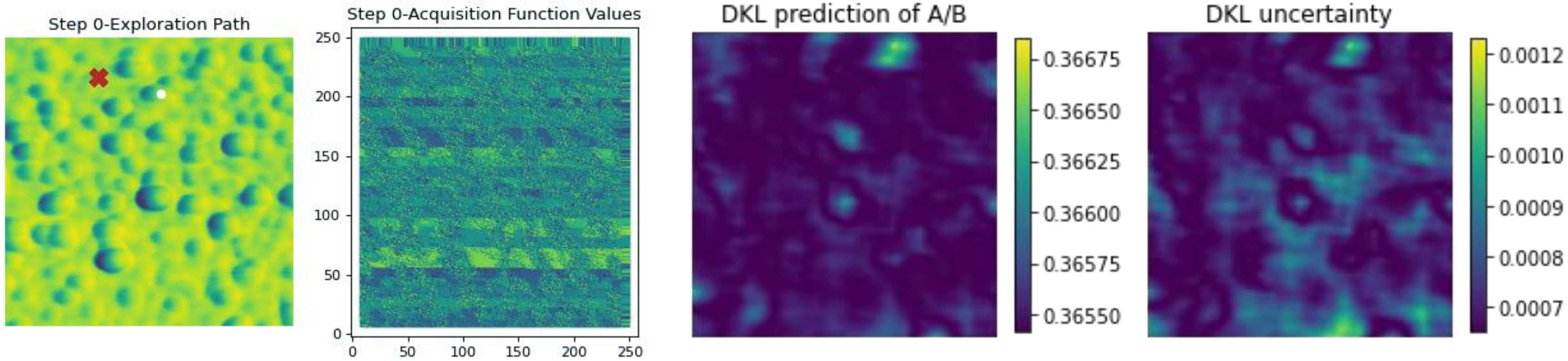
PTO and HZO thin films were studied.

- Shown are 200-step measurements of PTO and HZO thin films
- PFM amplitude was used as structure image; A/B was used to guide the measurement.

PTO experiment process and results



HZO experiment process and results



- In conventional microscopy experiment, human runs everything directly – defines scan, positions the probe, defines measurement parameters.

- In AE SPM, the policies are defined before the experiment and do not change. Sometimes it works – but not always.

- How would we:
  – (a) explain the AE progression after the experiment and
  – (b) control it during the experiment ?

Is there a problem with this type of automated experiment?

Taking the Human Out of the Loop: A Review of Bayesian Optimization

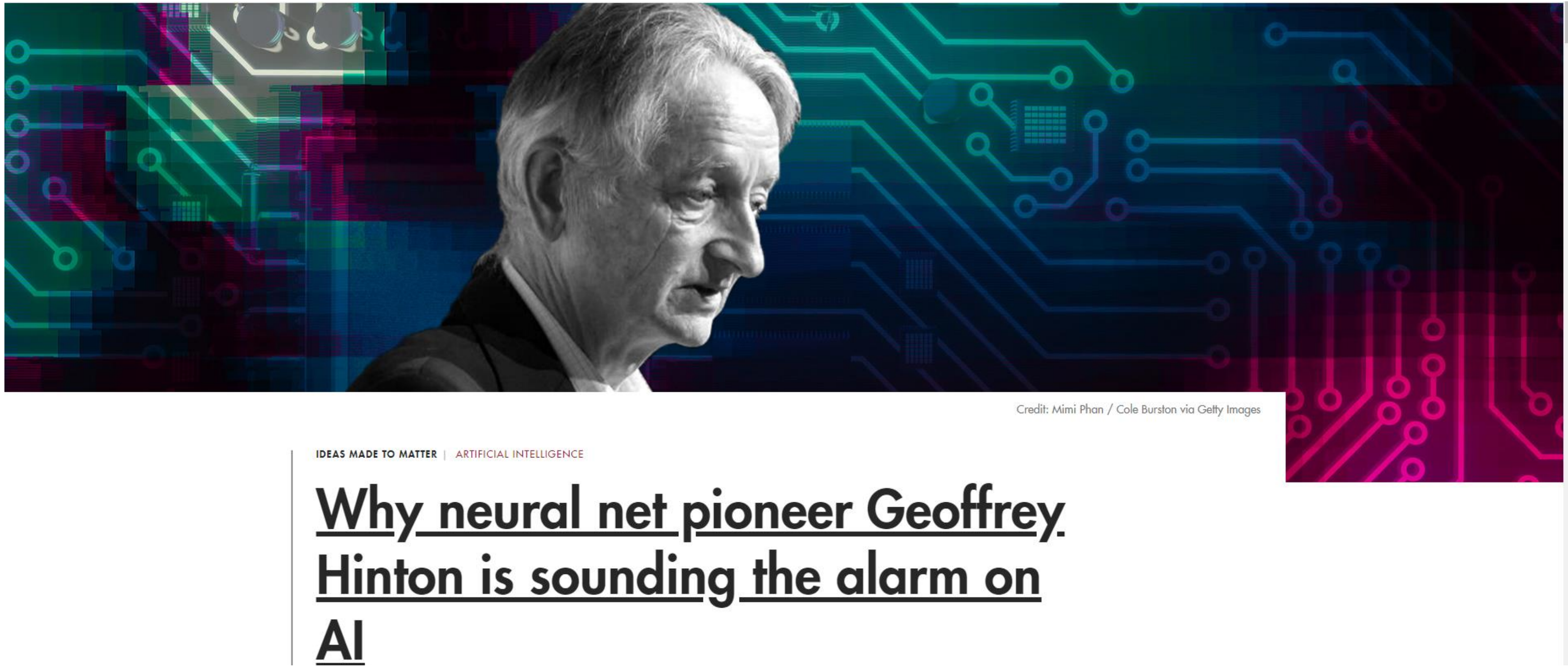
NNESSEE KNOXVILLE

Citation

Shahriari, Bobak, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. 2016. "Taking the Human Out of the Loop: A Review of Bayesian Optimization." Proc. IEEE 104 (1) (January): 148–175. doi:10.1109/jproc.2015.2494218.

Published Version

doi:10.1109/JPROC.2015.2494218



Solution: if we can monitor in real time, we can adjust rewards and policies!

Interactive AI

Human-in-the-loop Automated Experiment

Definitions:

- **Objective:** overall goal that we aim to achieve. Not available during or immediately after experiment.

- **Reward:** the measure of success available at the end of experiment

- **Value:** expected reward. Difference between reward and value is a feedback signal for multiple types of active learning

- **Action:** how can ML agent interact with the system

- **State:** information about the system available to ML agent

- **Policy:** rulebook that defines actions given the observed state

Thompson sampling

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{p(X)}$$

- Initially, we don't have any reason to believe that the parameter is high or low for a given ad. Therefore, it makes sense to assume that $\boldsymbol{\theta_k}$ has a uniform distribution [0,1].

- Assume that we display the ad $\boldsymbol{k}$ and it results in a click. We take this as a signal to update the probability distribution for $\boldsymbol{\theta_k}$ so that the expected value shifts a little bit towards 1.

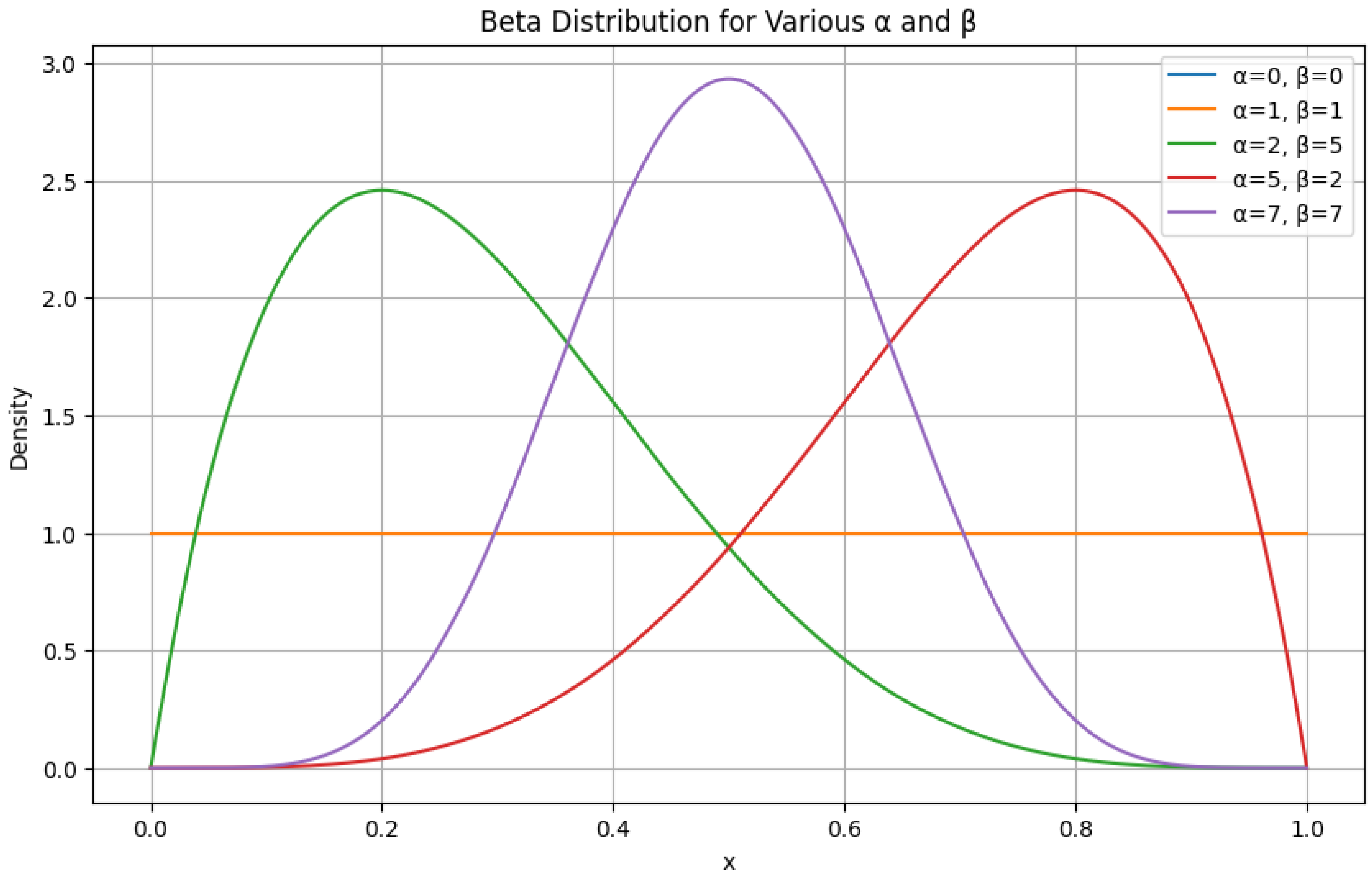
- As we collect more and more data, we should also see the variance estimate for the parameter shrink. Thompson sampling does it using Bayesian inference.

- This method tells us to take a sample from the posterior distribution of the parameter, $\boldsymbol{p(\theta_k|X)}$ If the expected value of $\boldsymbol{\theta_k}$ is high, we are likely to get samples closer to 1. If the variance is high because ad k has not been selected many times that far, our samples will also have high variance, which will lead to exploration.

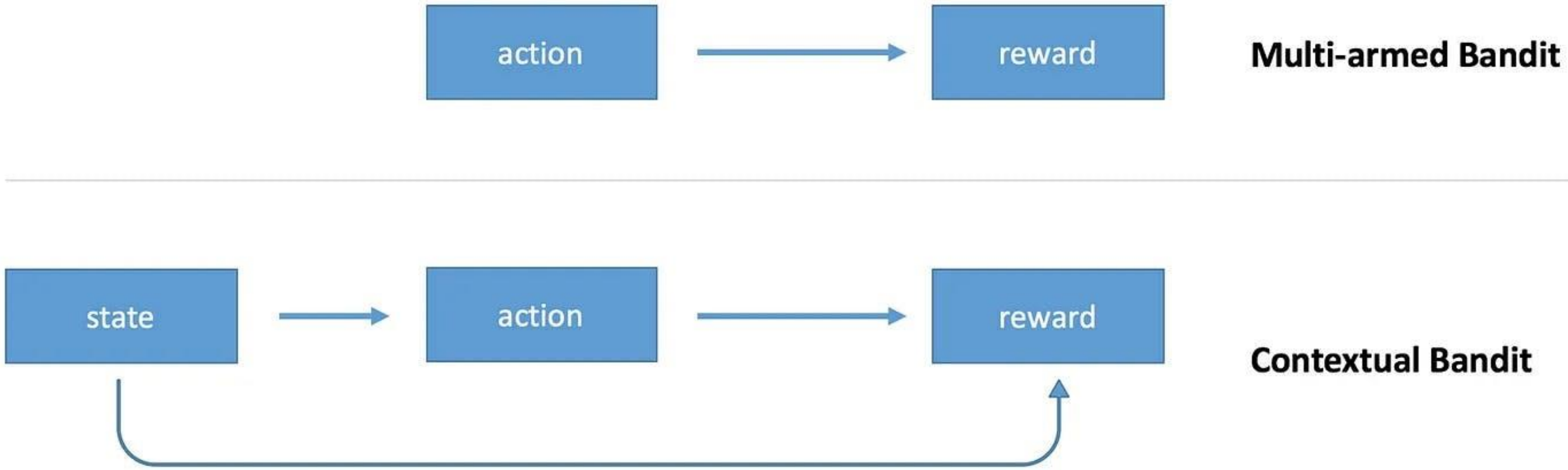
From Enes Bilgin, Mastering Reinforcement Learning with Python, Packt 2020

Thompson sampling: first taste of Bayes

$$p(\theta_k) = \frac{\theta_k^{\alpha_k - 1} \cdot (1 - \theta_k)^{\beta_k - 1}}{B(\alpha_k, \beta_k)}.$$



Contextual Bandits



- In contextual bandits the decision-making process is informed by the context.
- The context, in this case, refers to a set of observable variables that can impact the result of the action.
- This addition makes the bandit problem closer to real-world applications, such as personalized recommendations, clinical trials, or ad placement, where the decision depends on specific circumstances.

https://hackernoon.com/contextual-multi-armed-bandit-problems-in-reinforcement-learning

Simple example

Bandit

A	B	C	D
1.2	20	-9	-1

Contextual Bandit

	A	B	C	D
Shoes	1	-9	8	3
Medicine	3	-10	5	4
Chips	1	6	0.4	-4
Diapers	78	0.9	-0.11	-8

https://medium.com/data-science-in-your-pocket/contextual-bandits-in-reinforcement-learning-explained-with-example-and-codes-3c707142437b

Contextual Bandits

Algorithm 1 Contextual Bandit

Require: Number of actions A
Require: Context feature dimensions d
Require: Context generator
Require: Reward function $\rho(a_t, x_t)$
Require: Learning rate α
Require: Bandit model (e.g., LinUCB, Neural Bandit, Decision Tree Bandit)
  Initialize the bandit model
  for $t = 1$ to T **do**
    Receive the context x_t from the context generator
    Predict the expected reward for each action $\hat{r}(a_t|x_t)$ using the bandit model
    Select the action $a_t = \arg\max_a \hat{r}(a_t|x_t)$
    Receive the reward $r_t = \rho(a_t, x_t)$
    Update the bandit model using the pair (a_t, r_t)
  end for
Ensure: The trained bandit model, The cumulative reward

https://hackernoon.com/contextual-multi-armed-bandit-problems-in-reinforcement-learning

Contextual Bandits

At each time step t, the environment presents a context x_t to the algorithm (often called the agent). Based on this context, the agent chooses an action a_t from a set of possible actions A.

In response to the action, the agent receives a reward r_t. The goal of the agent is to learn a policy π, a mapping from contexts to actions, that maximizes the cumulative reward over time:

$$\max_{\pi} \mathbb{E}\left[\sum_{t=1}^{T} r_t | \pi, D\right] (1)$$

where $D = (x_1, a_1, r_1), ..., (x_T, a_T, r_T)$ is the data (context, action, reward) collected until time T. The expectation is taken over the randomness in the context and rewards.

- The goal of the algorithm is to maximize the cumulative reward over some time horizon.
- In each round, the reward for only the chosen action is observed.
- This is known as partial feedback, which differentiates bandit problems from supervised learning

https://hackernoon.com/contextual-multi-armed-bandit-problems-in-reinforcement-learning

Linear UCB

$$\mathbb{E}[r|x,a] = x^T\theta_a (1)$$

$\mathbb{E}[r|x,a] = x^T\theta_a$ is the expected reward of action a given context x under the linear model
θ_a is the parameter vector for action a.
The objective is to learn θ_a for all actions based on the data.

$\Rightarrow$ At each time step t, given context x_t, LinUCB selects the action a_t that has the maximum UCB:

$$a_t = \arg\max_{a \in A} \left(x_t^T \theta_a + \alpha \sqrt{x_t^T A_a^{-1} x_t} \right)$$

A_a is a matrix that keeps track of the features seen so far for action a,
α is a parameter controlling the trade-off between exploration and exploitation,
the second term is the UCB, which is proportional to the standard deviation of the estimated reward.

Linear UCB Update

Chosen action:

$$a_t = \arg\max_{a \in A} \left(x_t^T \theta_a + \alpha \sqrt{x_t^T A_a^{-1} x_t} \right)$$

Model updates:

$$A_a = A_a + x_t x_t^T \qquad b_a = b_a + r_t x_t$$

$$\theta_a = A_a^{-1} b_a$$

- LinUCB algorithm is a contextual bandit algorithm that models the expected reward of an action given a context as a linear function

- LinUCB it selects actions based on the Upper Confidence Bound (UCB) principle to balance exploration and exploitation.

- It exploits the best option available according to the linear, but it also explores options that could potentially provide higher rewards, considering the uncertainty in the model's estimates.

Decision Tree cMAB

Formally, a decision tree T defines a partition of the context space X into K disjoint subsets $X_1, X_2, ..., X_K$, each associated with a recommended action a_k.
The reward of action a_k in context x under the decision tree T is:

$$\mathbb{E}[r|x, a_k, T] = \mathbb{E}[r|x \in X_k]$$

$\Rightarrow$ Given a context x, the decision tree selects the action associated with the subset that x falls into.

The objective is to learn the best decision tree that maximizes the expected cumulative reward:

$$\max_T \mathbb{E}\left[\sum_{t=1}^{T} r_t | T, D\right]$$

Where
$D = (x_1, a_1, r_1), ..., (x_T, a_T, r_T)$ is the data (context, action, reward) collected until time T.

Decision Tree Bandit models the reward function as a decision tree, where each leaf node corresponds to an action and each path from the root to a leaf node represents a decision rule based on the context. It performs exploration and exploitation through a statistical framework, making splits and merges in the decision tree based on statistical significance tests.

Neural Network cMAB

- Deep learning models can be used to approximate the reward function in high-dimensional or non-linear cases. The policy is typically a neural network that takes the context and available actions as input and outputs the probabilities of taking each action.

- A popular deep learning approach is to use an actor-critic architecture, where one network (the actor) decides which action to take, and the other network (the critic) evaluates the action taken by the actor.

- For more complex scenarios where the relationship between the context and the reward is not linear, we can use a neural network to model the reward function. One popular method is to use a policy gradient method, such as REINFORCE or actor-critic.

Neural Network cMAB

Let $\pi_\theta(a|x)$ be the policy parameterized by θ, i.e., the probability of selecting action a given context x. The objective is to find θ that maximizes the expected cumulative reward:

$$\max_\theta \mathbb{E}\left[\sum_{t=1}^{T} r_t | \pi_\theta, D\right]$$

$\Rightarrow$ Using the policy gradient theorem, we can update θ iteratively:

$$\theta_{t+1} = \theta_t + \alpha(r_t - b(x_t))\nabla \log \pi_\theta(a_t|x_t)$$

where α is the learning rate, $b(x_t)$ is a baseline function (often the average reward), and the expectation is approximated by sampling actions according to π_θ.

- Neural Bandit uses neural networks to model the reward function, taking into account the uncertainty in the parameters of the neural network.

- It further introduces exploration through policy gradients where it updates the policy in the direction of higher rewards.

- This form of exploration is more directed, which can be beneficial in large action spaces.

LinUCB: The Linear Upper Confidence Bound (LinUCB) model uses linear regression to estimate the expected reward for each action given a context. It also keeps track of the uncertainty of these estimates and uses it to encourage exploration.

Advantages:
• It is simple and computationally efficient.
• It provides theoretical guarantees for its regret bound.

Disadvantages:
• It assumes the reward function is a linear function of the context and action, which may not hold for more complex problems.

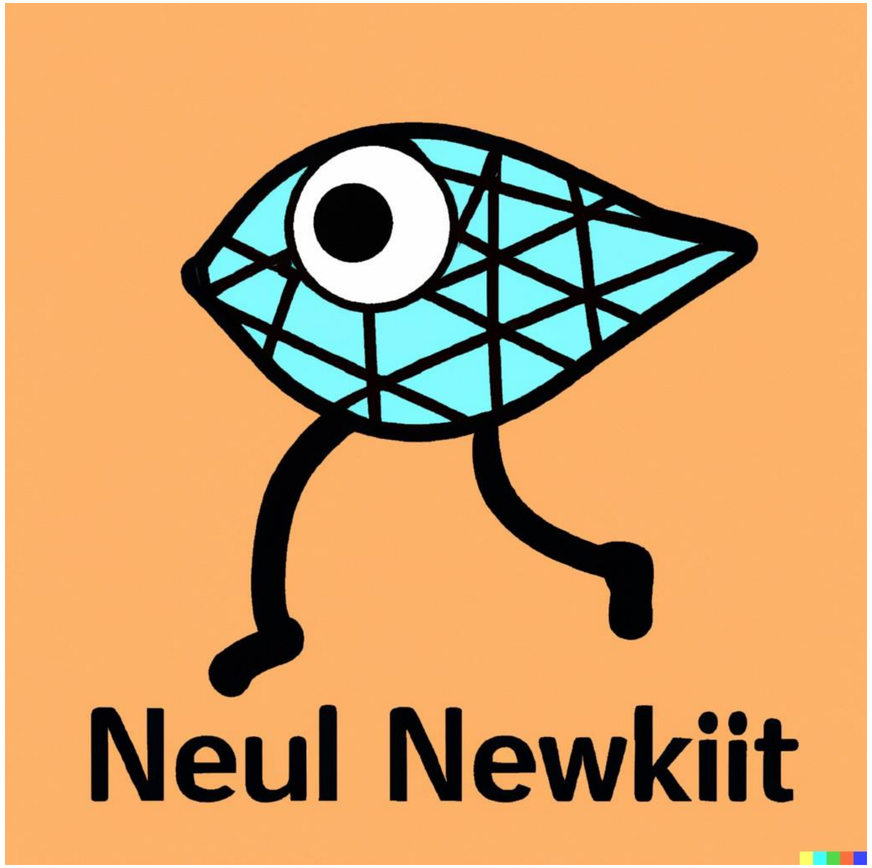
Decision Tree Bandit: The Decision Tree Bandit model represents the reward function as a decision tree. Each leaf node corresponds to an action, and each path from the root to a leaf node represents a decision rule based on the context.

Advantages:

- It provides interpretable decision rules.
- It can handle complex reward functions.

Disadvantages:

- It may suffer from overfitting, especially for large trees.
- It requires tuning hyperparameters such as the maximum depth of the tree.



Neural Bandit: The Neural Bandit model uses a neural network to estimate the expected reward for each action given a context. It uses policy gradient methods to encourage exploration.

Advantages:

- It can handle complex, non-linear reward functions.
- It can perform directed exploration, which is beneficial in large action spaces.

Disadvantages:

- It requires tuning hyperparameters such as the architecture and learning rate of the neural network.
- It may be computationally expensive, especially for large networks and large action spaces.

- Should experiments be reproducible?
- Where do the error bars come from?
- How do we know the hypothesis is proven/rejected?

Basic frequentist approach

Views probability as the long-run frequency of events. For example, the probability of getting a 'heads' in a coin toss is interpreted as the long-run frequency of getting 'heads' over many trials.

- **Mean (Expected Value)**:
  - Represents the central tendency of a dataset.
  - It is the sum of all observations divided by the number of observations.

- **Dispersion**:
  - Measures the spread or variability in a dataset.
  - Common metrics include variance (the average squared deviation from the mean) and standard deviation (the square root of variance)

- **Error Bars**:
  - Graphical representation of the variability of data.
  - Commonly used error bars include standard deviation, standard error, and confidence intervals.
  - Help indicate the reliability of a point estimate.
  - Wider error bars typically suggest more variability or uncertainty in the data.

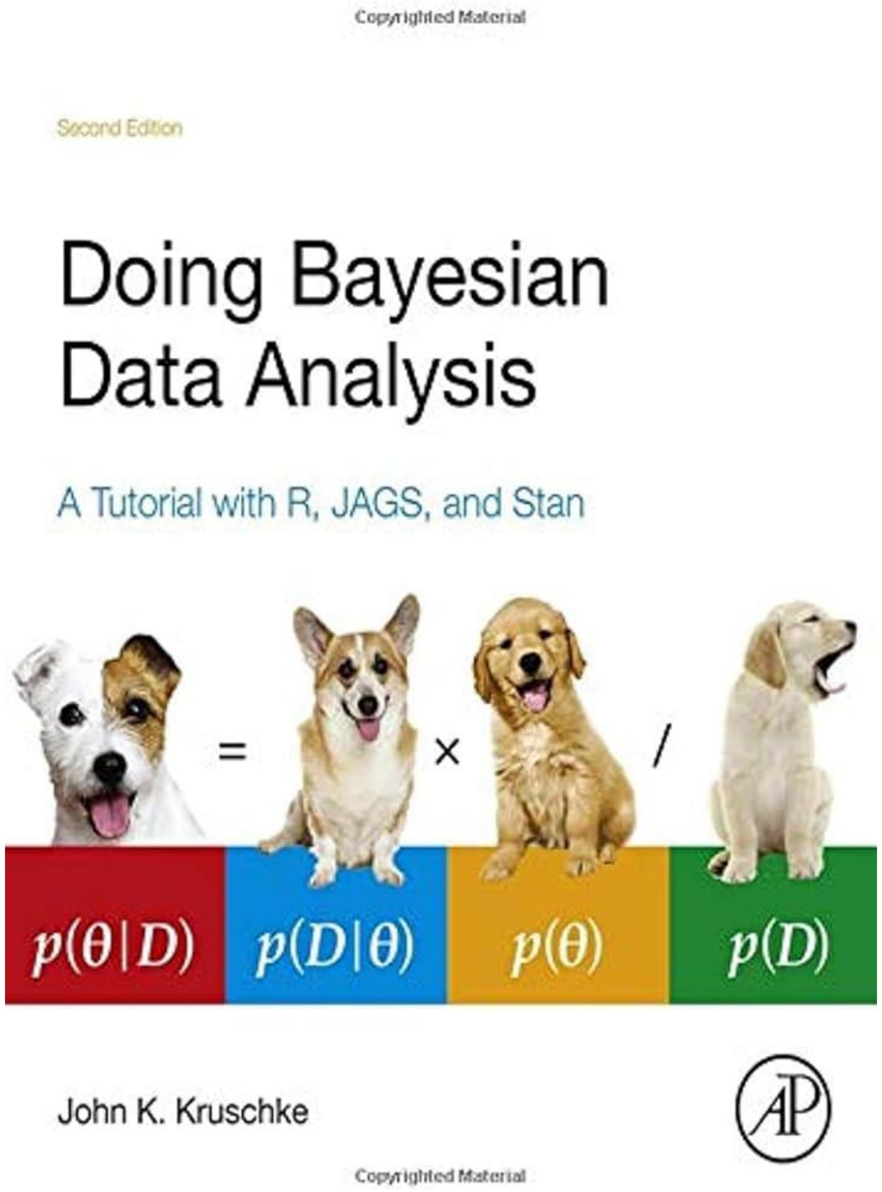
Basic frequentist approach

- **Null Hypothesis (H0)**:
  - A statement or assumption that there is no effect or no difference, serving as a default or starting point
  - It is the hypothesis that researchers typically aim to test against using statistical tests

- **Alternative Hypothesis (H1 or Ha)**:
  - A statement indicating the presence of an effect or difference

- **P-values**:
  - A p-value measures the strength of the evidence against the null hypothesis.
  - It represents the probability of observing data (or something more extreme) given that the null hypothesis is true.
  - A small p-value (typically < 0.05) suggests that the observed data is inconsistent with the null hypothesis, leading researchers to reject H0 in favor of Ha.

If feels non-straightforward, you are not alone...



Setting of the problem

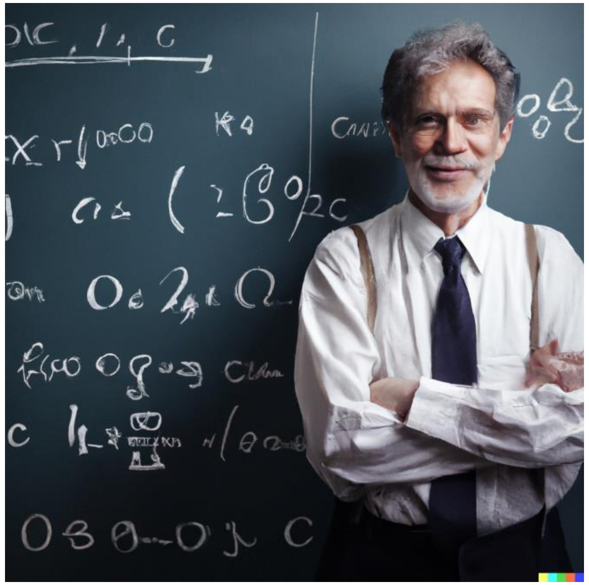
- Imagine that you are tossing a coin with your friend to decide who goes for groceries. What is the probability that the coin will land tails?

- Imagine that previously you have tossed this coin several times, and it landed tails 3 times in a row.

- … what if it was 100 times in a row?

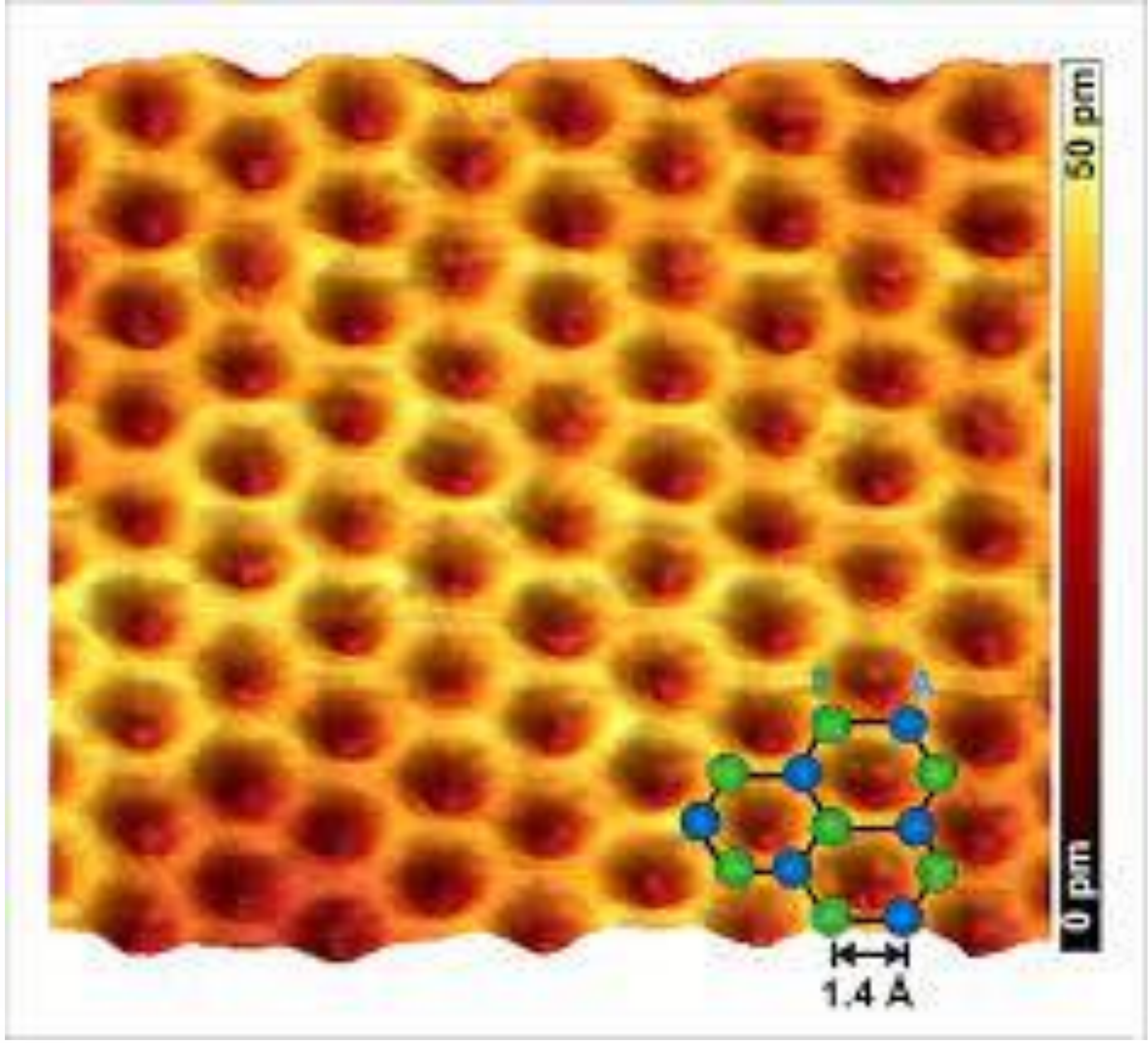
Setting of the problem - 2

- You have a math exam problem made by Prof. Calculus stating:
    - You have a fair coin.
    - You have tossed it 100 times, and it landed tails.
    - What is the probability that it will land tails during 101 attempt?

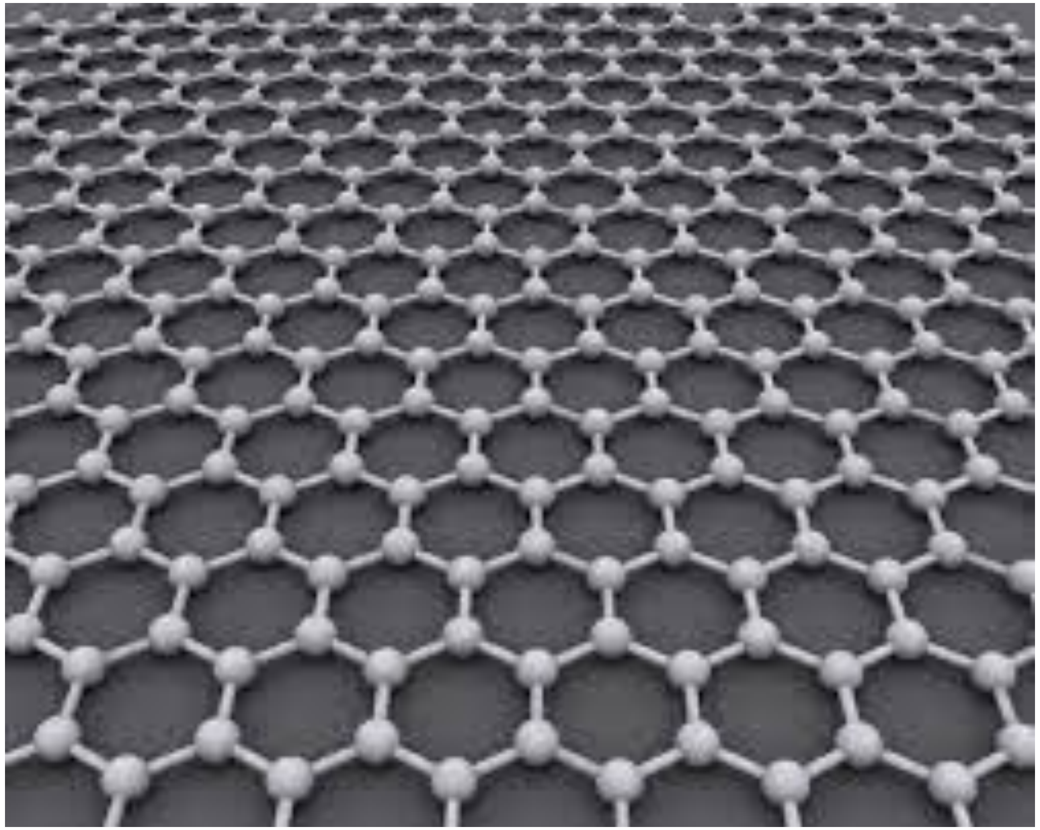


- You play the coin toss game with a sketchy person named Joe the Gambler in Las Vegas.
    - He tells you that “We have a fair coin”.
    - You have tossed it 10 times, and it landed tails.
    - What is the probability that it will land tails during 11 attempt?

Off to scientific examples

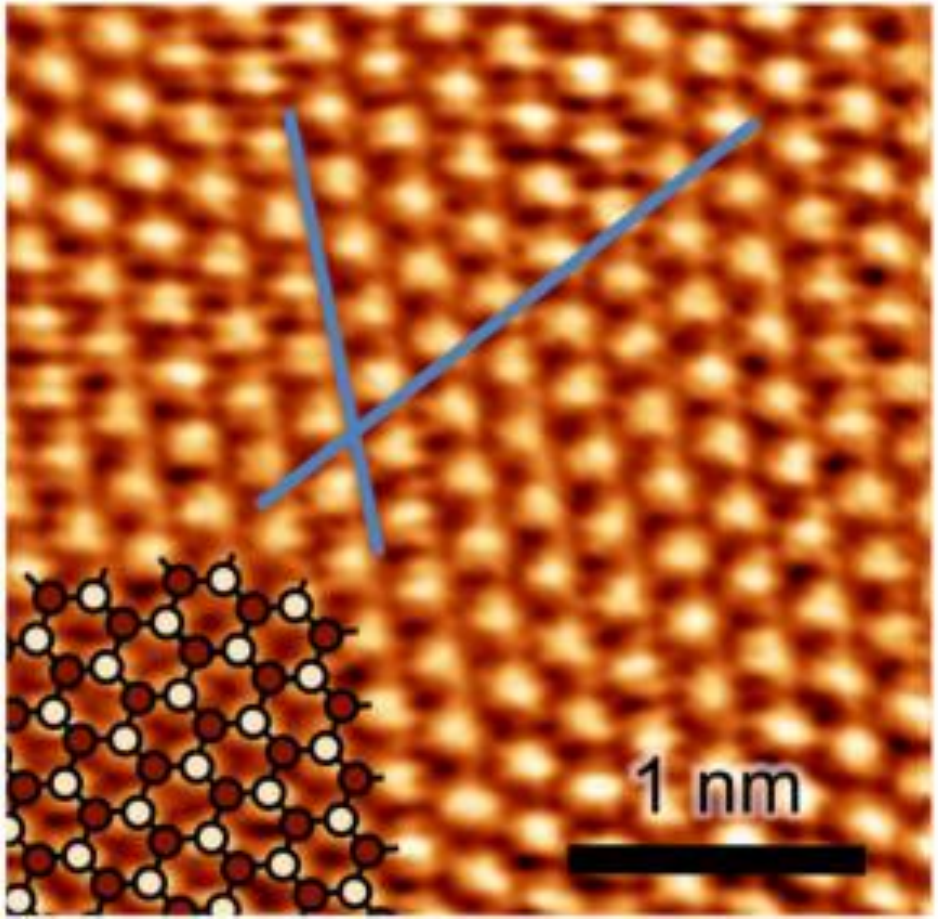


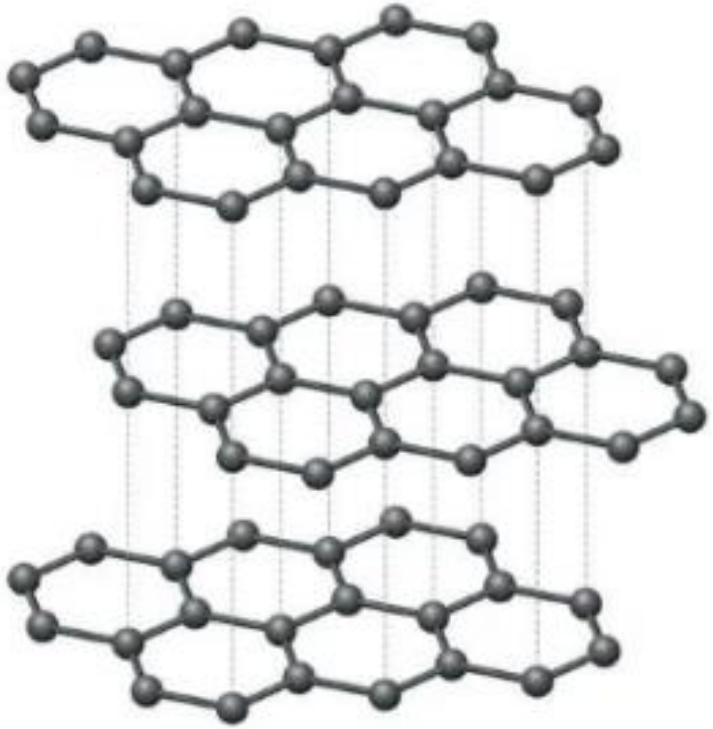
http://www.nanoscience.de/HTML/research/graphene.html



https://en.wikipedia.org/wiki/Graphene

Off to scientific examples





https://unacademy.com/content/jee/study-material/chemistry/structure-of-graphite-and-uses/

Atomically resolved STM image of the graphite surface. A schematic drawing of the hexagonal lattice of graphite is overlaid . Bright and dark filled circles correspond to the two different sublattice sites, and only the white sublattice is observed in STM image.

A. Amend, T. Matsui, H. Sato, and H Fukuyama, STS Studies of Zigzag Graphene Edges Produced by Hydrogen-Plasma Etching, Journal of Surface Science and Nanotechnology 16, 72 (2018) DOI:10.1380/ejssnt.2018.72

Frequentist paradigm

- Defines probability as a long-run frequency independent, identical trials

- Looks at parameters (i.e., the true mean of the population, the true probability of heads) as fixed quantities

This paradigm leads one to specify the null and alternative hypotheses, collect data, calculate the significance probability under the assumption that the null is true, and draw conclusions based on these significance probabilities using size of the observed effects to guide decisions

R. A. Fisher (1890–1962) https://en.wikipedia.org/wiki/Ronald_Fisher

Bayesian paradigm

- Defines probability as a subjective belief (which must be consistent with all of one's other beliefs)

- Looks at parameters (i.e., the true mean population, the true probability of heads) as random quantities because we can never know them with certainty

This paradigm leads one to specify plausible models to assign a prior probability to each model, to collect data, to calculate the probability of the data under each model, to use Bayes' theorem to calculate the posterior probability of each model, and to make inferences based on these posterior probabilities. The posterior probabilities enable one to make predictions about future observations and one uses one's loss function to make decisions that minimize the probable loss

Thomas Bayes, 1701 - 1761 https://en.wikipedia.org/wiki/Thomas_Bayes

Bayesianism vs frequentism in life

You are waiting on a subway platform for a train that is known to run on a regular schedule, only you don't know how much time is scheduled to pass between train arrivals, nor how long it's been since the last train departed.

As more time passes, do you:

(a) grow **more confident** that the train will arrive soon, since its eventual arrival *can only be getting closer*, not further away, or

(b) grow **less confident** that the train will arrive soon, since the longer you wait, the more likely it seems that either the scheduled arrival times are far apart or else that you happened to arrive just after the last train left – or both.

from Floyd Bullard

From Jeff Witmer, Introduction to Bayesian Statistics, slides 24 February 2016

Bayesianism vs frequentism in life

An opaque jar contains thousands of beads (but obviously a finite number!). You know that all the beads are either red or white but you have no idea at all what fraction of them are red. You begin to draw beads out of the bin at random without replacement. You notice that all of the first several beads have been red. As you observe more and more red beads, is the conditional probability (i.e., conditional upon the previous draws' colors) of the next bead being red

(a) decreasing, ***as it must***, since you're removing red beads from a finite population, or

(b) increasing, because you initially didn't know that there would be so many reds, but now it seems that the jar must be mostly reds.

from Floyd Bullard

From Jeff Witmer, Introduction to Bayesian Statistics, slides 24 February 2016

But what is Bayes' theorem?

Conditional probabilities: $P(a,b) = P(a|b)\,P(b) = P(a)P(b|a)$

Thus, $P(a|b) = P(a,b)/P(b) = P(a)P(b|a)/P(b)$

But what is P(b)? $P(b) = \sum_a P(a,b) = \sum_a P(a)P(b|a)$

$$\boxed{P(a|b) = \frac{P(a)P(b|a)}{P(b)} = \frac{P(a)P(b|a)}{\sum_{a^*} P(a^*)P(b|a^*)}}$$

- **Prior**
- **Posterior**
- **Likelihood**
- **Evidence**

Bayesian paradigm in science

- Bayes' theorem can be usefully re-written for science as:

Posterior: probability of the model given the data

Likelihood: probability of the data given the model

Prior: probability of the model

$$P(\text{model}|\text{data}) = \frac{P(\text{data}|\text{model})\,P(\text{model})}{P(\text{data})}$$

Evidence [can typically be absorbed into the normalization of the posterior]

Bayesian paradigm in science

Typical statistics problem: There is a parameter, θ, that we want to estimate, and we have some data.

Traditional (frequentist) methods: Study and describe P(data | θ). If the data are unlikely for a given θ, then state "that value of θ is not supported by the data." (A hypothesis test asks whether a *particular* value of θ might be correct; a CI presents a range of plausible values.)

Bayesian methods: Describe the distribution P(θ | data).

A frequentist thinks of θ as fixed (but unknown) while a Bayesian thinks of θ as a random variable that has a distribution.

Bayesian reasoning is natural and easy to think about. It is becoming much more commonly used.

From Jeff Witmer, Introduction to Bayesian Statistics, slides 24 February 2016

Bayesian method

Bayes answers the questions we really care about.

Pr(I have disease | test +) vs Pr(test + | disease)

Pr(A better than B | data) vs Pr(extreme data | A=B)

Bayes is natural (vs interpreting a CI or a P-value)

Note: blue = Bayesian, red = frequentist

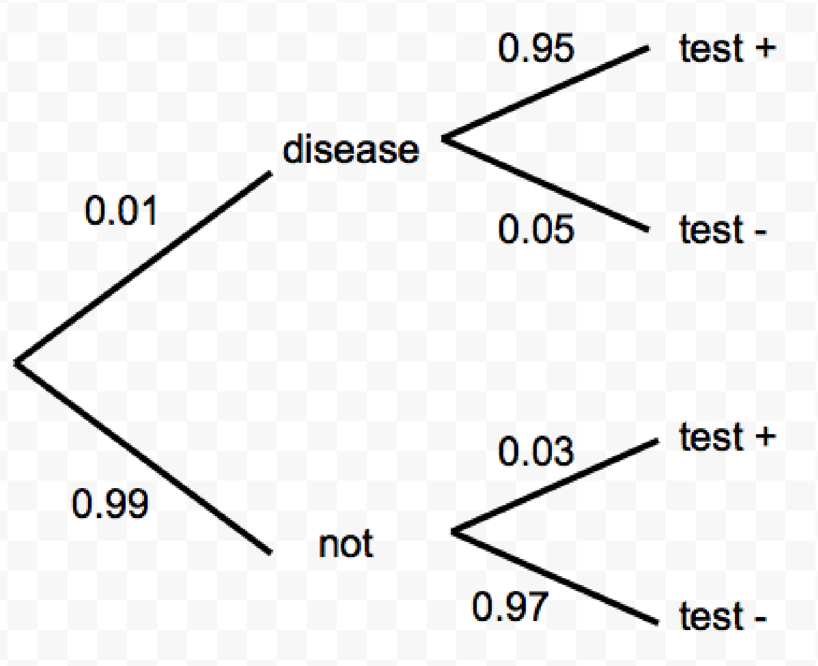
From Jeff Witmer, Introduction to Bayesian Statistics, slides 24 February 2016

Medical test example

Suppose a test is 95% accurate when a disease is present and 97% accurate when the disease is absent. Suppose that 1% of the population has the disease.

What is P(have the disease | test +)?

Medical test example



$$p(\text{dis} \mid \text{test+}) = \frac{P(\text{dis})P(\text{test+} \mid \text{dis})}{P(\text{dis})P(\text{test+} \mid \text{dis}) + P(\text{Ødis})P(\text{test+} \mid \text{Ødis})}$$

$$= \frac{(0.01)(0.95)}{(0.01)(0.95) + (0.99)(0.03)} = \frac{0.0095}{0.0095 + 0.0297} \text{ » } 0.24$$

From Jeff Witmer, Introduction to Bayesian Statistics, slides 24 February 2016

Why has Bayes not been popular?

(1) Without Markov Chain Monte Carlo, it wasn't practical.

(2) Some people distrust prior distributions, thinking that science should be objective (as if that were possible).

Bayes is becoming much more common, due to MCMC.

From Jeff Witmer, Introduction to Bayesian Statistics, slides 24 February 2016

Main Elements of Bayesian Models

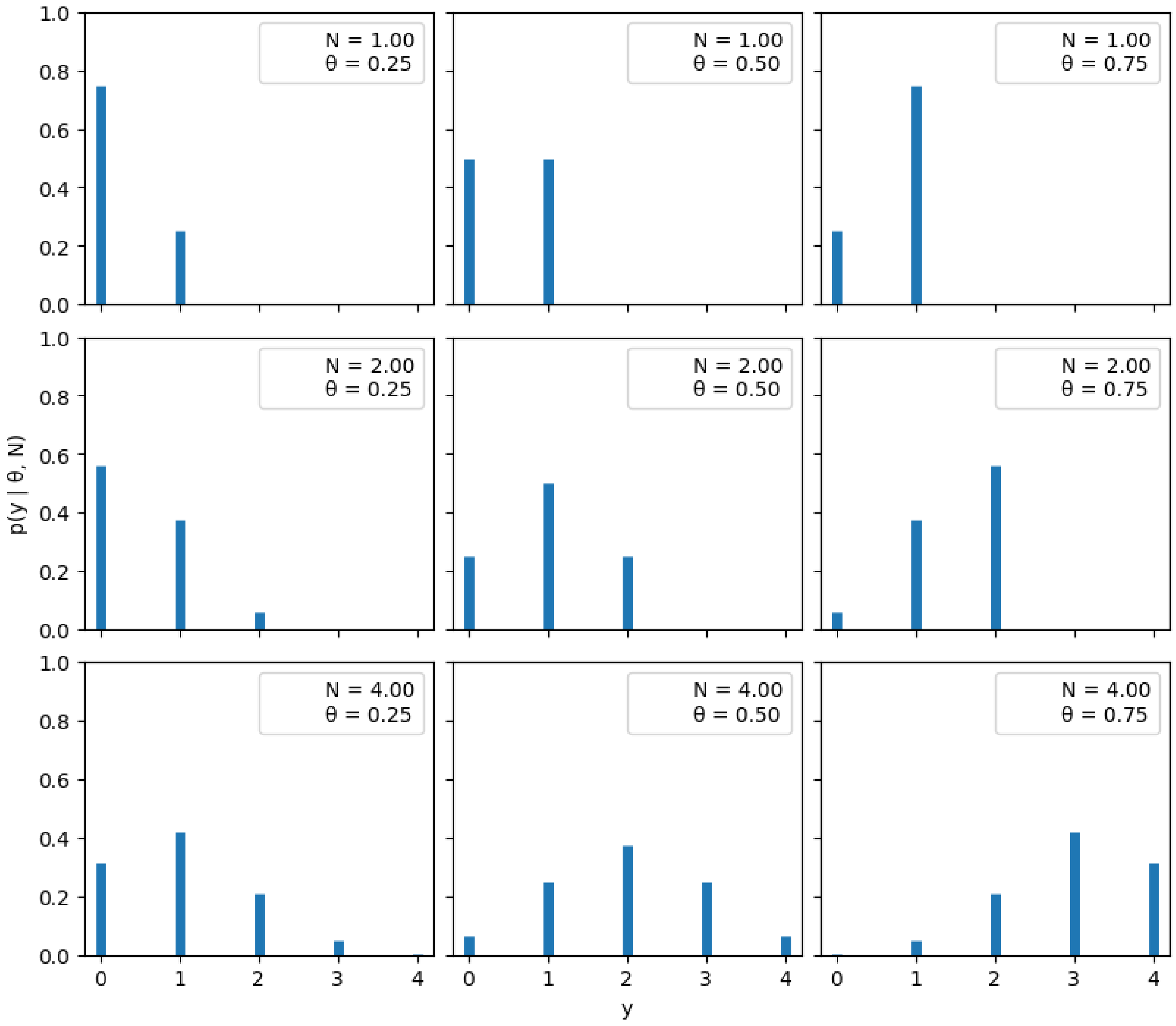
- **Prior Distribution** – use probability to quantify uncertainty about unknown quantities (parameters)
- **Likelihood** – relates all variables into a “full probability model”
- **Posterior Distribution** – result of using data to update information about unknown quantities (parameters)

Coin Toss

- **Probability of tails:** p
- **Probability of heads:** 1-p

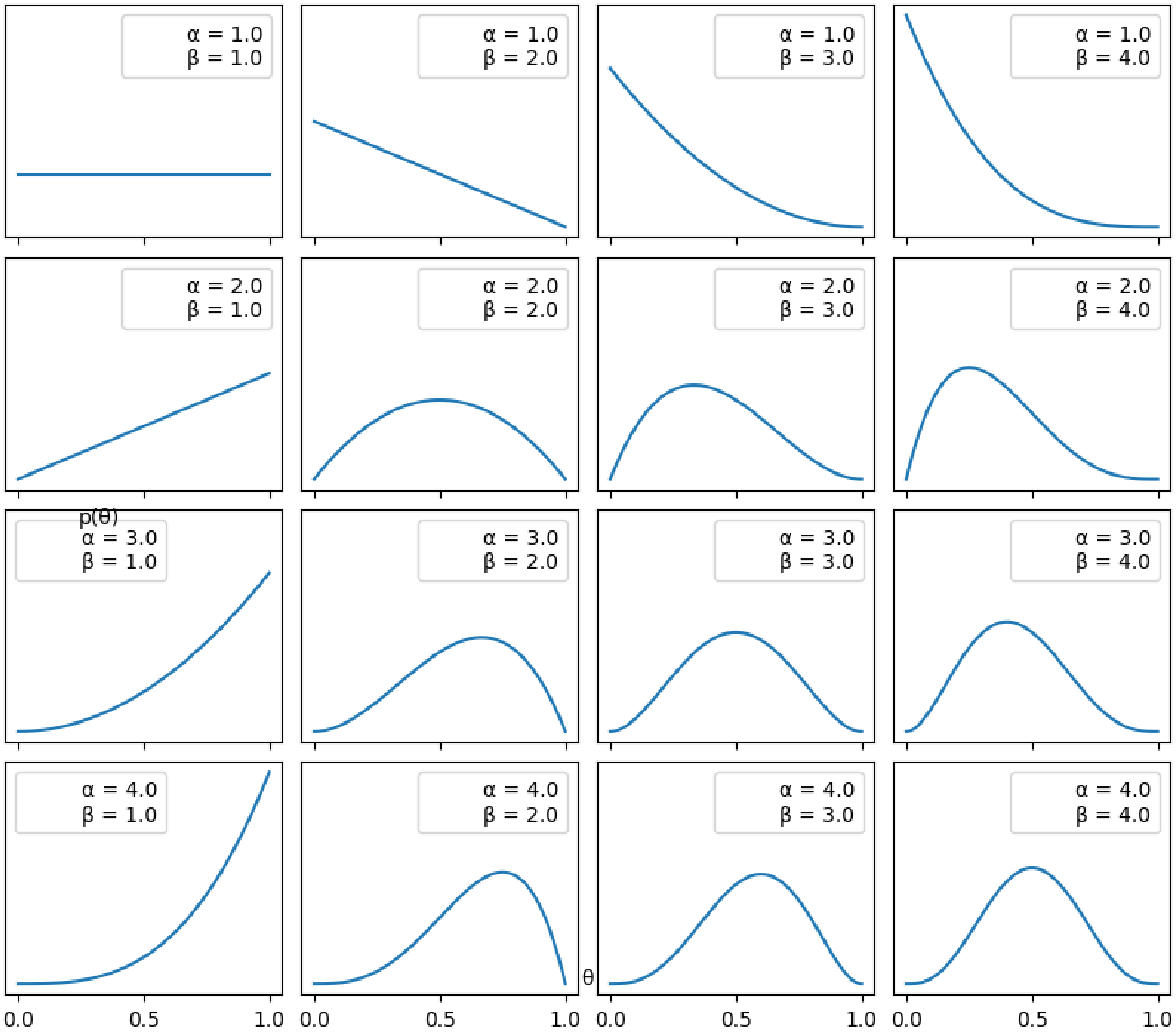
- Probability of getting n tails out of N tosses of coin:

$$C_N^n p^n (1-p)^{N-n}$$



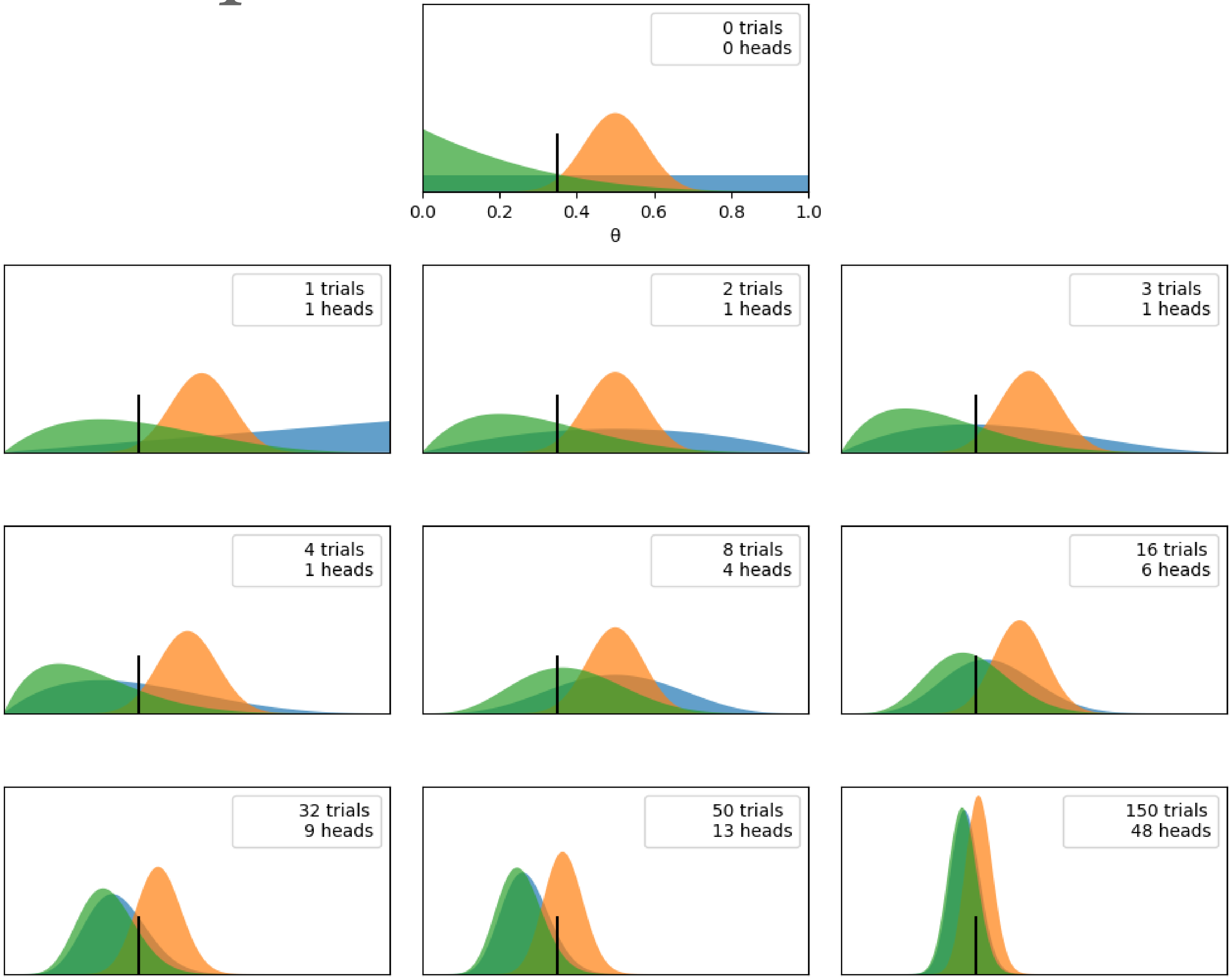
Beta distribution: conjugate to binomial

$$
\begin{aligned}
f(x;\alpha,\beta) &= \text{constant}\cdot x^{\alpha-1}(1-x)^{\beta-1} \\
&= \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1}\,du} \\
&= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\, x^{\alpha-1}(1-x)^{\beta-1} \\
&= \frac{1}{\mathrm{B}(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}
\end{aligned}
$$



https://en.wikipedia.org/wiki/Beta_distribution

Can we learn p from several coin tosses?



Conjugate Prior: A prior distribution is said to be conjugate to a likelihood function if the resulting posterior distribution is in the same family as the prior. In other words, if you start with a certain type of distribution as your prior, and after observing data and updating your beliefs (via Bayes' theorem), your posterior is still of that same type, then the prior is a conjugate prior for that likelihood function.

• **Computational Convenience**: Using conjugate priors can greatly simplify the mathematical computation required to find the posterior distribution. This can be especially useful in situations where you're continually updating your beliefs with new data; with conjugate priors, you can easily update your posterior without complex integrals or advanced sampling methods.

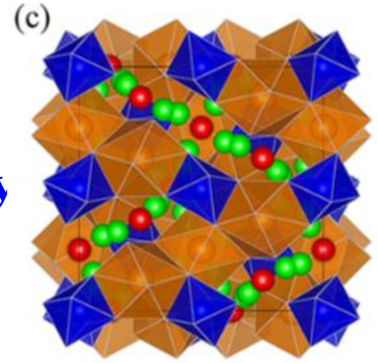
•**Analytical Solutions**: Many standard problems in Bayesian statistics can be solved analytically using conjugate priors, leading to exact posterior distributions.

Examples:

1. **Beta distribution is conjugate to the Binomial likelihood**: This means that if you have a Binomial likelihood (e.g., flipping coins) and a Beta-distributed prior on the probability of heads, the resulting posterior distribution after observing some data will also be a Beta distribution.
2. **Gamma distribution is conjugate to the Poisson likelihood**: If you're observing the number of events occurring in fixed intervals of time or space (modeled by a Poisson distribution) and have a Gamma-distributed prior on the rate parameter, the posterior will also be Gamma-distributed.
3. **Normal distribution is conjugate to itself**: If both the likelihood and the prior are normally distributed, then the posterior will also be normally distributed.

Symmetry in materials science

- Crystal structure and symmetry play a crucial role in material science.
- Knowing chemical composition and crystal structure - the way atoms are arranged in space is an essential ingredient for predicting properties of a material.
    - Phase transitions
    - Order parameters
    - Physical properties
    - Vibrations and quasiparticles
- It is well known fact that the crystal structure has a direct impact on materials properties.



Cubic
High Li ion conductivity

LLZO

Tetragonal
Conductivity two orders of magnitude lower

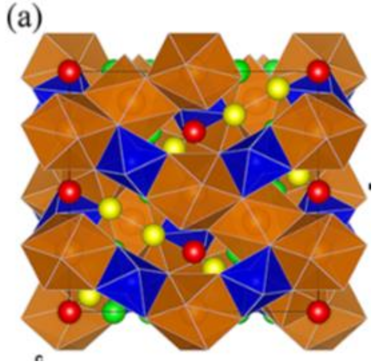


Image: J. Phys. Chem. C 2018, 122, 1963–1972

Usual way: scattering

- Diffraction techniques were predominantly used to identify crystal structure and symmetry in condensed matter physics community.

- These techniques have been successfully applied to atomic[1], magnetic[2], superconducting vortex[3] and protein[4] lattices and their structures were disinterred.

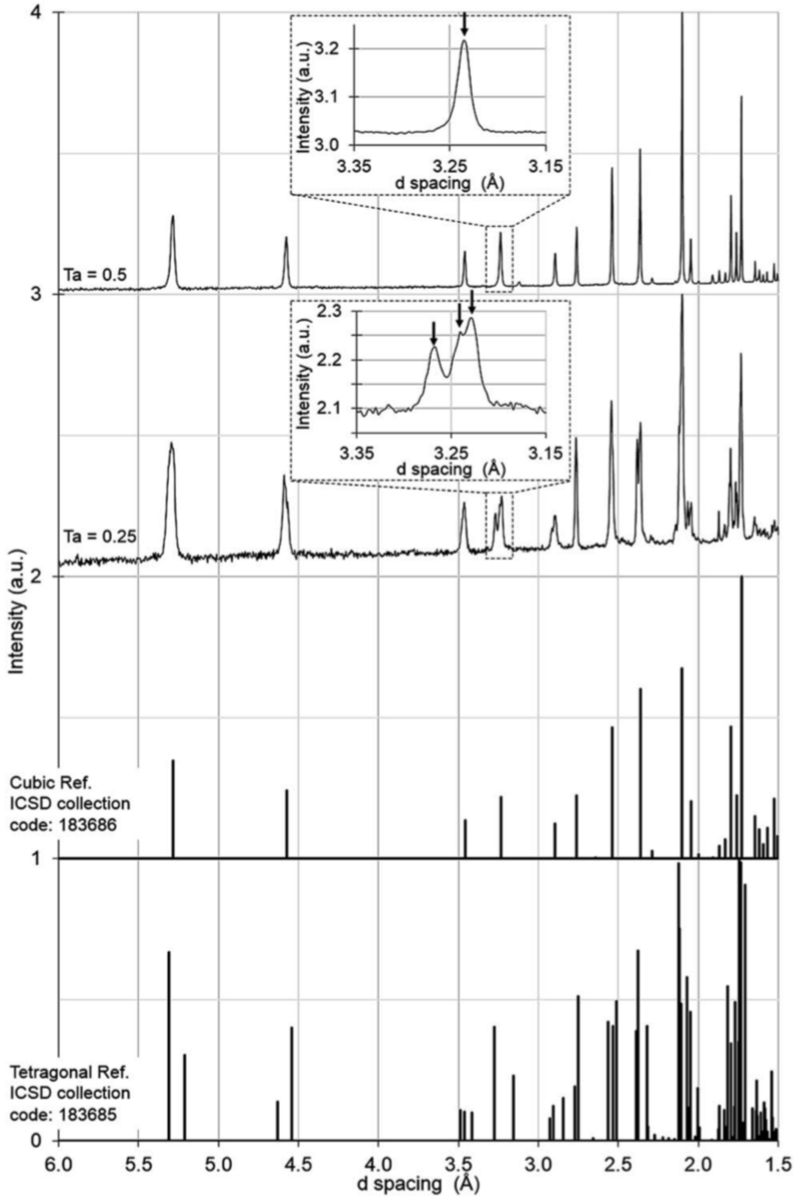
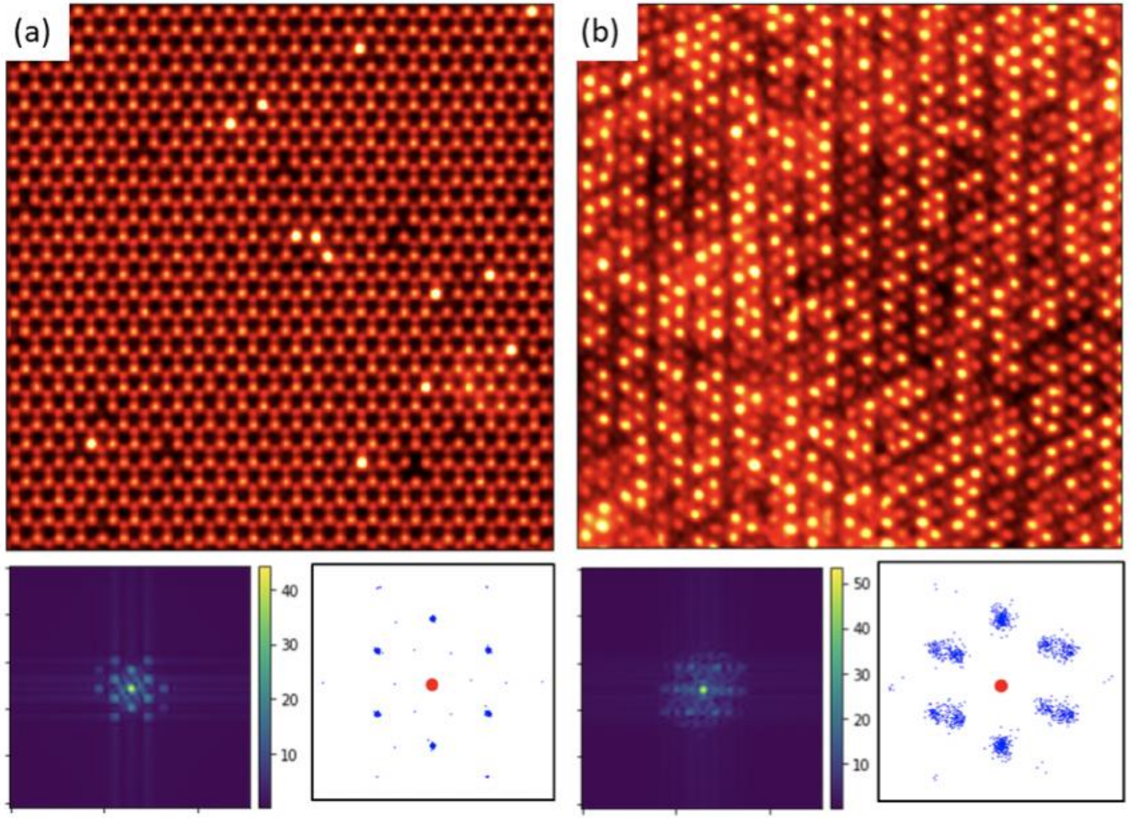


Image: J. Mater. Chem. A, 2014, 2, 13431–13436

1) Journal of Applied Crystallography, 1969. **2**(2): p. 65-71.
2) Physica B: Condensed Matter, 1993. **192**(1): p. 55-69.
3) Reports on Progress in Physics, 2011. **74**(12)
4) Nature, 1958. **181**(4610): p. 662-666.

… But what about microscopy?

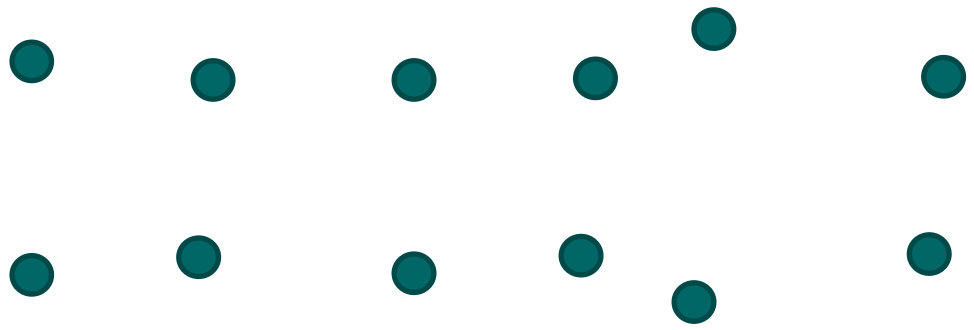
- Scanning Transmission Electron Microscopy has gained significant traction in the last decades and resolution in the order of picometers was achieved.
- Real space images or Fourier transformed images can now be used to determine symmetry in crystal structures.
- Machine learning techniques (mostly CNNs) were applied on real space and k-space images to determine underlying symmetries.
- But: in all cases we essentially apply macroscopic criterion developed for systems with large number of atoms to systems with small number of atoms
- What is the limit?



High-resolution scanning transmission electron microscopy images of $Mo_{1-x}Ru_xS_2$ with x = 0.05, 0.55 and their corresponding Fourier transforms.

… But what about microscopy?

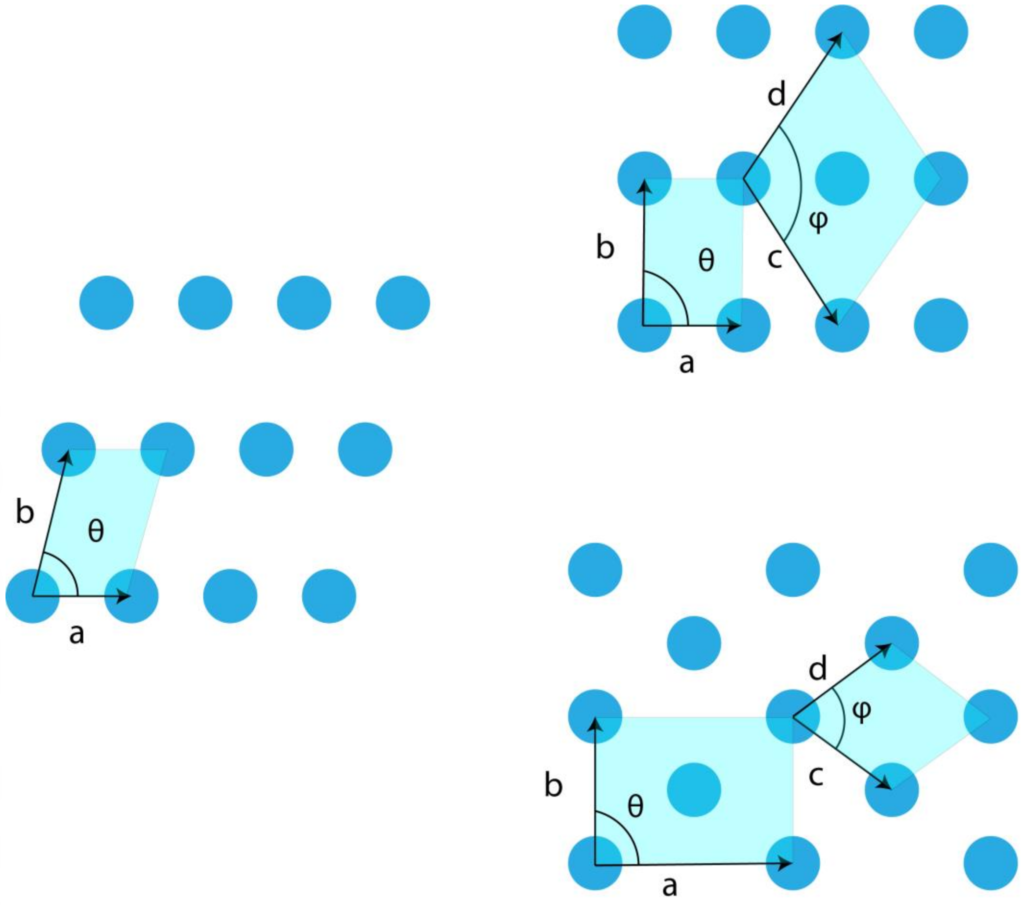
- Can we start talking symmetry as a microscopic property?
- If so, how many lattice units do we need before we define a particular symmetry in a crystal?

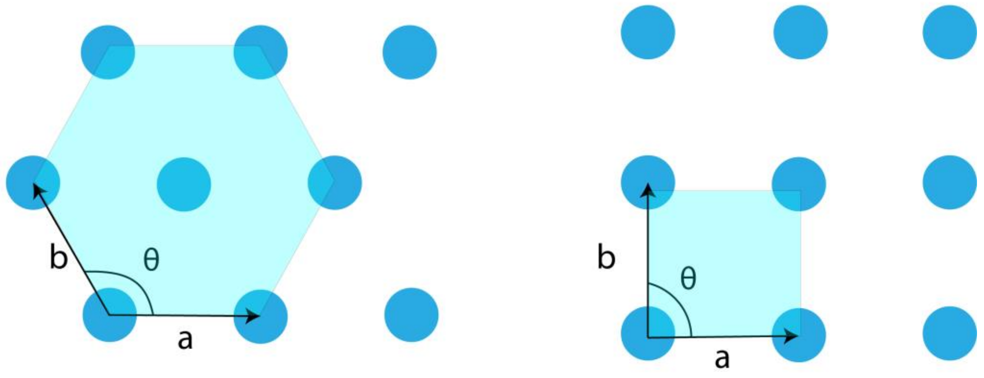


…….

2D Bravais lattices

Let’s start by the definition (Kittel):
Look at lattice parameters:
Square?: a = b, $\boldsymbol{\theta}$ = 90
Rectangular, a != b,
And so on…

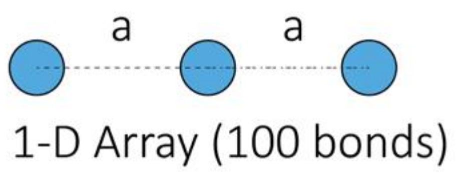




- In macroscopic systems, these equalities determine peak splitting in scattering data
- What about real space images?
- Especially when our data is limited?

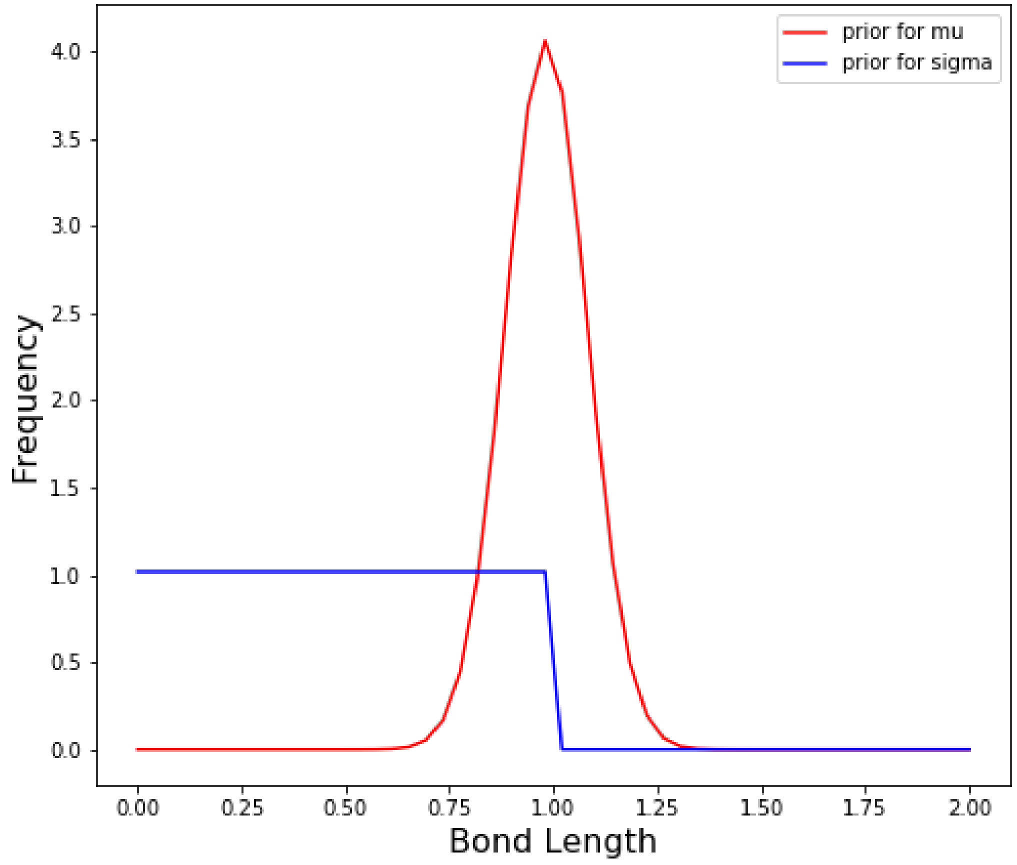
2D Bravais lattices

- An array of hundred bond lengths is generated using a normal distribution
- Mean of the Gaussian is $\boldsymbol{\mu}^* = 1.0$
- Whereas standard deviation is $\boldsymbol{\sigma}^* = 0.1$
- Standard deviation represents the bond disorder present in the system whereas mean represents the average bond length

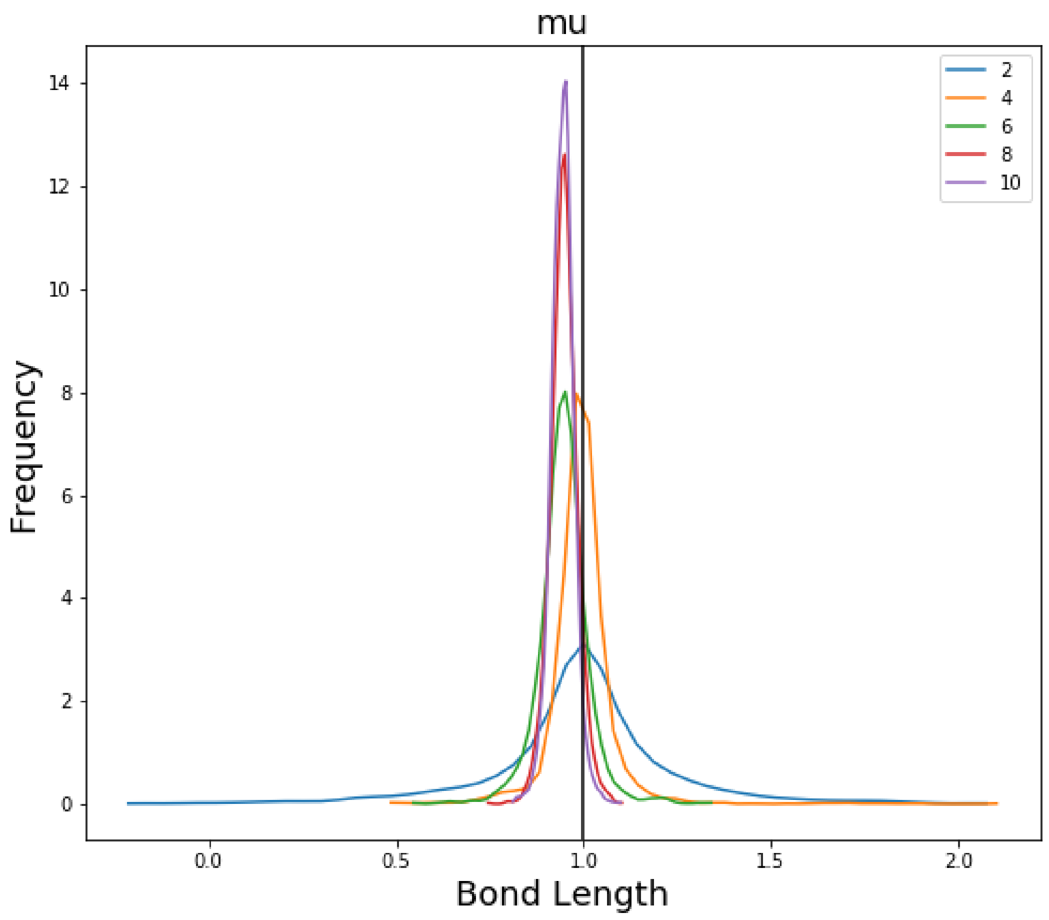


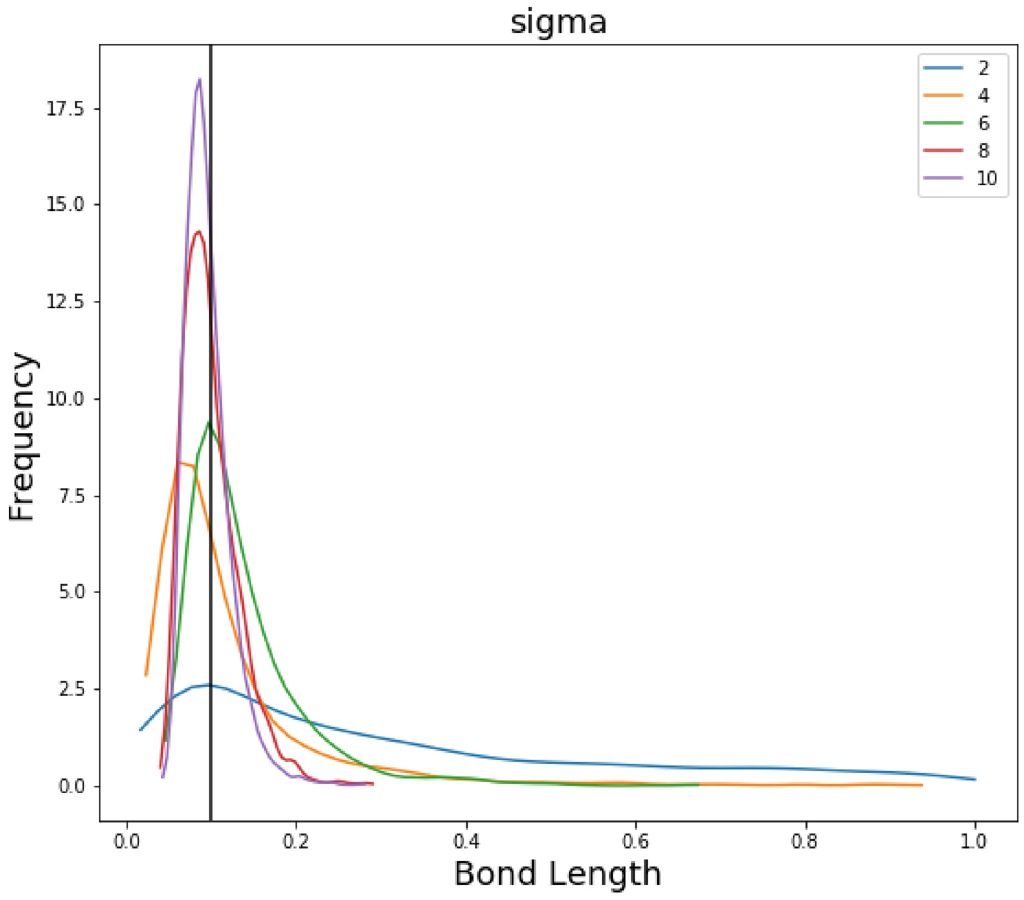
Priors

- Prior information of microscopes and of the material under consideration can be used in generating priors.
- Or if you know nothing like Jon Snow, the priors can be uniform
- For this analysis, priors are formed using the first twenty data points of the dataset generated.
- A gaussian prior for bond lengths and an uniform priors for bond disorders are considered.

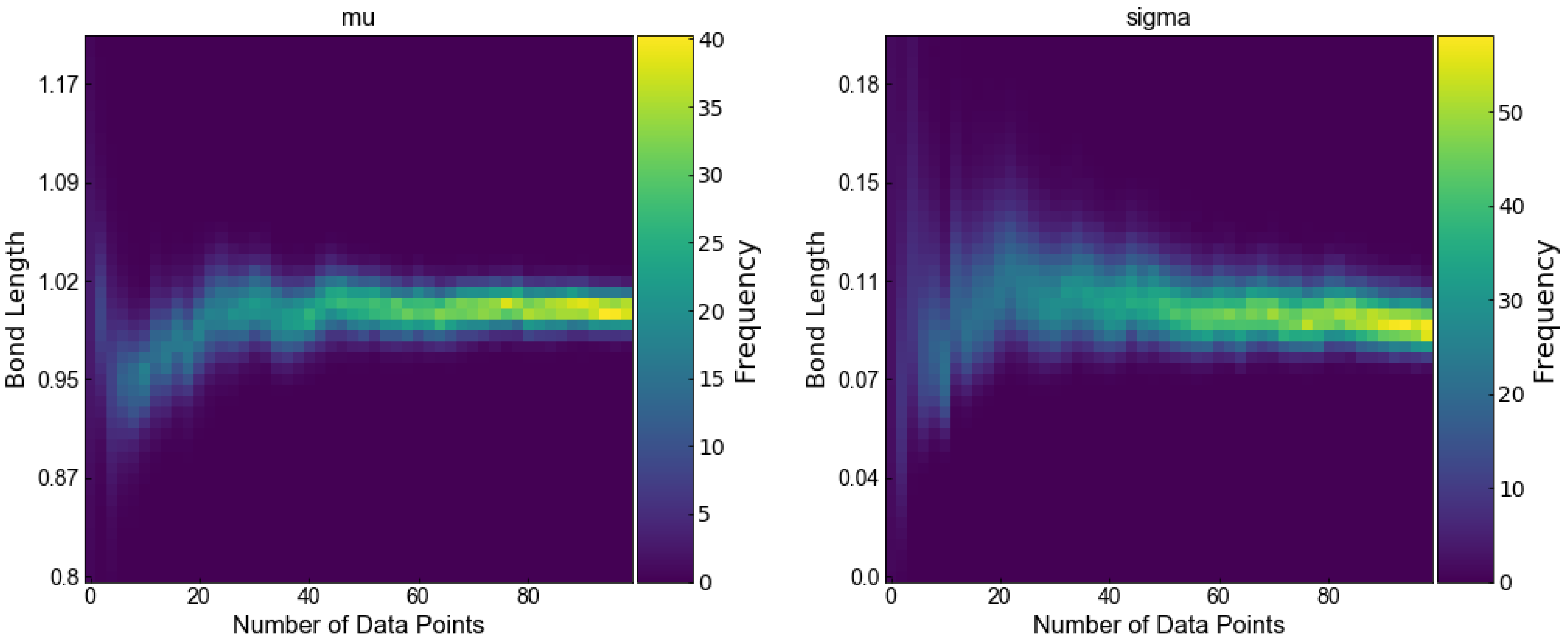


Few initial iterations

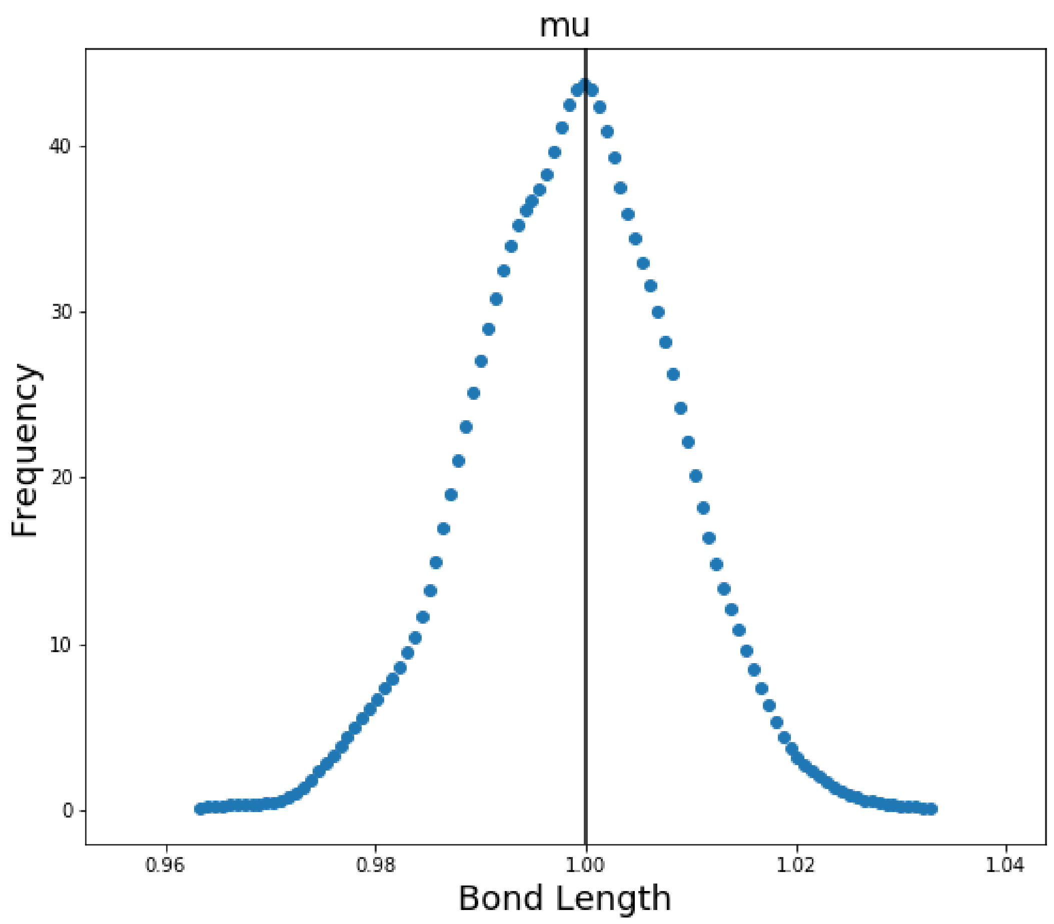


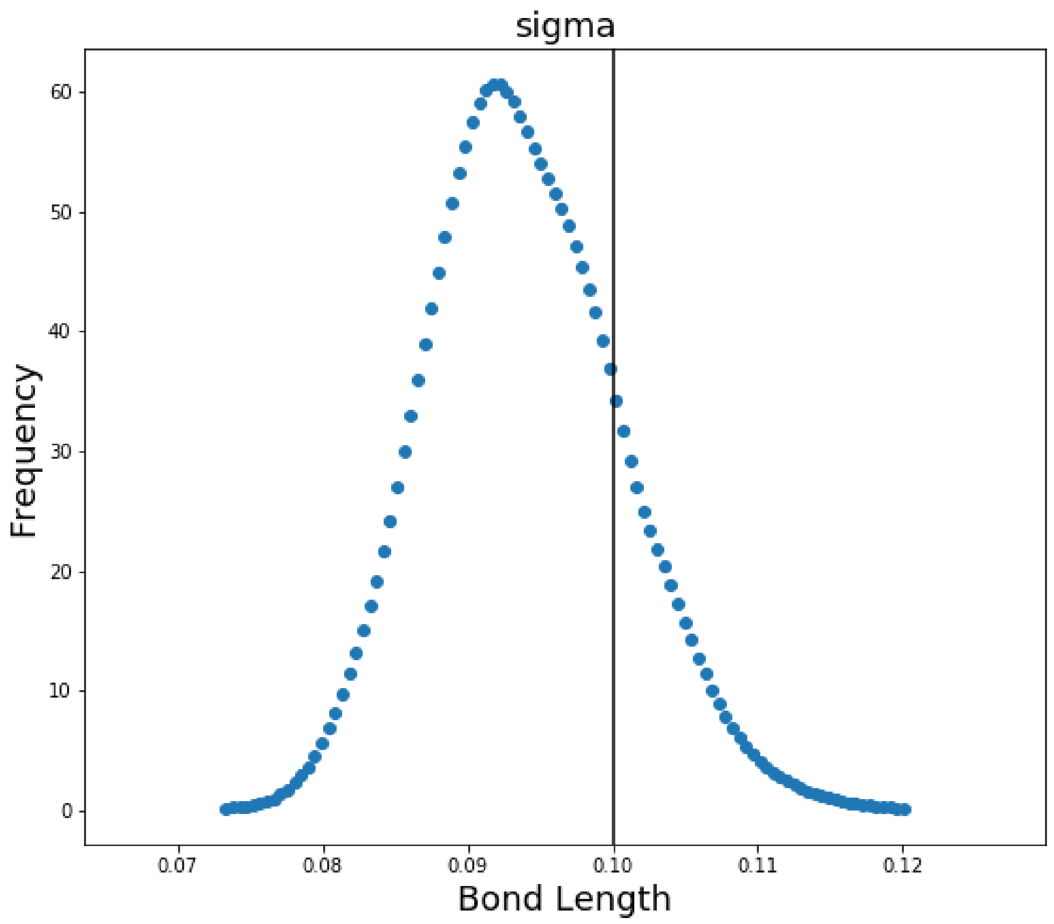


More iterations….



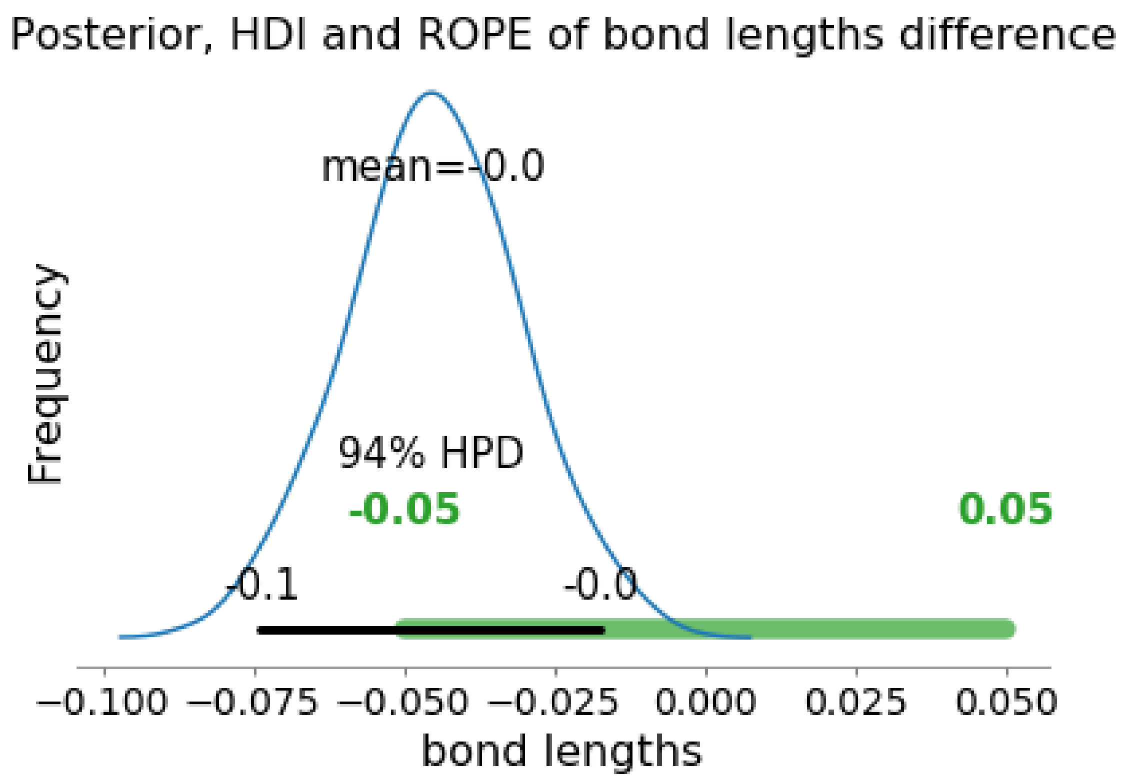
Final posterior distribution





Decision making: ROPE

- We construct an interval (ROPE) around the hypothesis and a decision can be made by comparing the intervals of HDI (94% credible interval) and ROPE.
- Decision rules for different criteria are listed in "Bayesian Analysis with Python" by Osvaldo Martin



Priors are the key!

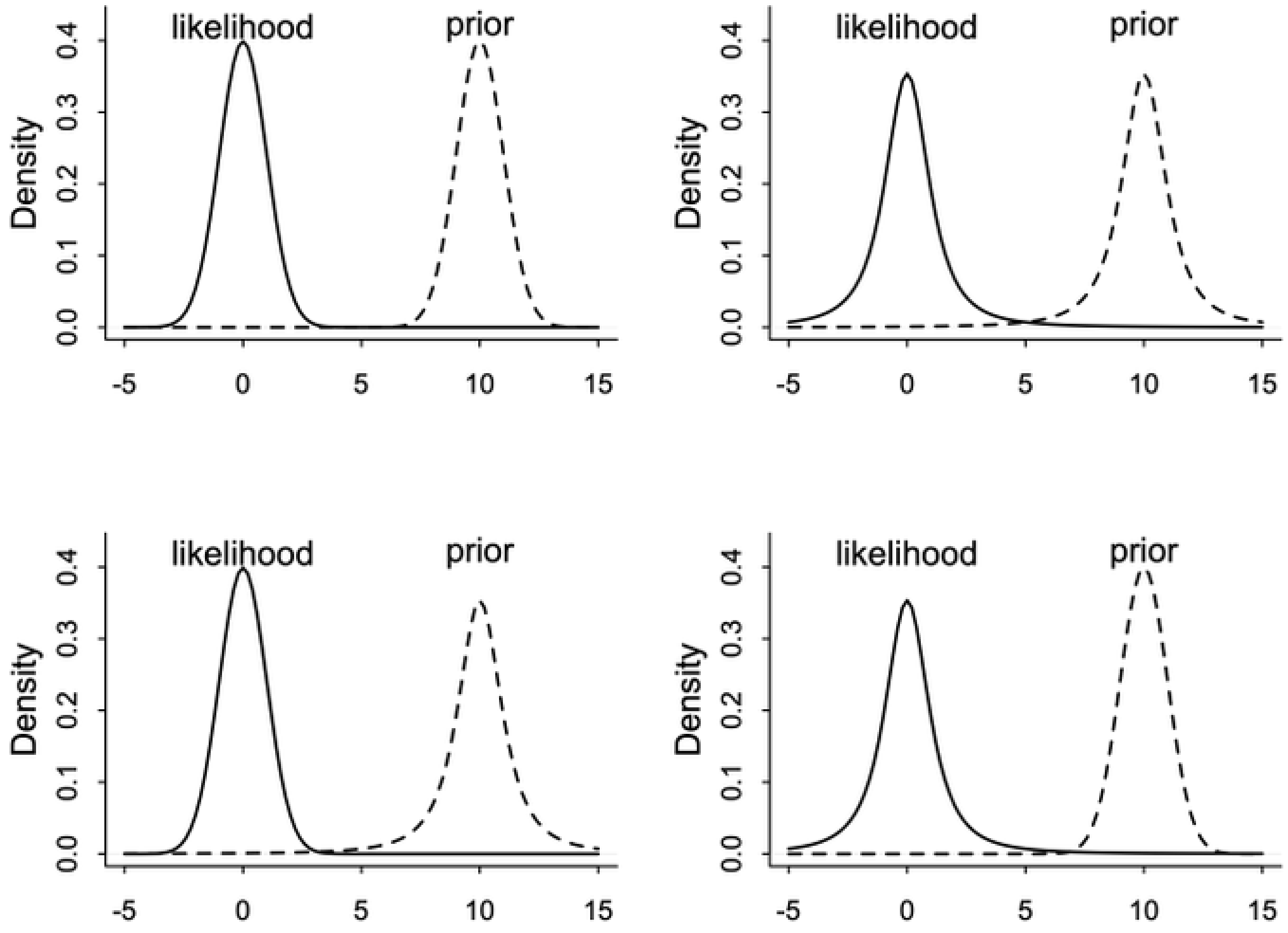
- Most Bayesian analysts assume "uninformative priors"
  – no strong assumptions about the parameter estimates other than the shape of their distributions
- When we use uninformative priors and analyze the data using both a traditional approach and a Bayesian approach, the resulting parameter estimates are the same (for all practical purposes) = *no strong rationale for Bayesian*
  – Uninformative priors means the results are strongly determined by the current experiment's data

When to use strong priors

- When you are willing to specify informative priors
- When there is no existing analysis for your design

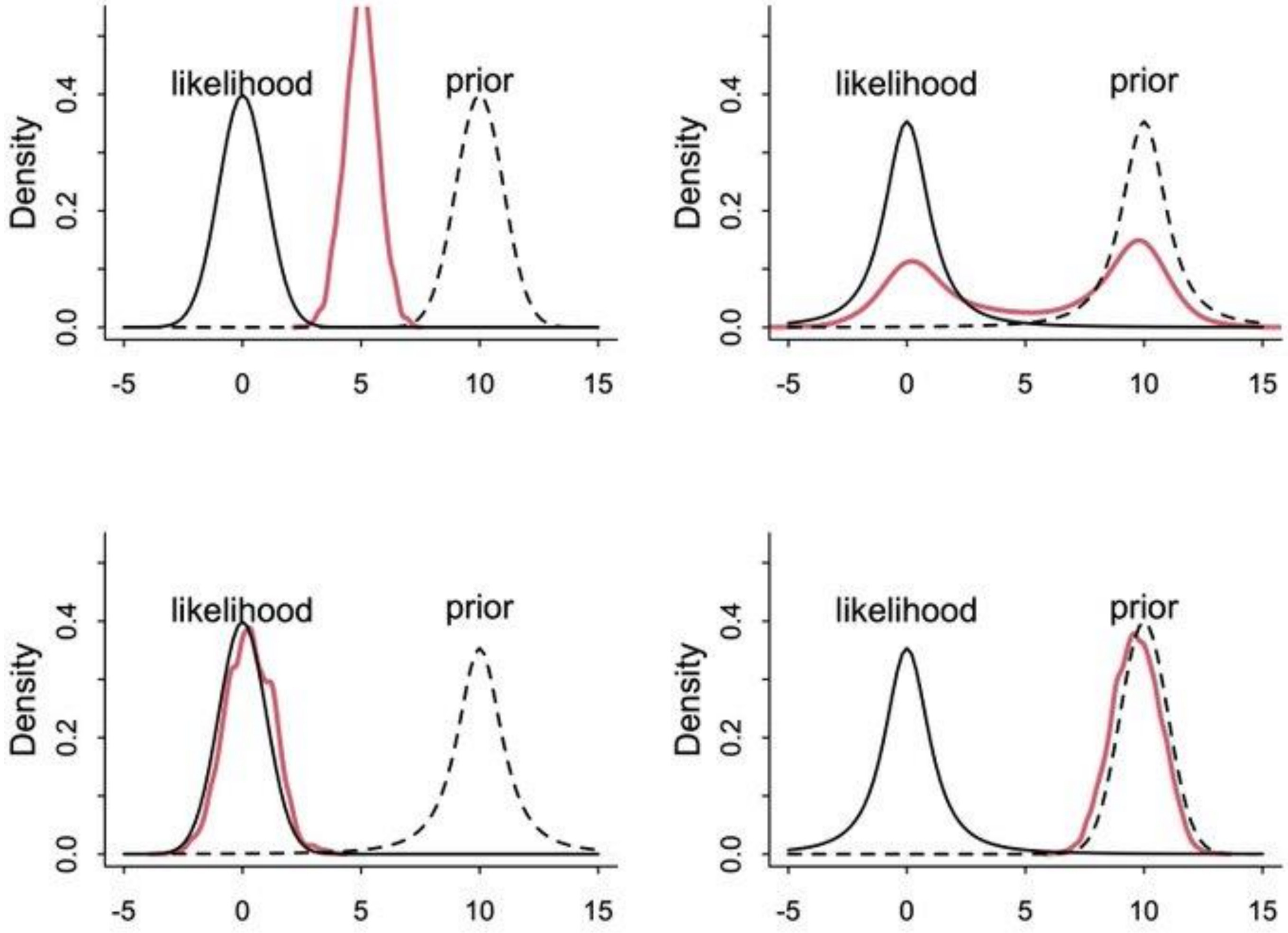
- Narrower priors will have a bigger effect on the posterior estimation
- The effect will be larger if the new data is limited or highly variable
  - This situation indicates that the new data are equivocal and offer little new

McElreath Quartet



https://twitter.com/rlmcelreath/status/1701165075493470644

McElreath Quartet



https://twitter.com/rlmcelreath/status/1701165075493470644

How does it work?

Human scientist

Hypothesis making

Model of the process (priors, model)

Bayesian Inference

Inferred parameters of the model (posteriors)

Prediction

Simulated data

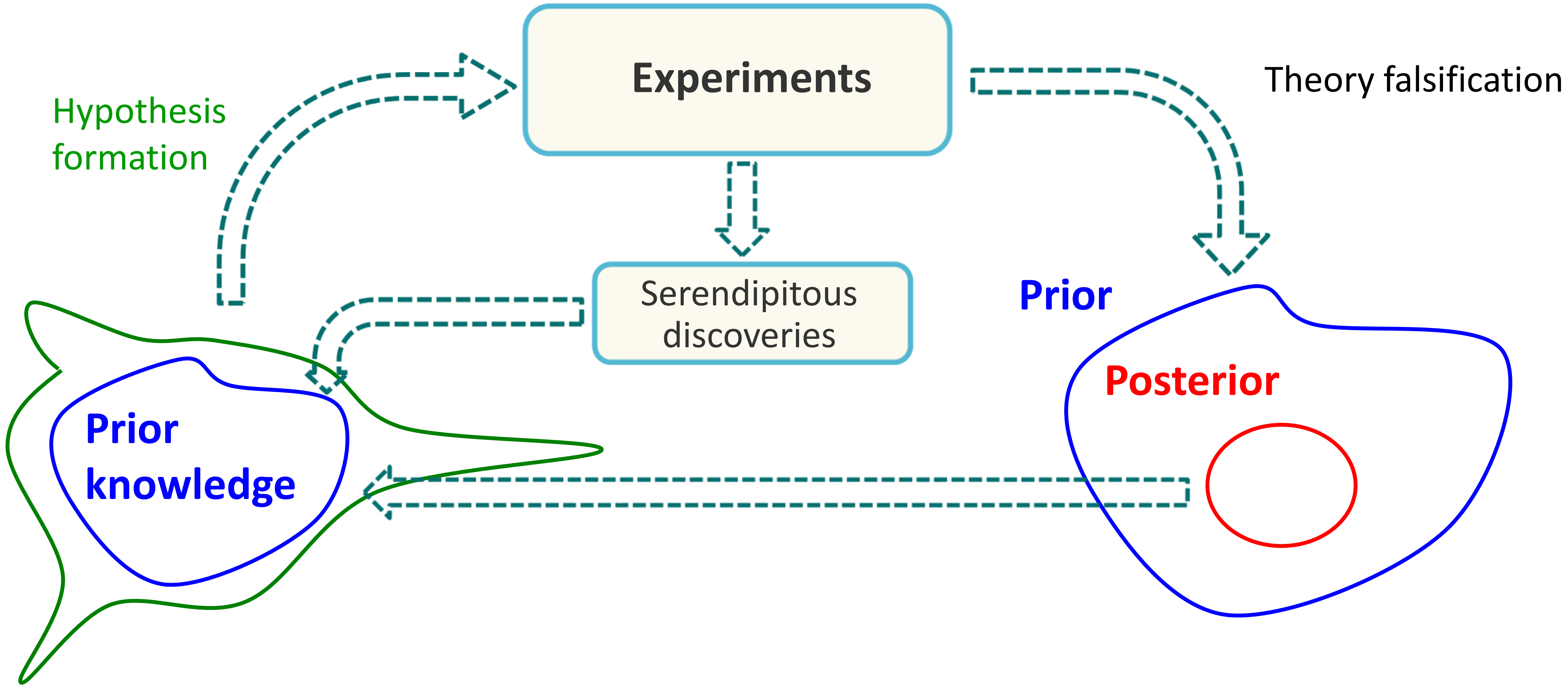
Physical process (generates data with certain distribution)

Experiment

Observed Data

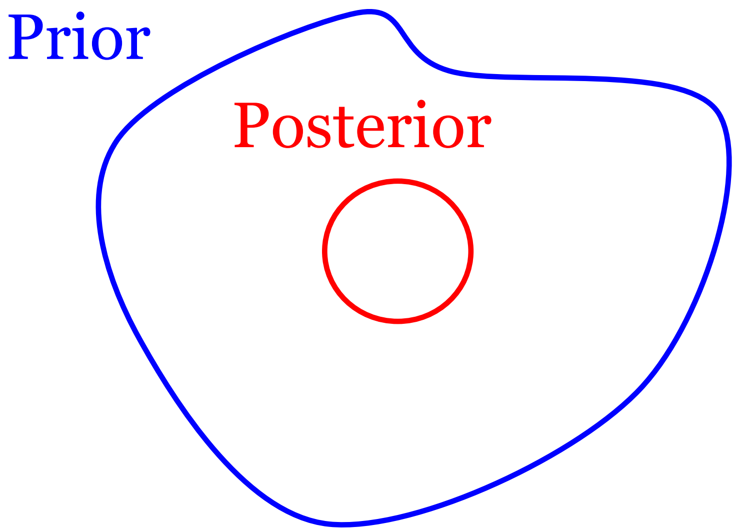
Posterior predictive check

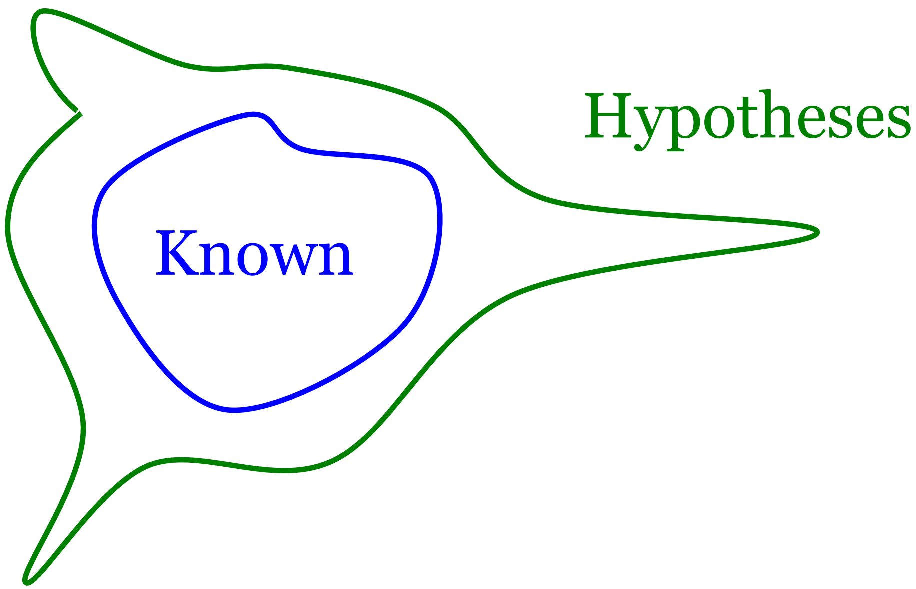
Where do Bayesian methods fit into R&D?



Rewards: Why are we doing science/R&D?
Policies: Exploration-exploitation balance
Instrument development: New tools create new opportunities
Hypothesis making: Extrapolation into the unknown

What Bayesian Methods (and ML) cannot do?



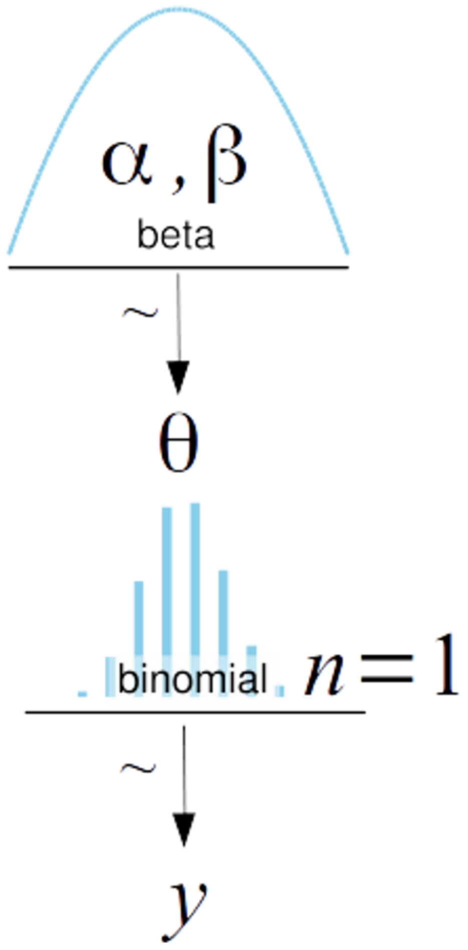


Refinement:
Can be defined as probabilistic model

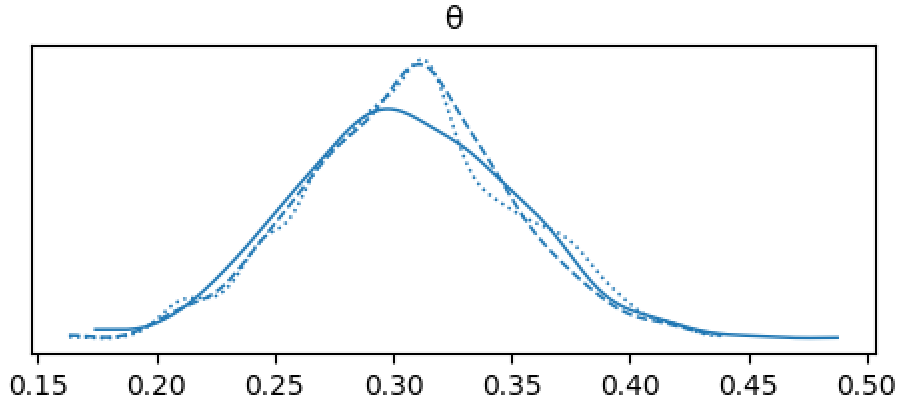
Hypothesis formation:
How can we do it?

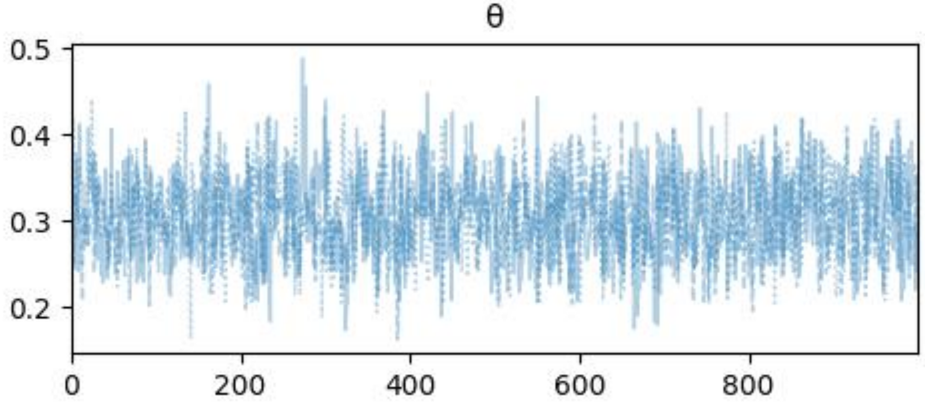
Let’s toss a coin!

Let's toss a coin (with a PYMC)!



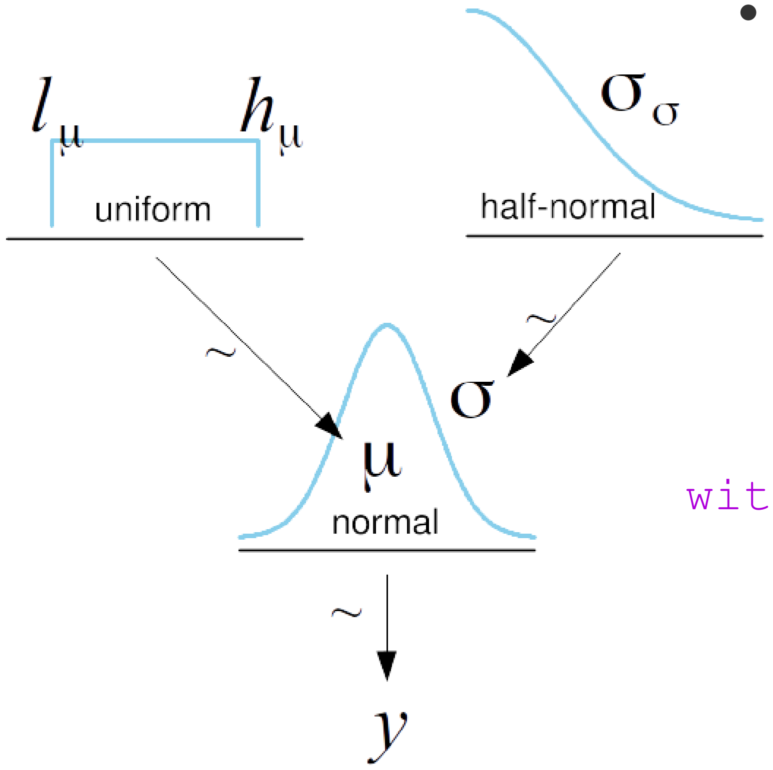

```
with pm.Model() as our_first_model:
    θ = pm.Beta('θ', alpha=1., beta=1.)
    y = pm.Bernoulli('y', p=θ, observed=data)
    trace = pm.sample(1000, random_seed=123, chains = 3)
```





Colab: 19_PYMC_BayesianEZ.ipynb

But how do we get physics done?

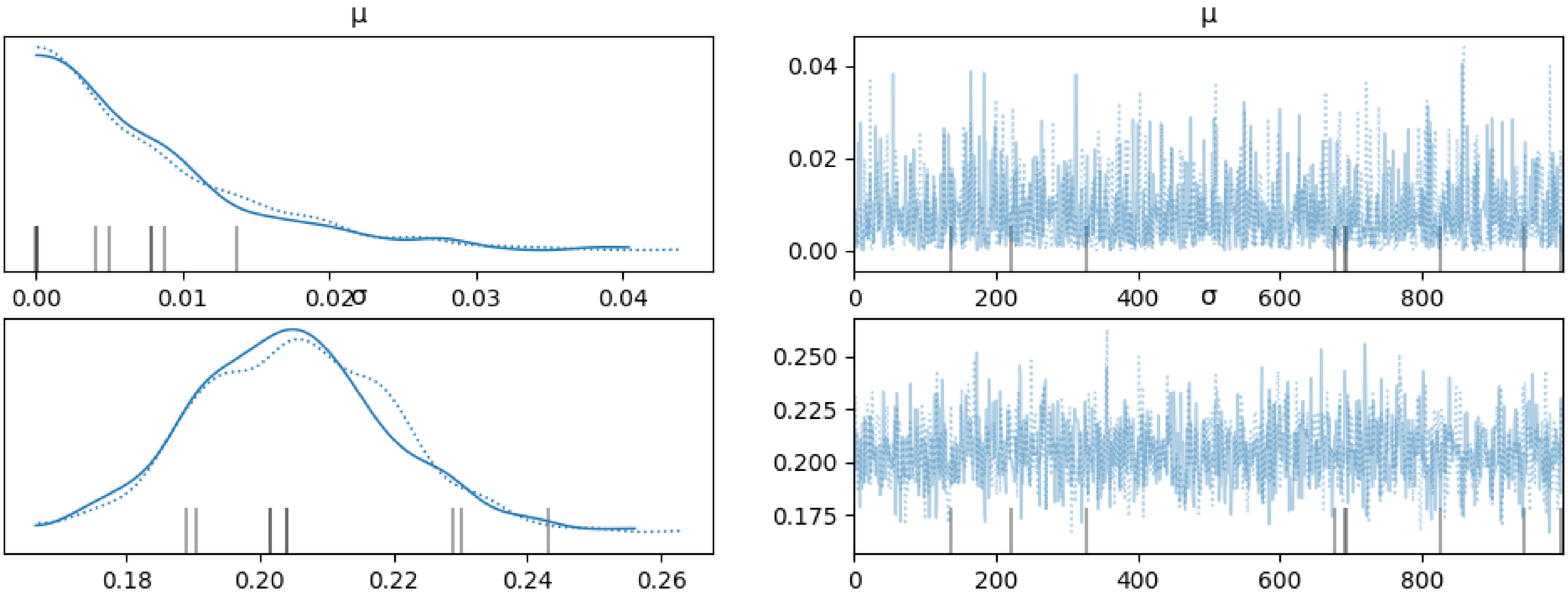


- Imagine that we have some observational data
- Can we say which distribution has it come from?

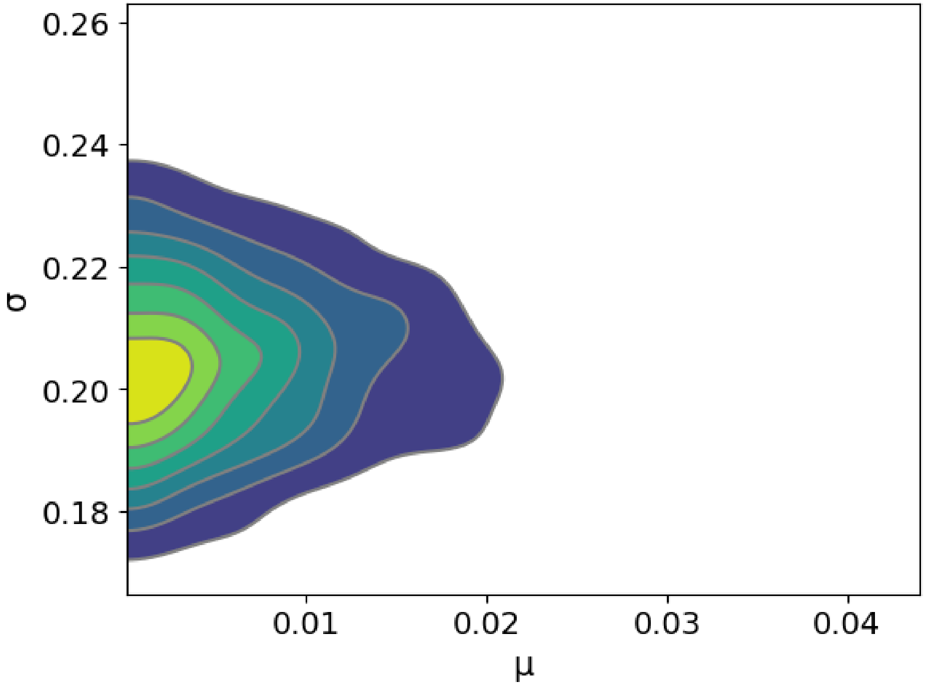
Define Bayesian model!

```
with pm.Model() as model_g:
    μ = pm.Uniform('μ', lower=0, upper=2)
    σ = pm.HalfNormal('σ', sigma=1)
    y = pm.Normal('y', mu=μ, sigma=σ, observed=arr)
    idata_g = pm.sample(1000)
```

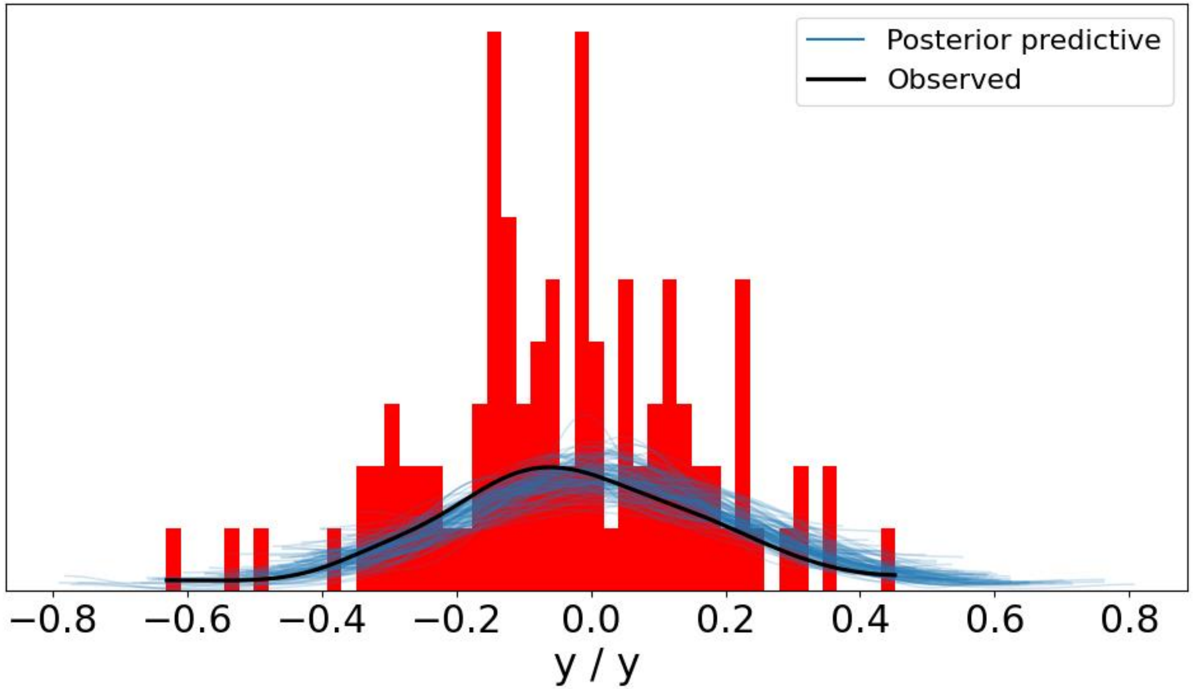
Presto!


μ
0.00
0.01
0.02
0.03
0.04
μ
0.04
0.02
0.00
0
200
400
600
800
σ
0.18
0.20
0.22
0.24
0.26
σ
0.250
0.225
0.200
0.175
0
200
400
600
800

Presto!



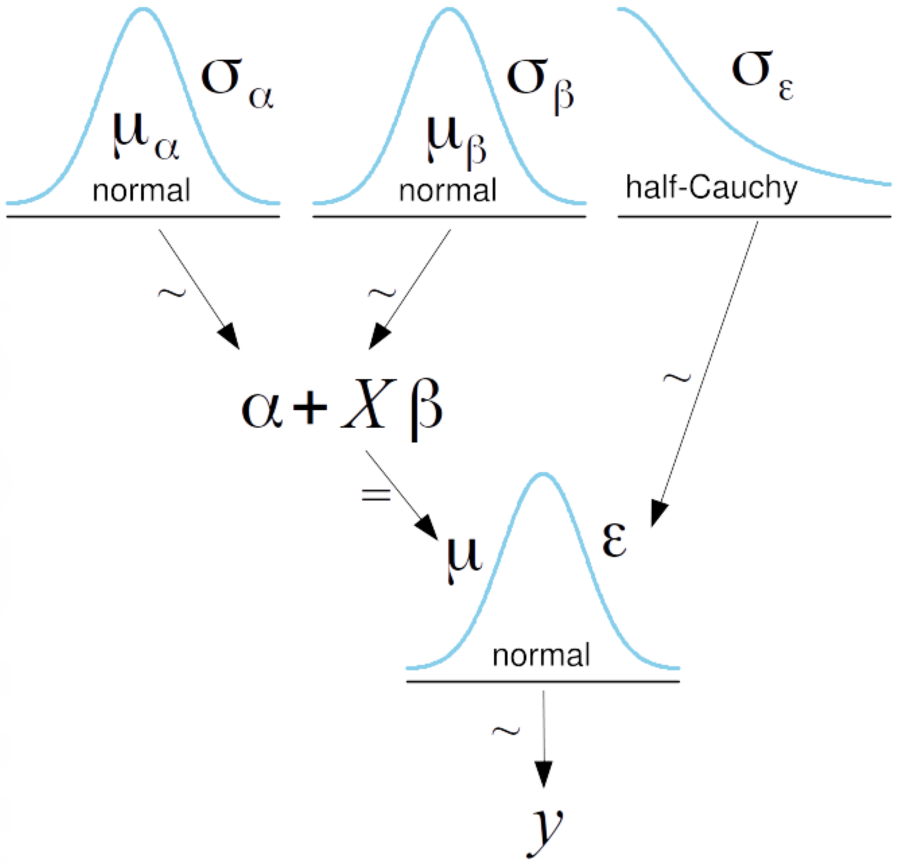
0.26
0.24
0.22
0.20
0.18
σ
0.01
0.02
0.03
0.04
μ



Posterior predictive
Observed
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
y / y

Linear regression Bayesian style

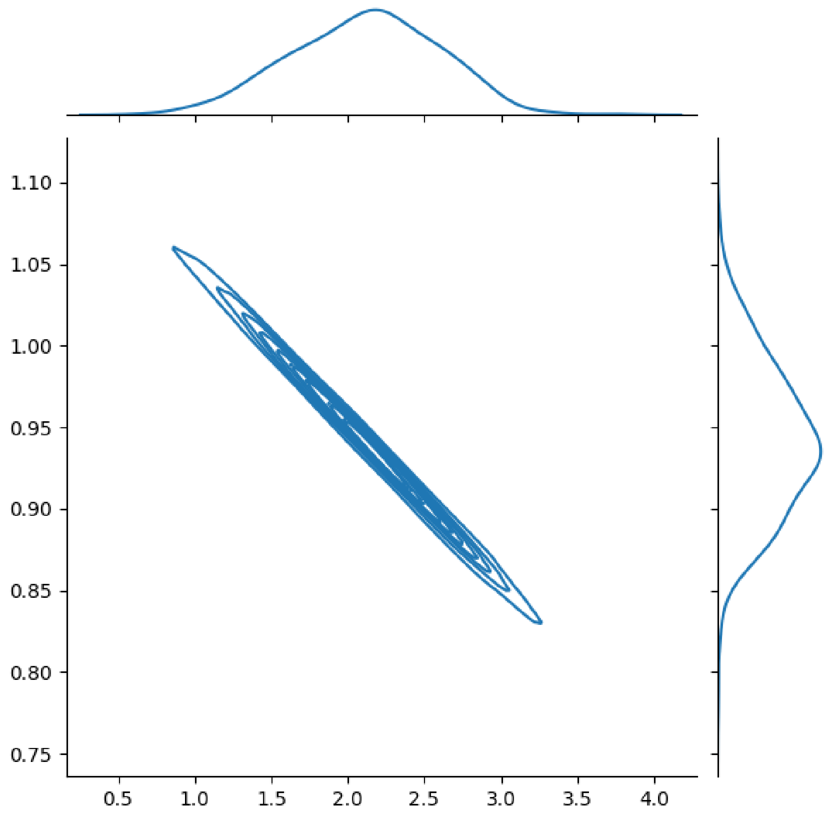
- Imagine that we have some observational data
- Can we fit it by linear function?

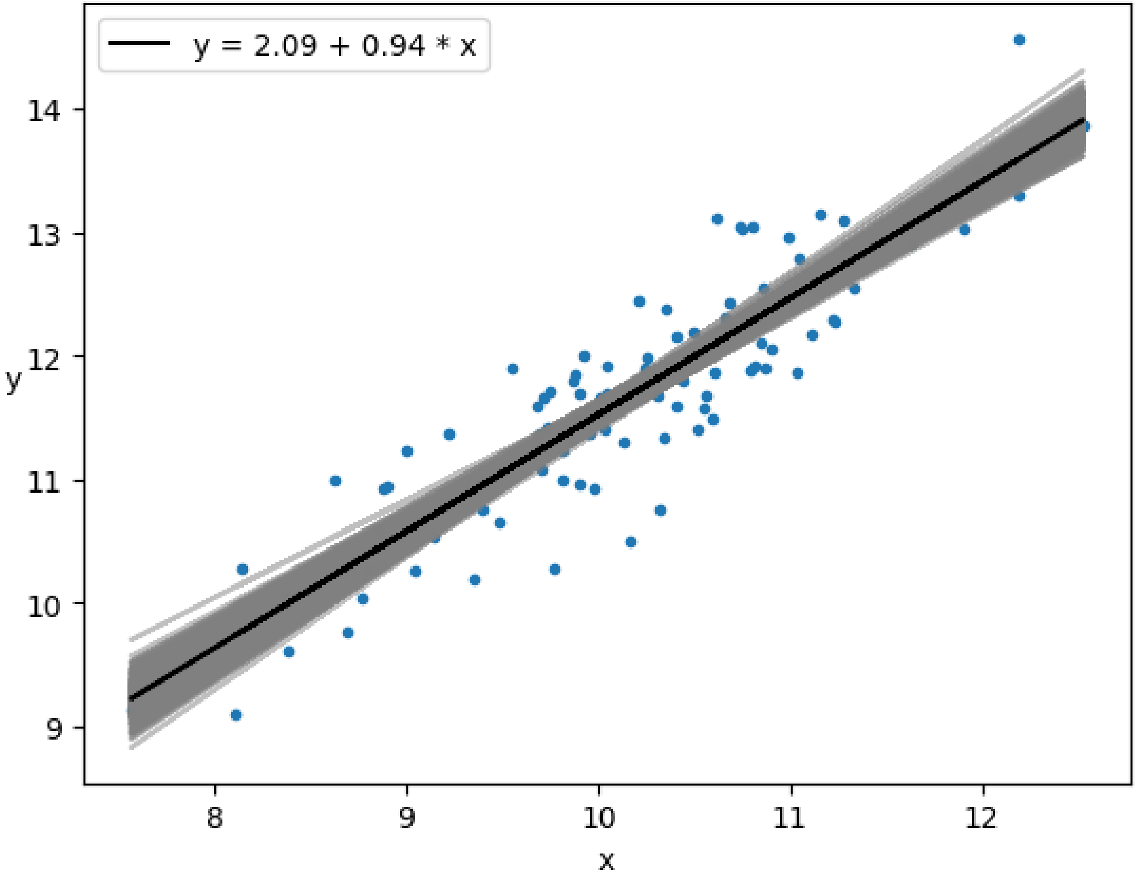


Define Bayesian model!

```
with pm.Model() as model_g:
    α = pm.Normal('α', mu=0, sigma=10)
    β = pm.Normal('β', mu=0, sigma=1)
    ϵ = pm.HalfCauchy('ϵ', 5)

    μ = pm.Deterministic('μ', α + β * x)
    y_pred = pm.Normal('y_pred', mu=μ,
                       sigma=ϵ, observed=y)

    idata_g = pm.sample(2000, tune=2000)
```

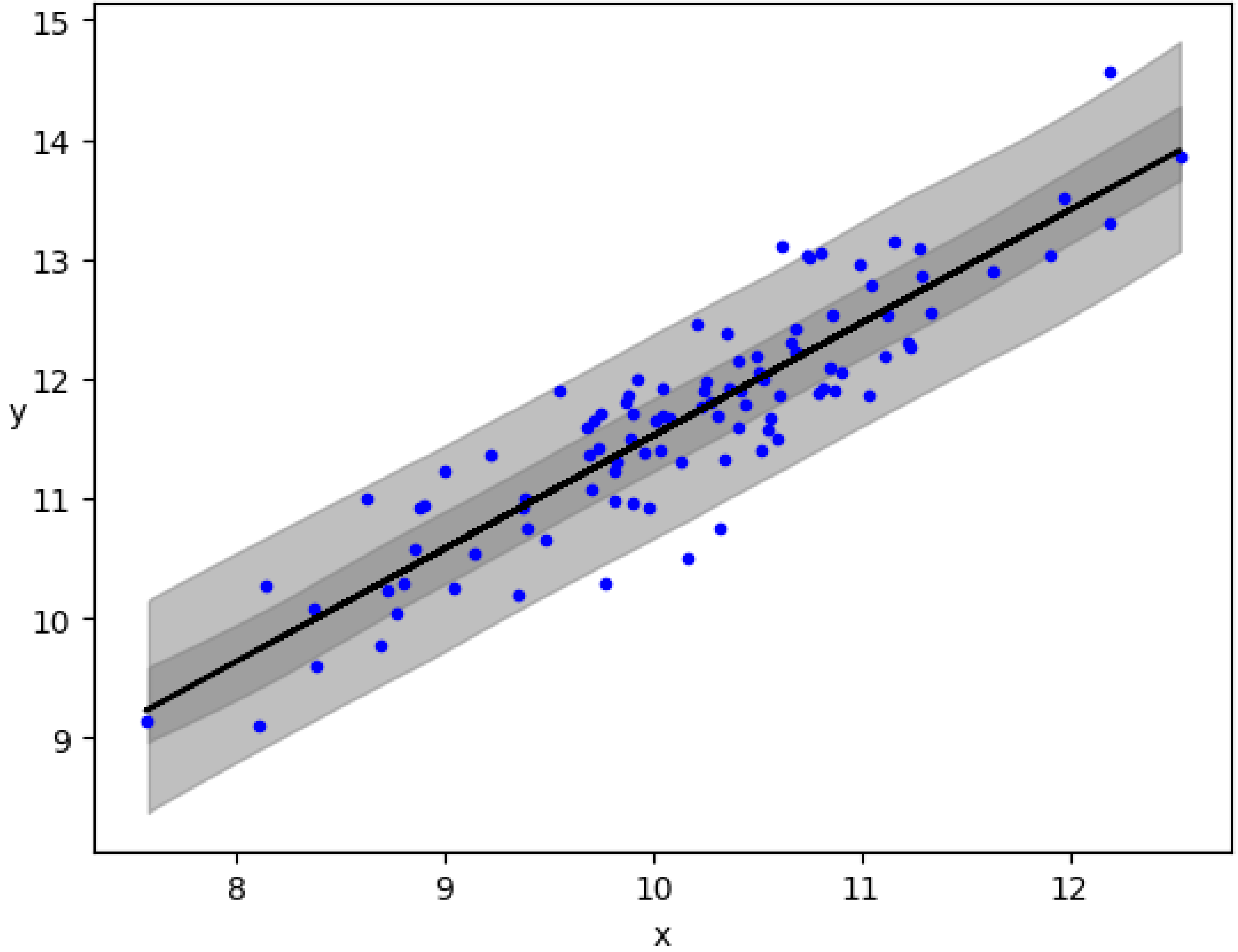
The outputs?





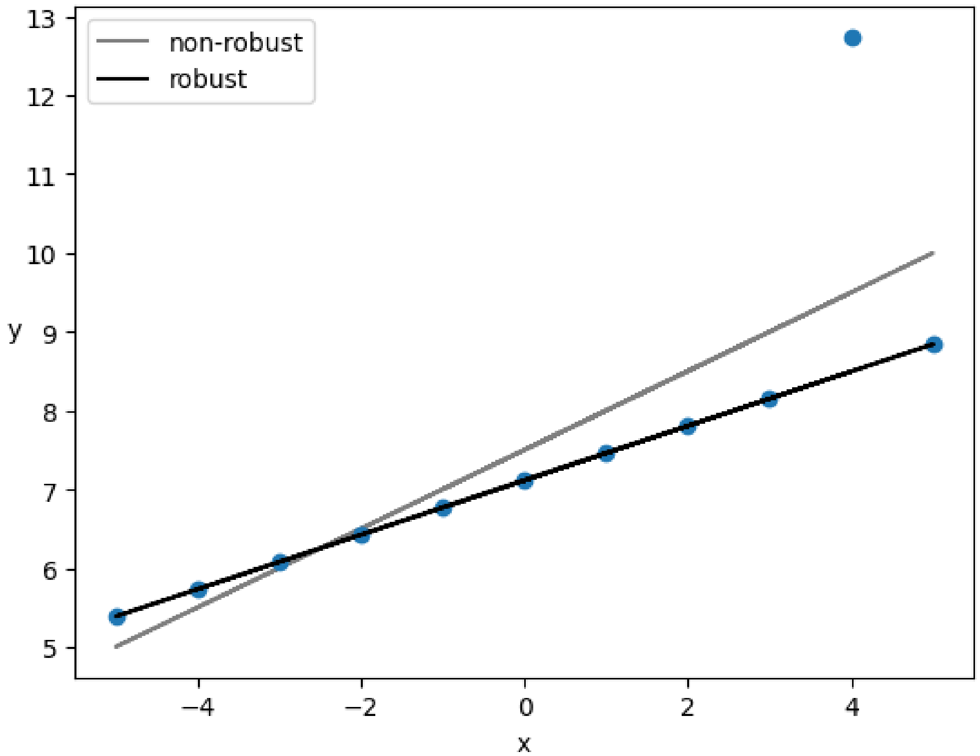
- The output of the Bayesian model will be draws for the slope and offset: families of lines consistent with data
- We can analyze their joint distribution
- And plot these lines

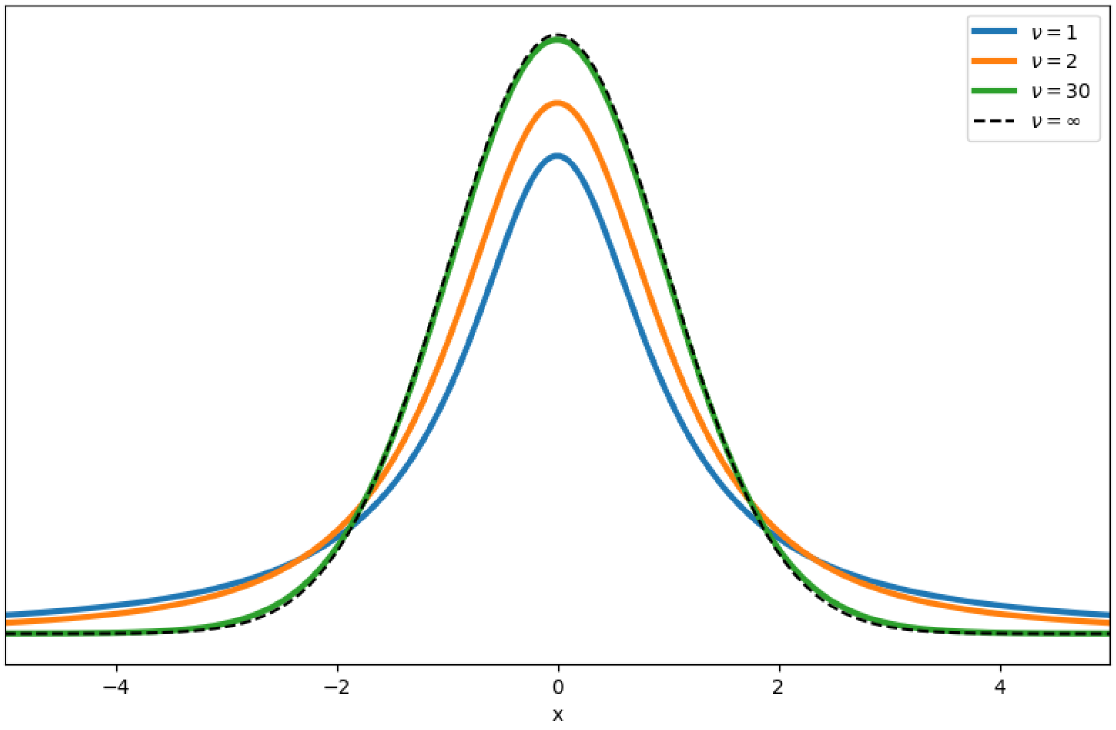
How do we decide if the “data is reasonable”?



- That’s what the posterior predictive checks are for!

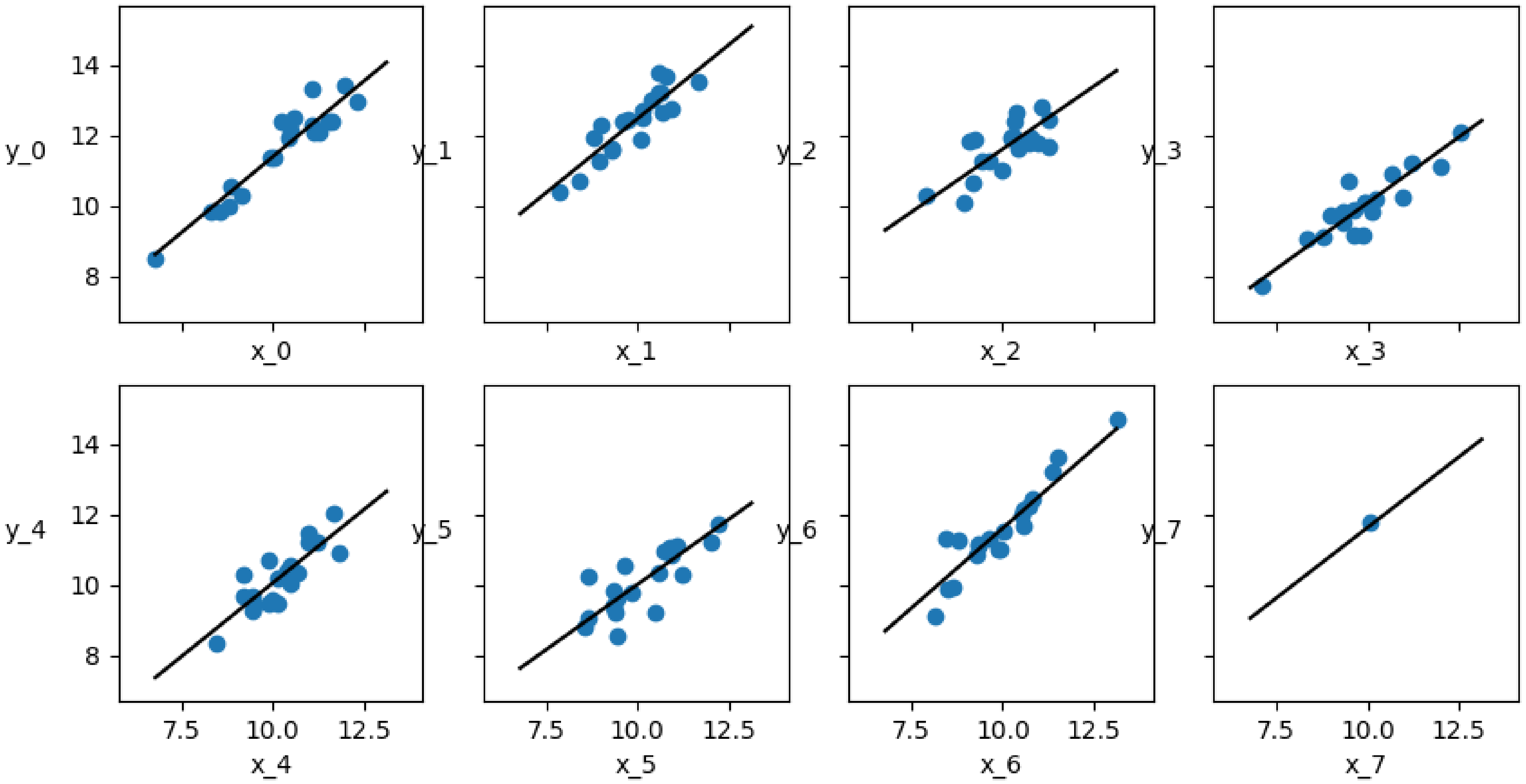
Bayesian methods for “impossible” problems





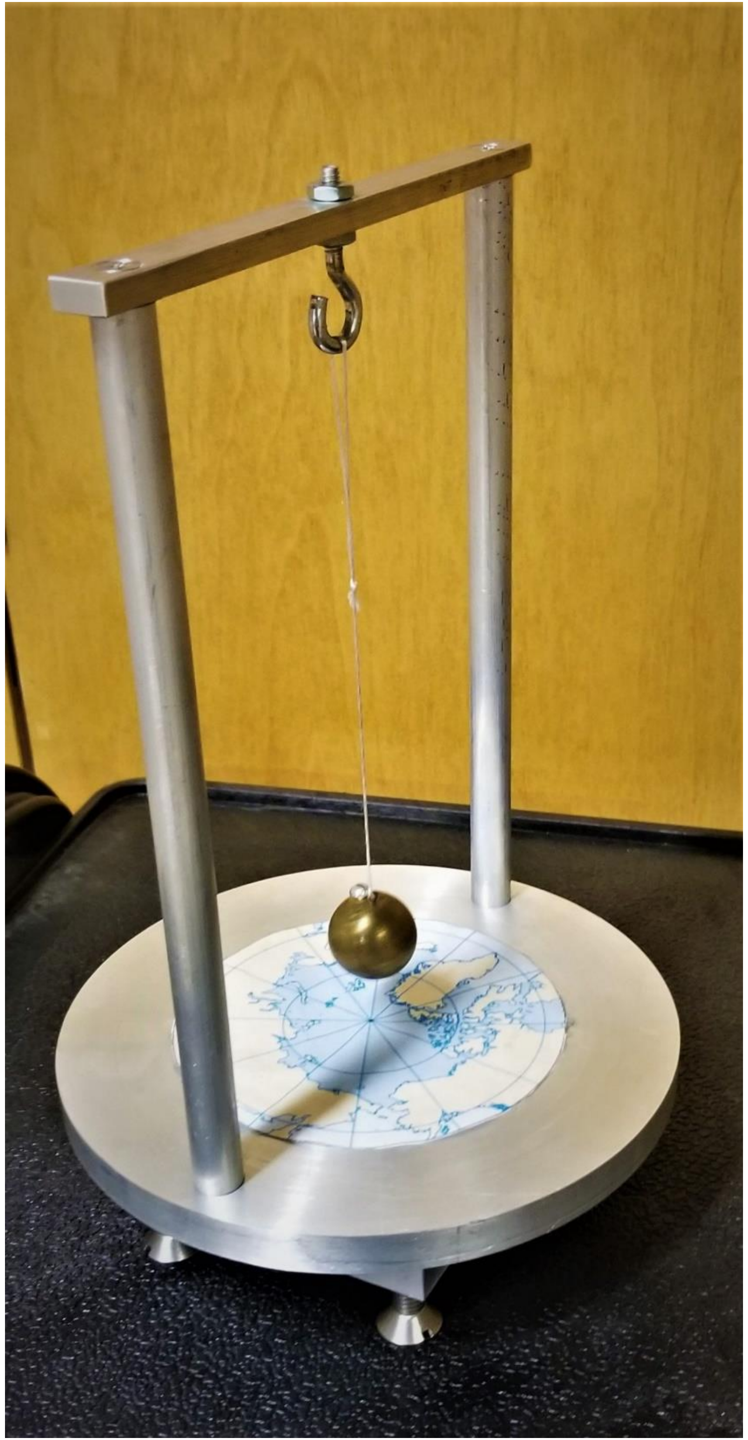

```
with pm.Model() as model_t:
    α = pm.Normal('α', mu=y_3.mean(), sigma=1)
    β = pm.Normal('β', mu=0, sigma=1)
    ϵ = pm.HalfNormal('ϵ', 5)
    ν_ = pm.Exponential('ν_', 1/29)
    ν = pm.Deterministic('ν', ν_ + 1)
    y_pred = pm.StudentT('y_pred', mu=α + β * x_3,sigma=ϵ, nu=ν, observed=y_3)
    idata_t = pm.sample(2000)
```

Bayesian methods for “impossible” problems



Least Square Fits

- We have some observational data
- And physical model expressed by formula
- All we have to do is to fit formula to the observations



Theory: For small angles, a simple pendulum follows a simple harmonic motion, where the period of a full swing back and forth (the time for one complete cycle) is given by the formula:

$T=2\pi\ Sqrt(L/g)$

- T is the period (time for one complete cycle, in seconds).
- L is the length of the pendulum
- g is the acceleration due to gravity (in $m/s2$).

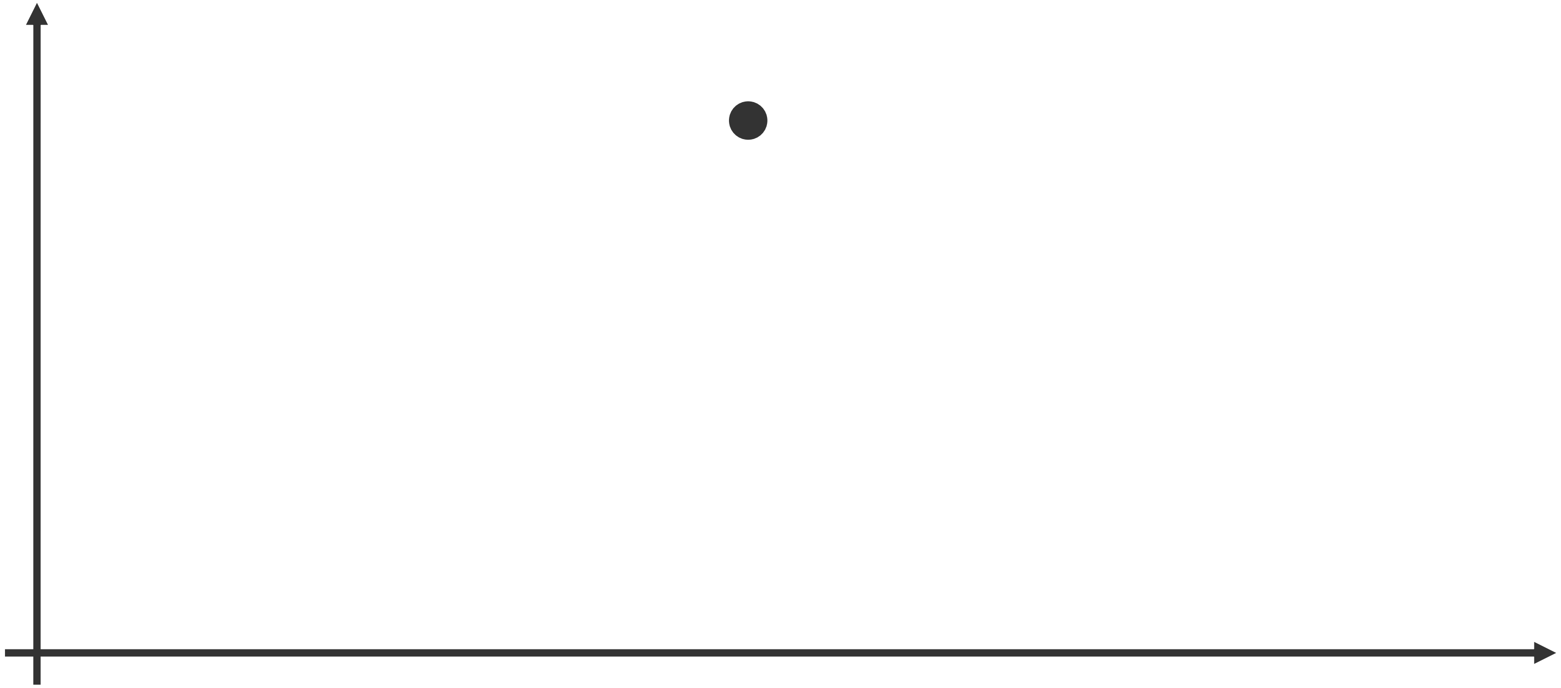
Procedure:

1. Measure the length of the pendulum (L) from the pivot point to the center of mass of the bob.
2. Displace the pendulum to a small angle (less than 15°) to ensure that the motion approximates simple harmonic motion and release it.
3. Use the stopwatch to measure the time it takes for the pendulum to complete a number of oscillations.
4. To reduce error, measure the time for multiple oscillations (say, 10 or 20) and then divide by the number of oscillations to find the average period (T).
5. Repeat a few times and average to minimize random errors.

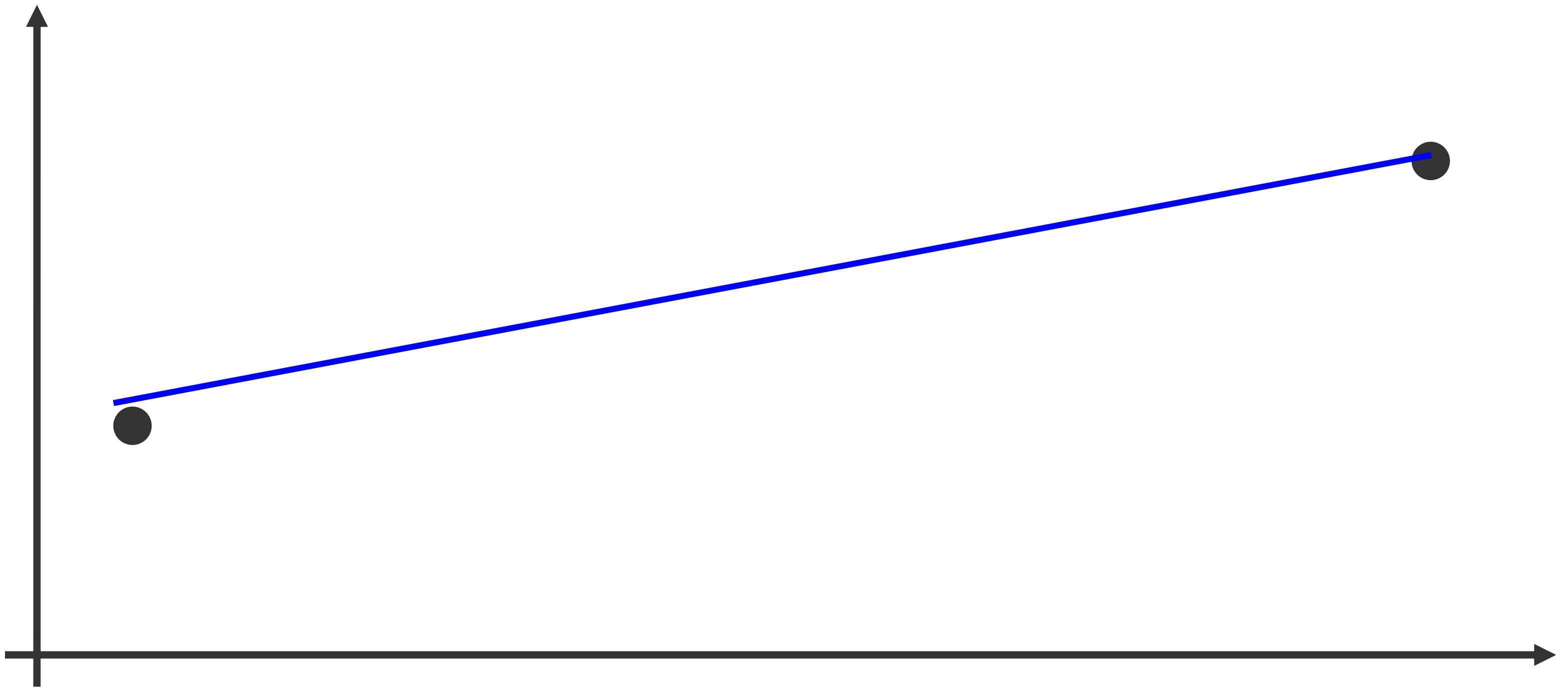
Pendulum example

- Let's assume that we have a ruler, balance, and stop watch
- However, the measurements of g gives us 12 m/s^2. We know that the true value is 9.8 m/s^2
- How can we analyze the uncertainties?

Physics vs. data science

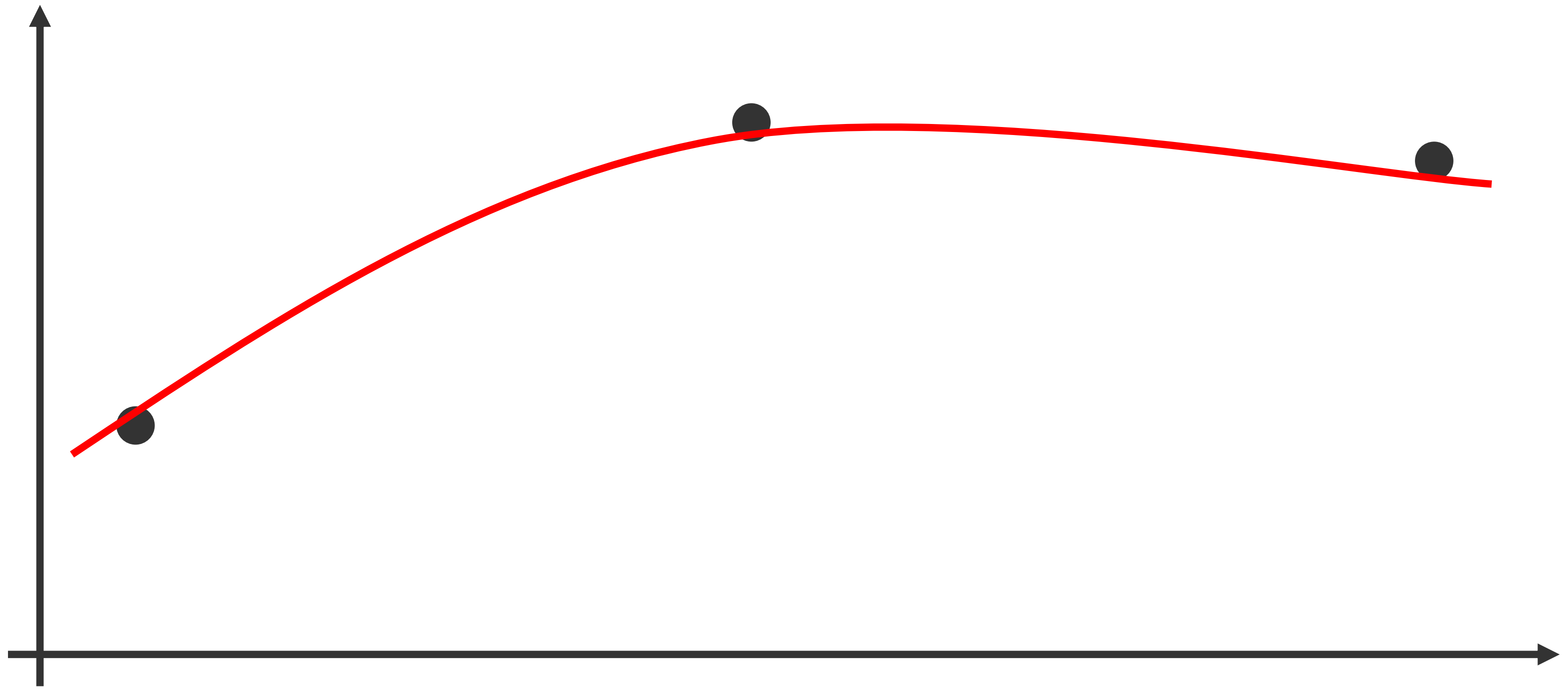


Physics vs. data science



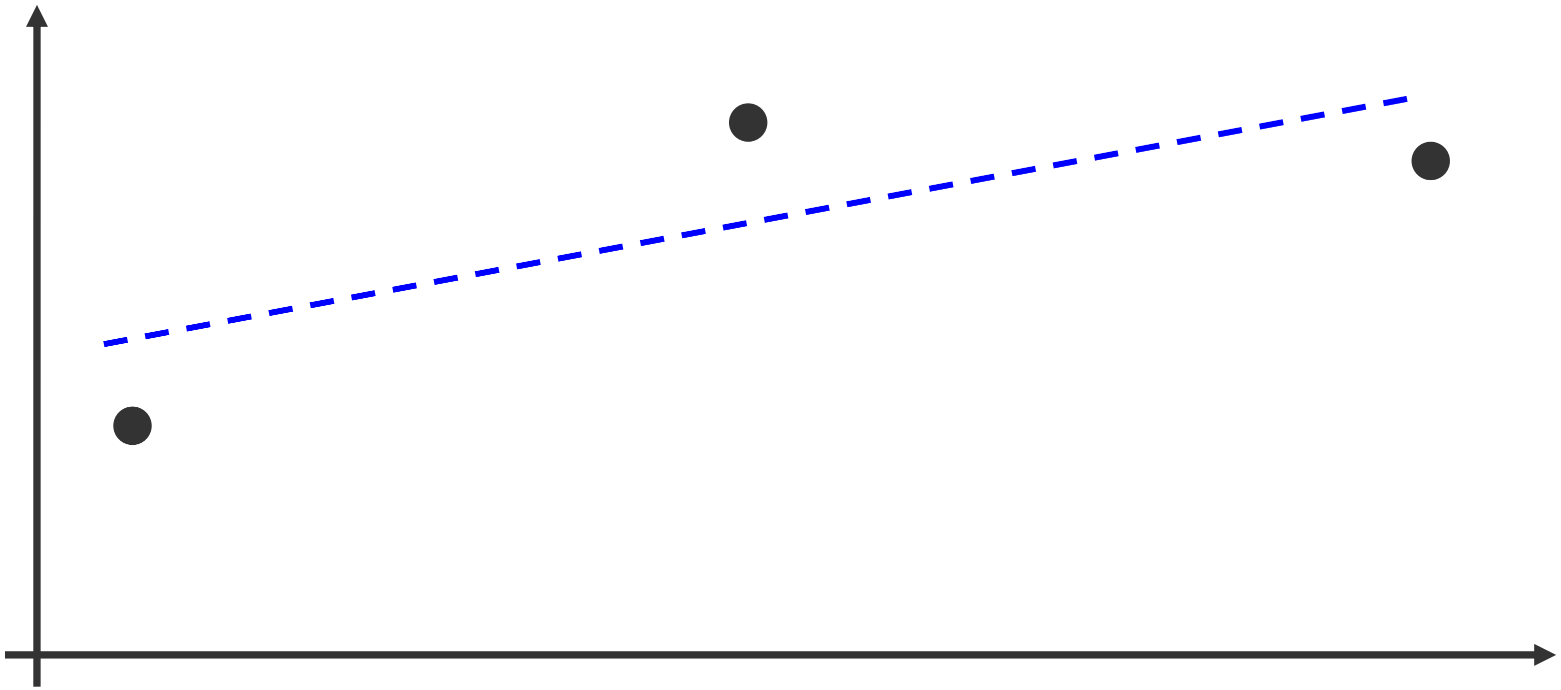
- If we have 2 data points, we "naturally" use linear model
- What should we use if we have three data points? Parabola or linear?
- What if we have one data point?

Physics vs. data science



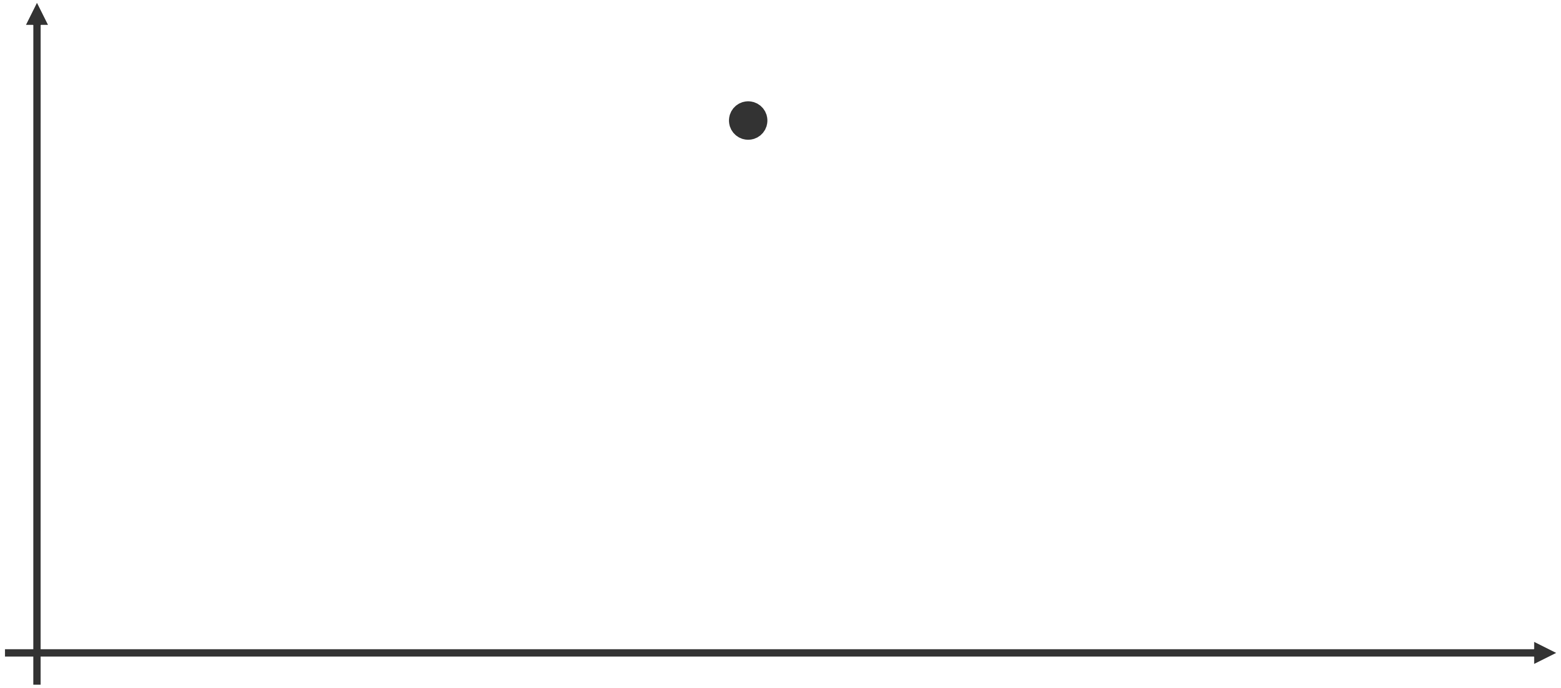
- If we have 2 data points, we “naturally” use linear model
- What should we use if we have three data points? Parabola or linear?
- What if we have one data point?

Physics vs. data science



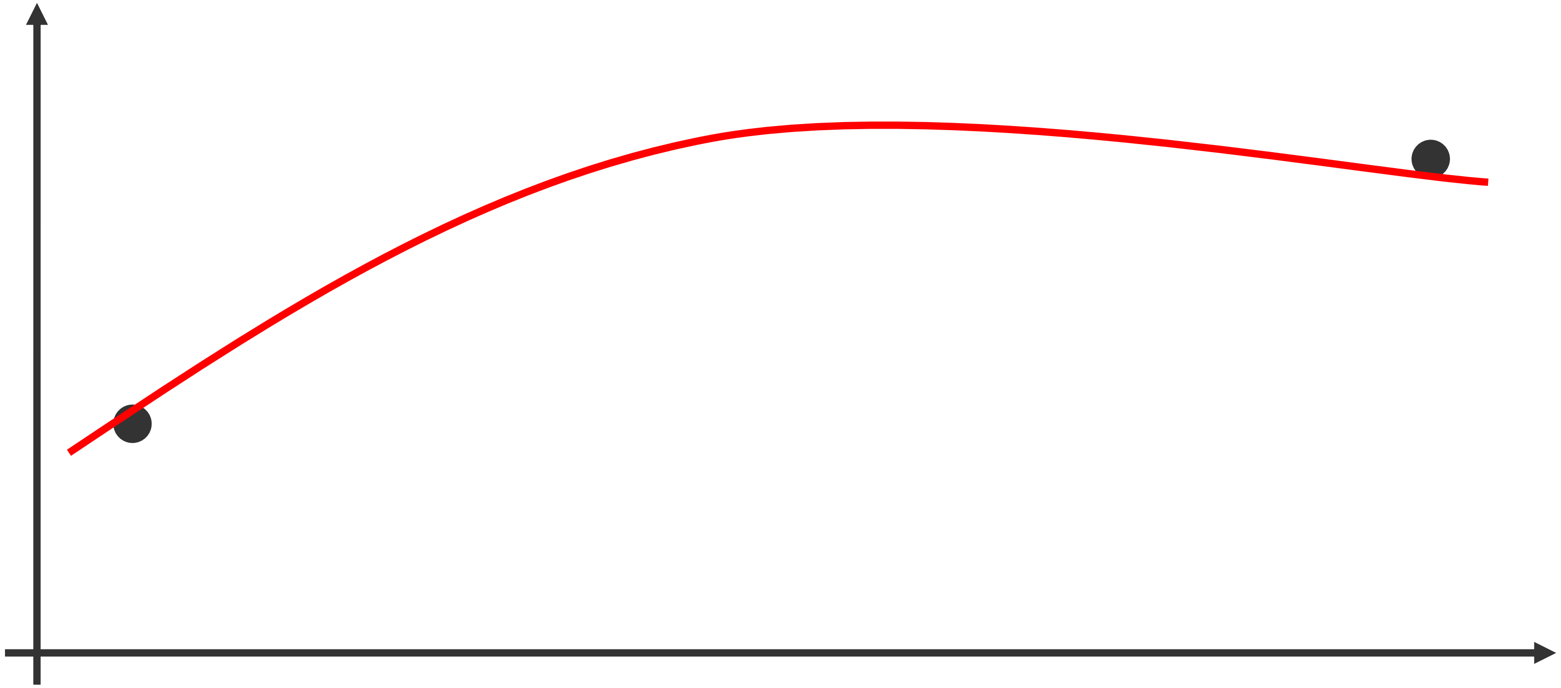
- If we have 2 data points, we "naturally" use linear model
- What should we use if we have three data points? Parabola or linear?
- What if we have one data point?

Physics vs. data science



- If we have 2 data points, we "naturally" use linear model
- What should we use if we have three data points? Parabola or linear?
- What if we have one data point?

Physics vs. data science



- If we have 2 data points, we "naturally" use linear model
- What should we use if we have three data points? Parabola or linear?
- What if we have one data point?