THE UNIVERSITY of TENNESSEE KNOXVILLE

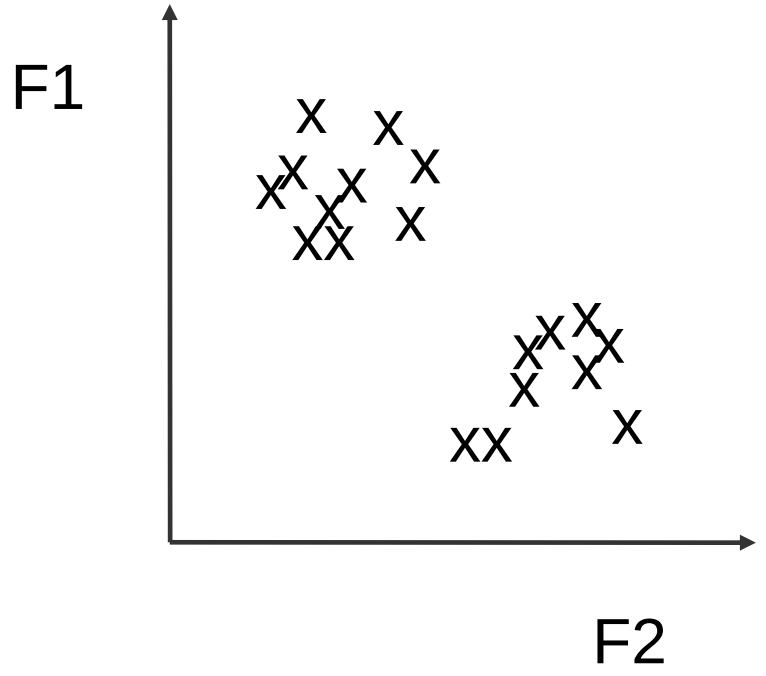
Lecture 11: Clustering on Imaging and Spectroscopic Data

Instructor: Sergei V. Kalinin

Clustering

- The process of grouping a set of objects into classes of similar objects
  - Objects within a cluster should be similar.
  - Objects from different clusters should be dissimilar.
- The most common form of *unsupervised learning*

- Given a set of data points, each described by a set of attributes, find clusters such that:
  - Inter-cluster similarity is maximized
  - Intra-cluster similarity is minimized
- Requires the definition of a similarity measure



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

CS276: Information Retrieval and Web Search Pandu Nayak and Prabhakar Raghavan

Taxon Cetartiodactyla

Taxon Equidae

Clustering: Data Structures

- Hierarchical clustering
- K-means clustering
- Gaussian Mixture Models
- Density-based clustering
- Spectral clustering

Data matrix (two modes)

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

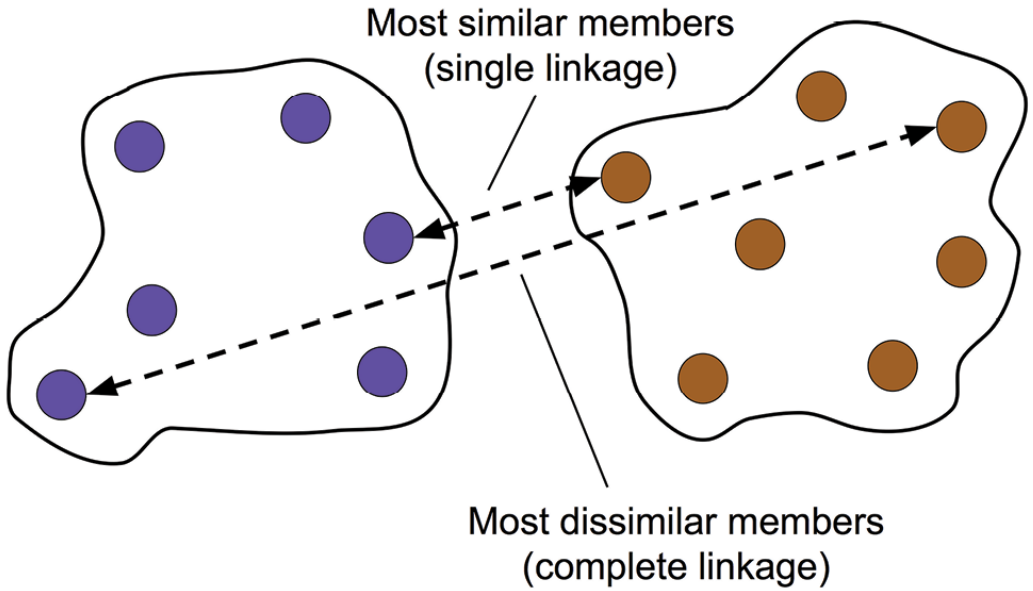
Dissimilarity matrix (one mode)

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

From: Han and Kamber, Data Mining: Concepts and Techniques

Distances

Figure by S. Raschka



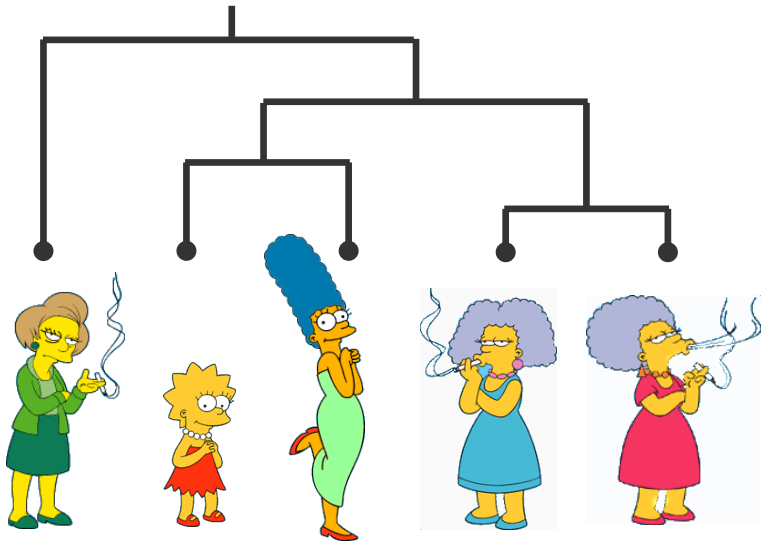
- Have a distance measure on pairs of objects, $d(x, y)$:
- Single linkage: $\text{dist}(\text{A}, \text{B}) = \min_{\text{x}\in A, x'\in B} \text{d}(\text{x}, \text{x}')$
- Complete linkage: $\text{dist}(\text{A}, \text{B}) = \max_{\text{x}\in A, x'\in B} \text{d}(\text{x}, \text{x}')$
- Average linkage: $\text{dist}(\text{A}, \text{B}) = \text{average}_{\text{x}\in A, x'\in B} \text{d}(\text{x}, \text{x}')$
- Ward's method $\text{dist}(\text{A}, \text{B}) = \frac{|A|\,|B|}{|A|+|B|} \|\text{mean}(A) - \text{mean}(B)\|^2$

Slide credit: Maria-Florina Balcan

Hierarchical Clustering

The number of dendrograms with n leafs $= (2n - 3)!/[(2^{(n-2)})(n-2)!]$

Number of Leafs	Number of Possible Dendrograms
2	1
3	3
4	15
5	105
...	…
10	34,459,425



Since we cannot test all possible trees, we will have to heuristic search of all possible trees. We could do this..

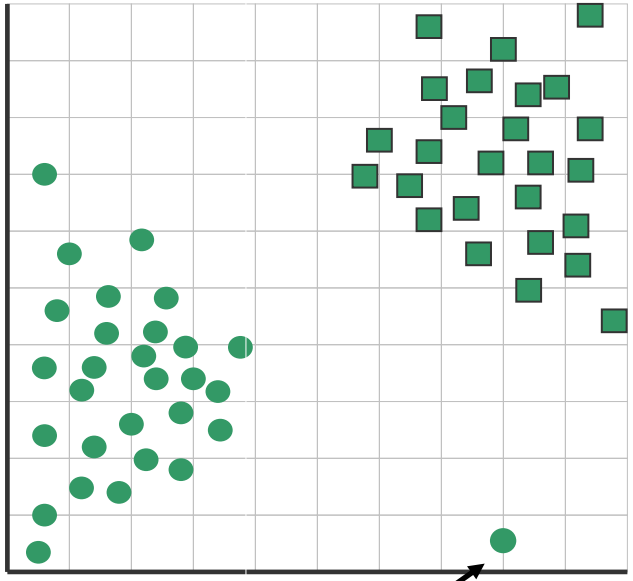
Bottom-Up (agglomerative): Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

Top-Down (divisive): Starting with all the data in a single cluster, consider every possible way to divide the cluster into two. Choose the best division and recursively operate on both sides.

Slide from Eamonn Keogh, from lecture by Carla Brodley, Tufts University

One potential use of a dendrogram: detecting outliers

The single isolated branch is suggestive of a data point that is very different to all others



Outlier

Slide from Eamonn Keogh, from lecture by Carla Brodley, Tufts University

Slide from Eamonn Keogh, from lecture by Carla Brodley, Tufts University

Hierarchical Clustering Methods Summary

- No need to specify the number of clusters in advance
- Hierarchical nature maps nicely onto human intuition for some domains
- They do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects
- Like any heuristic search algorithms, local optima are a problem
- Interpretation of results is (very) subjective

Partition Algorithm 1: k-means

1. Decide on a value for *k*.

2. Initialize the *k* cluster centers (randomly, if necessary).

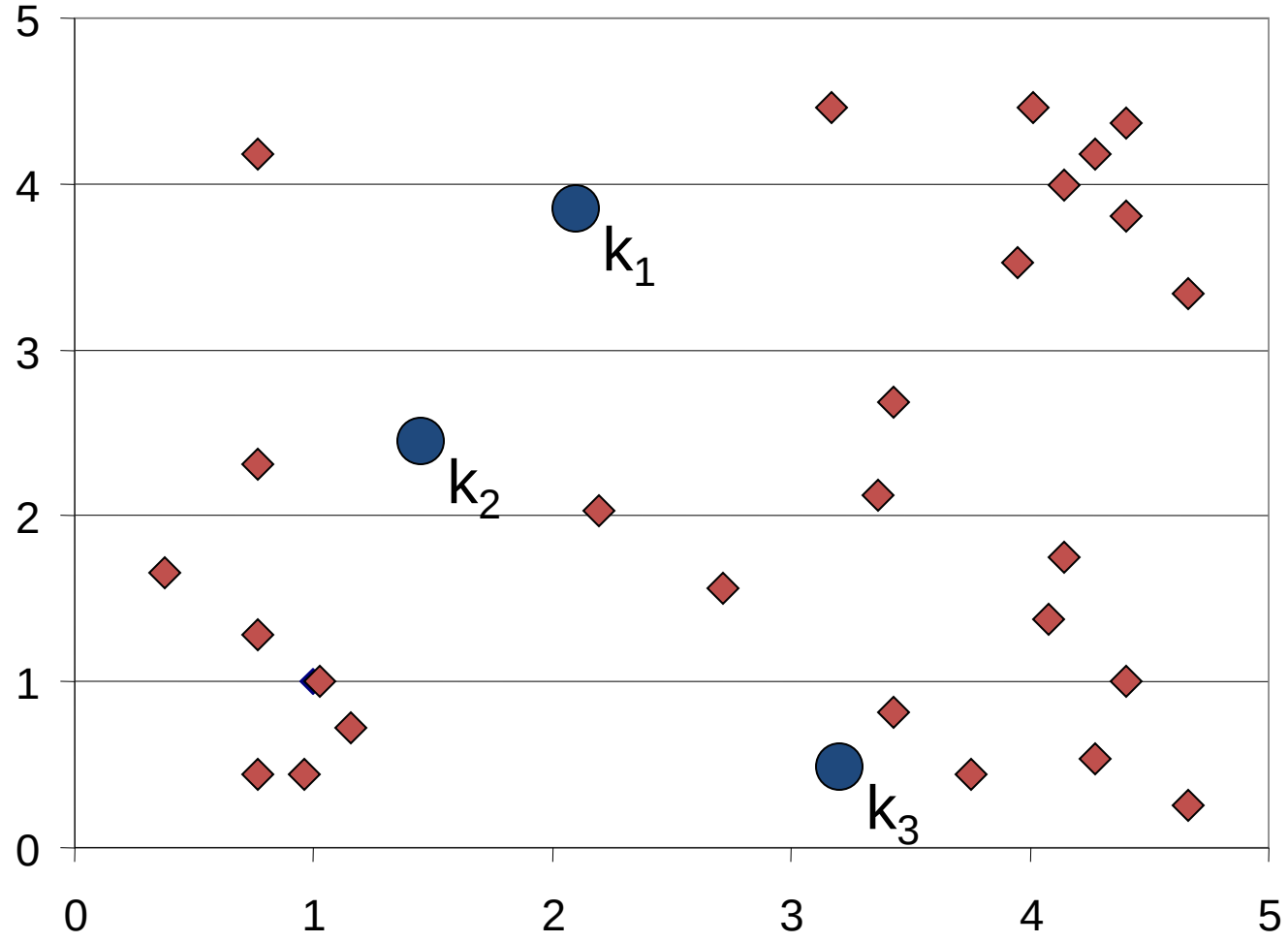
3. Decide the class memberships of the *N* objects by assigning them to the nearest cluster center.

4. Re-estimate the *k* cluster centers, by assuming the memberships found above are correct.

5. If none of the *N* objects changed membership in the last iteration, exit. Otherwise goto 3.

K-means Clustering: Step 1

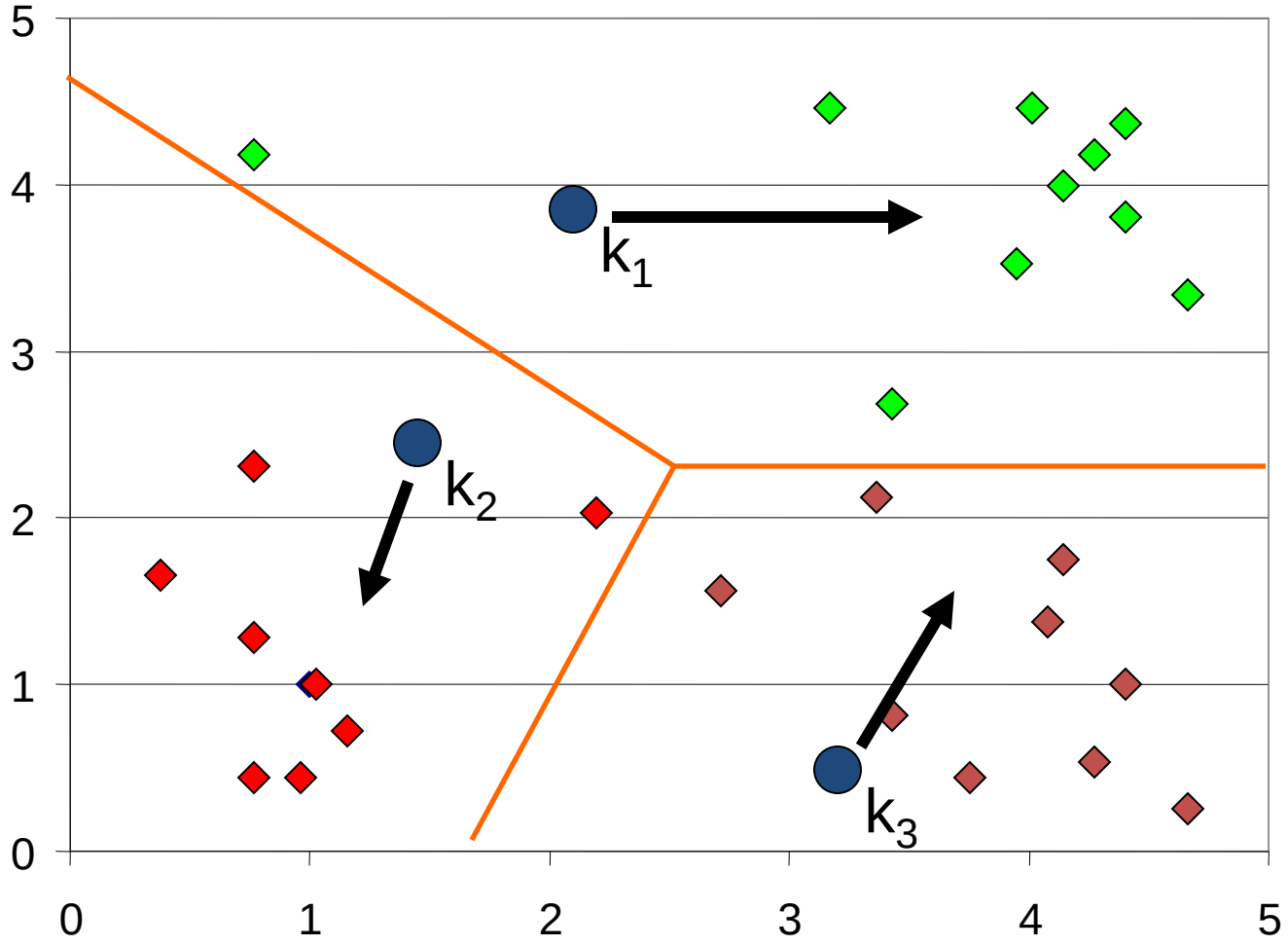
Algorithm: k-means, Distance Metric: Euclidean Distance



Slide from Eamonn Keogh, from lecture by Carla Brodley, Tufts University

K-means Clustering: Step 2

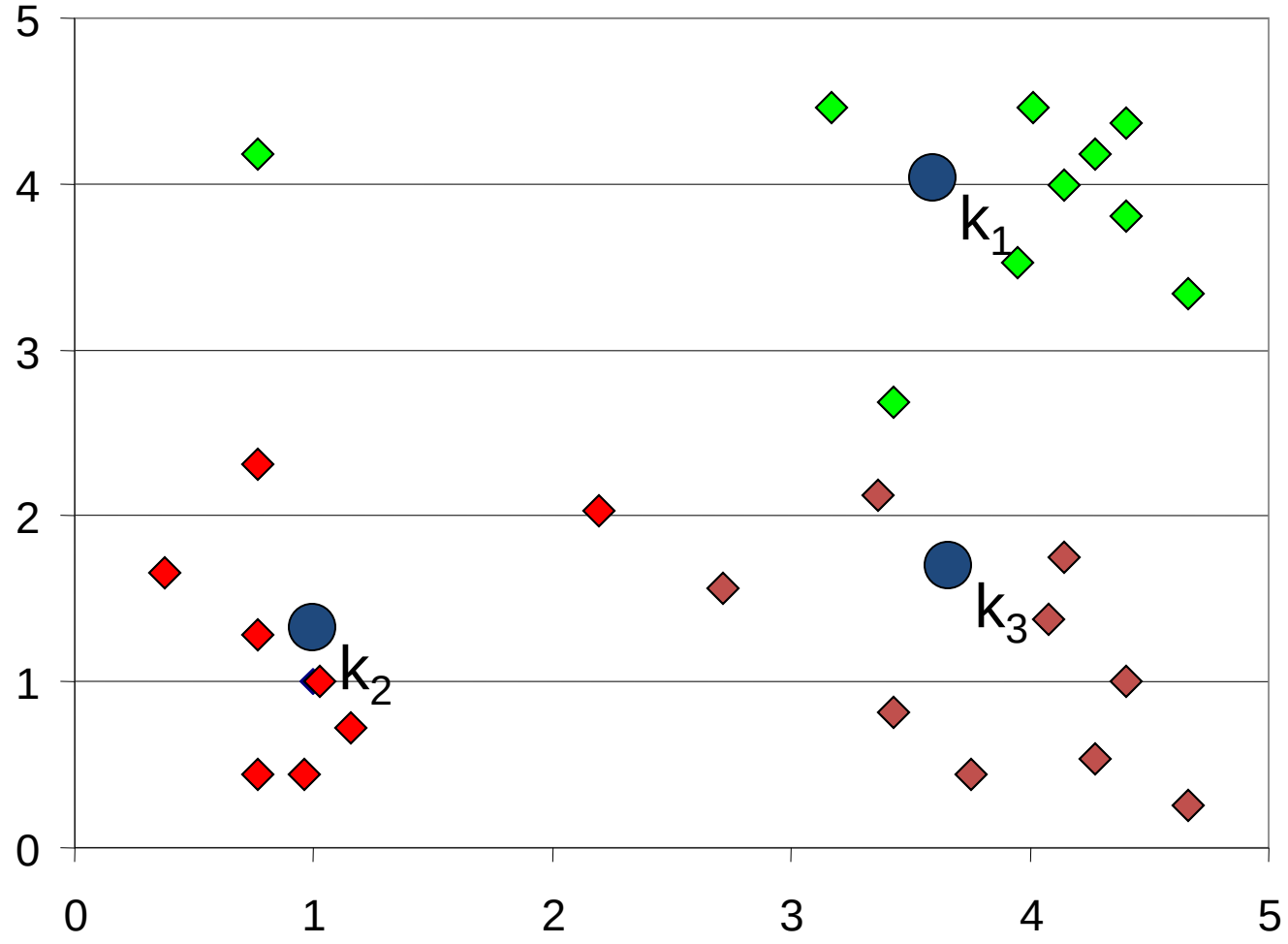
Algorithm: k-means, Distance Metric: Euclidean Distance



Slide from Eamonn Keogh, from lecture by Carla Brodley, Tufts University

K-means Clustering: Step 3

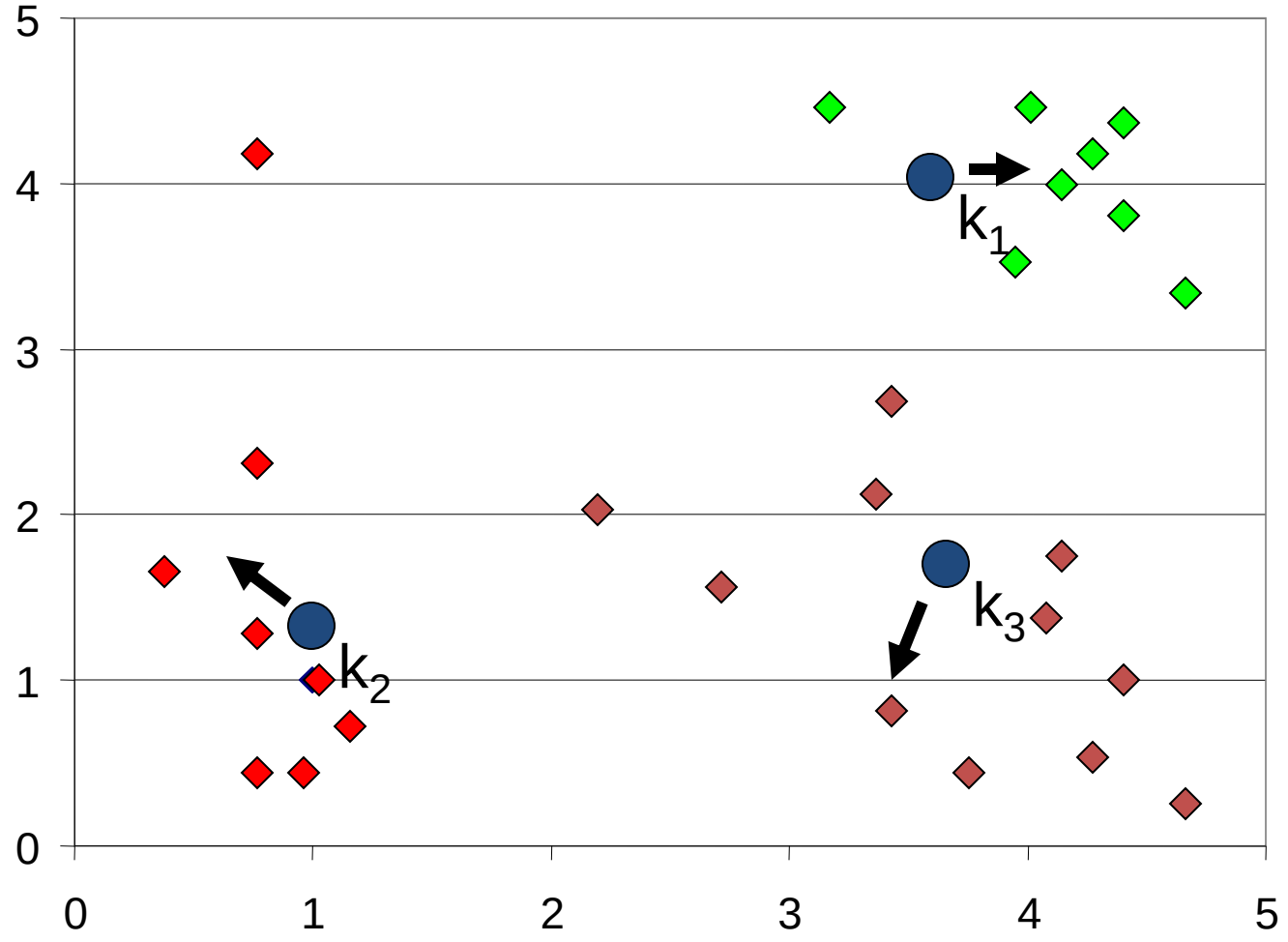
Algorithm: k-means, Distance Metric: Euclidean Distance



Slide from Eamonn Keogh, from lecture by Carla Brodley, Tufts University

K-means Clustering: Step 4

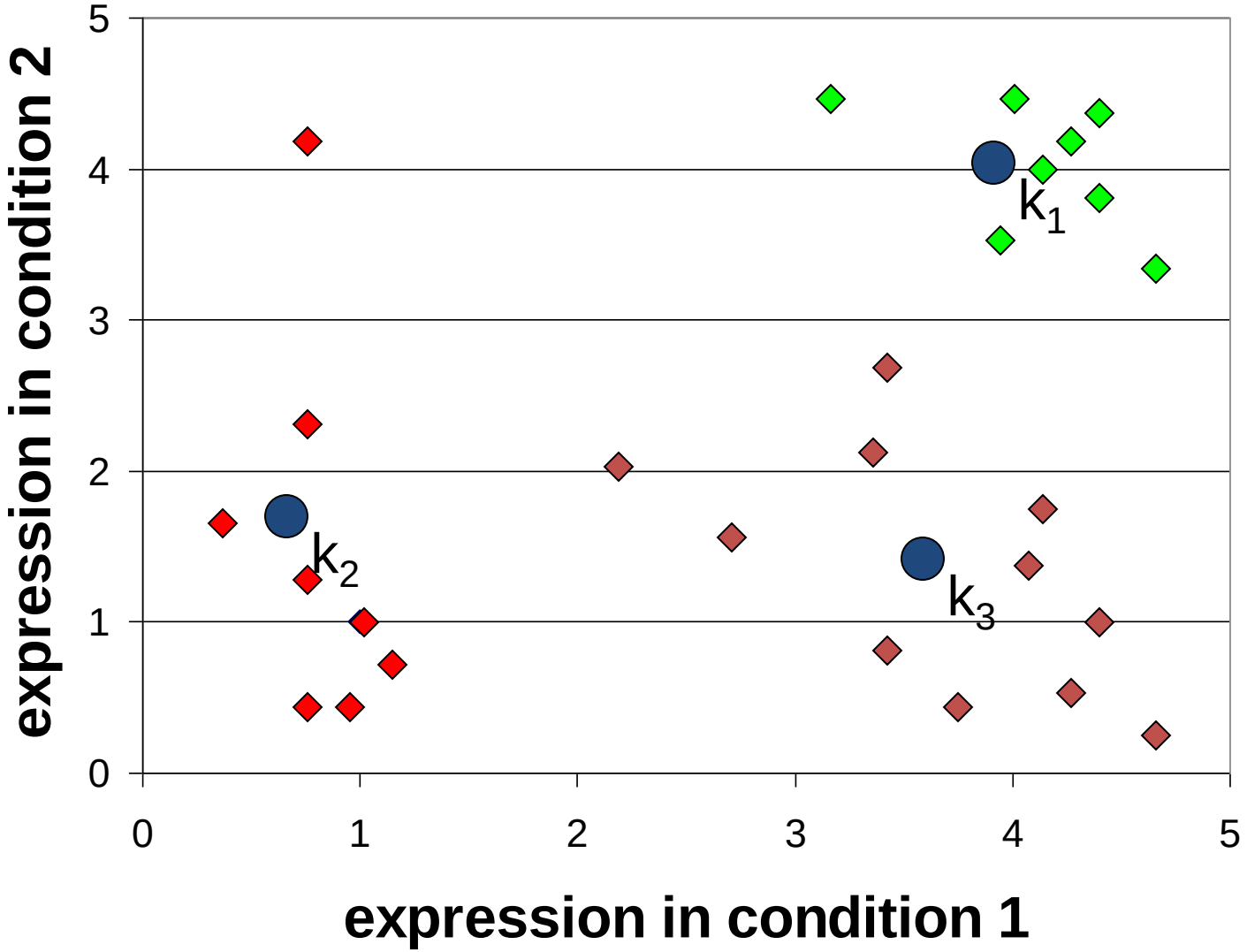
Algorithm: k-means, Distance Metric: Euclidean Distance



Slide from Eamonn Keogh, from lecture by Carla Brodley, Tufts University

K-means Clustering: Step 5

Algorithm: k-means, Distance Metric: Euclidean Distance



Slide from Eamonn Keogh, from lecture by Carla Brodley, Tufts University

Summary of k-means clustering

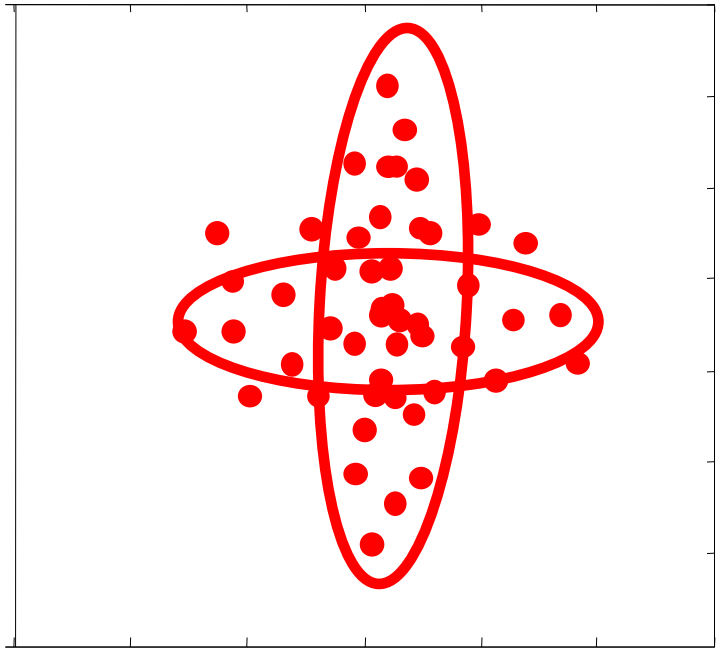
- **Strengths**
  - *Relatively efficient*: $O(tkn)$, where n is number of objects, k is number of clusters, and t is number of iterations. Normally, $k, t << n$.
  - Often terminates at a local optimum

- **Weakness**
  - Applicable only when mean is defined (what about categorical data)?
  - Need to specify k, the number of clusters, in advance
  - Unable to handle noisy data and outliers
  - Not suitable to discover clusters with non-convex shapes
  - Scales matter

Mixture of Gaussians

K-means algorithm

- Assigned each example to exactly one cluster
- What if clusters are overlapping?
  - Hard to tell which cluster is right
  - Maybe we should try to remain uncertain
- Used Euclidean distance
- What if cluster has a non-circular shape?



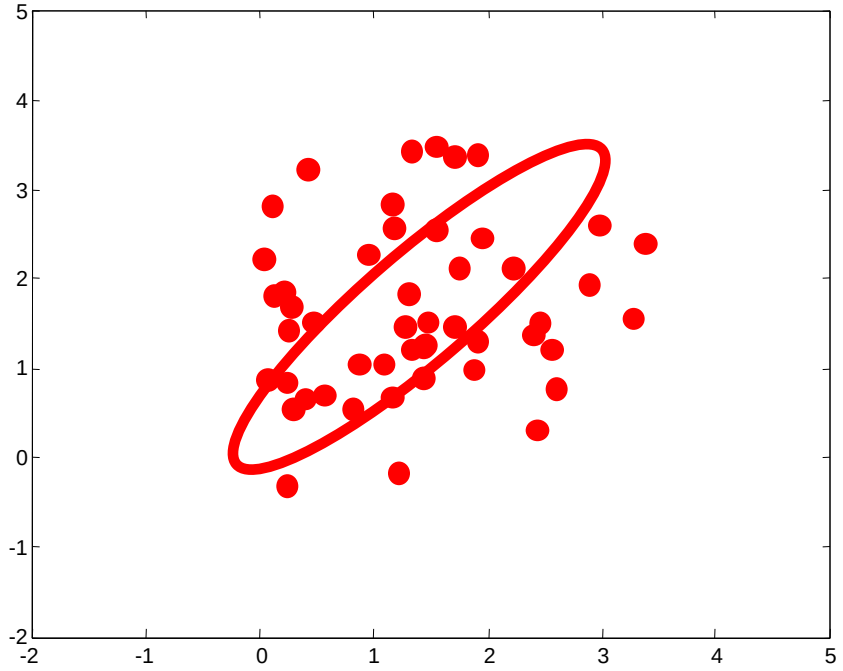
Gaussian mixture models

- Clusters modeled as Gaussian distributions
- EM algorithm: assign data to cluster with some *probability*

(adapted from) Prof. Alexander Ihler

Multivariate Gaussian Model

$$\mathcal{N}(\underline{x}\ ;\ \underline{\mu}, \Sigma) = \frac{1}{(2\pi)^{d/2}} |\Sigma|^{-1/2} \exp\left\{ -\frac{1}{2} (\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu}) \right\}$$



Maximum Likelihood estimates

$$\hat{\mu} = \frac{1}{N} \sum_i x^{(i)}$$

$$\hat{\Sigma} = \frac{1}{N} \sum_i (x^{(i)} - \hat{\mu})^T (x^{(i)} - \hat{\mu})$$

We model each cluster using Gaussian distribution

(adapted from) Prof. Alexander Ihler

Expectation Maximization: E-Step

- Initialize parameters of each cluster: mean μ_c, Covariance Σ_c, size π_c

- **E-step ("Expectation")**
  - For each datum (example) x_i,
  - Compute r_{ic}, the probability that it belongs to cluster c
    - Compute its probability under model c
    - Normalize to sum to one (over clusters c)

$$r_{ic} = \frac{\pi_c \mathcal{N}(x_i \; ; \; \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} \mathcal{N}(x_i \; ; \; \mu_{c'}, \Sigma_{c'})}$$

  - If x_i is very likely under the c^{th} Gaussian, it gets high weight
  - Denominator just makes probabilities to sum to one

(adapted from) Prof. Alexander Ihler

Expectation Maximization: M-Step

- Start with assignment probabilities r_{ic}
- Update parameters: mean μ_c, Covariance Σ_c, "size" π_c

- M-step ("Maximization")
  - For each Gaussian cluster x_c,
  - Update its parameters using the (weighted) data points

$$N_c = \sum_i r_{ic}$$

Total responsibility allocated to cluster c

$$\pi_c = \frac{N_c}{N}$$

Fraction of total assigned to cluster c

$$\mu_c = \frac{1}{N_c} \sum_i r_{ic} x_i$$

Weighted mean of assigned data

$$\Sigma_c = \frac{1}{N_c} \sum_i r_{ic} (x_i - \mu_c)^T (x_i - \mu_c)$$

Weighted covariance of assigned data (use new weighted means here)

(adapted from) Prof. Alexander Ihler

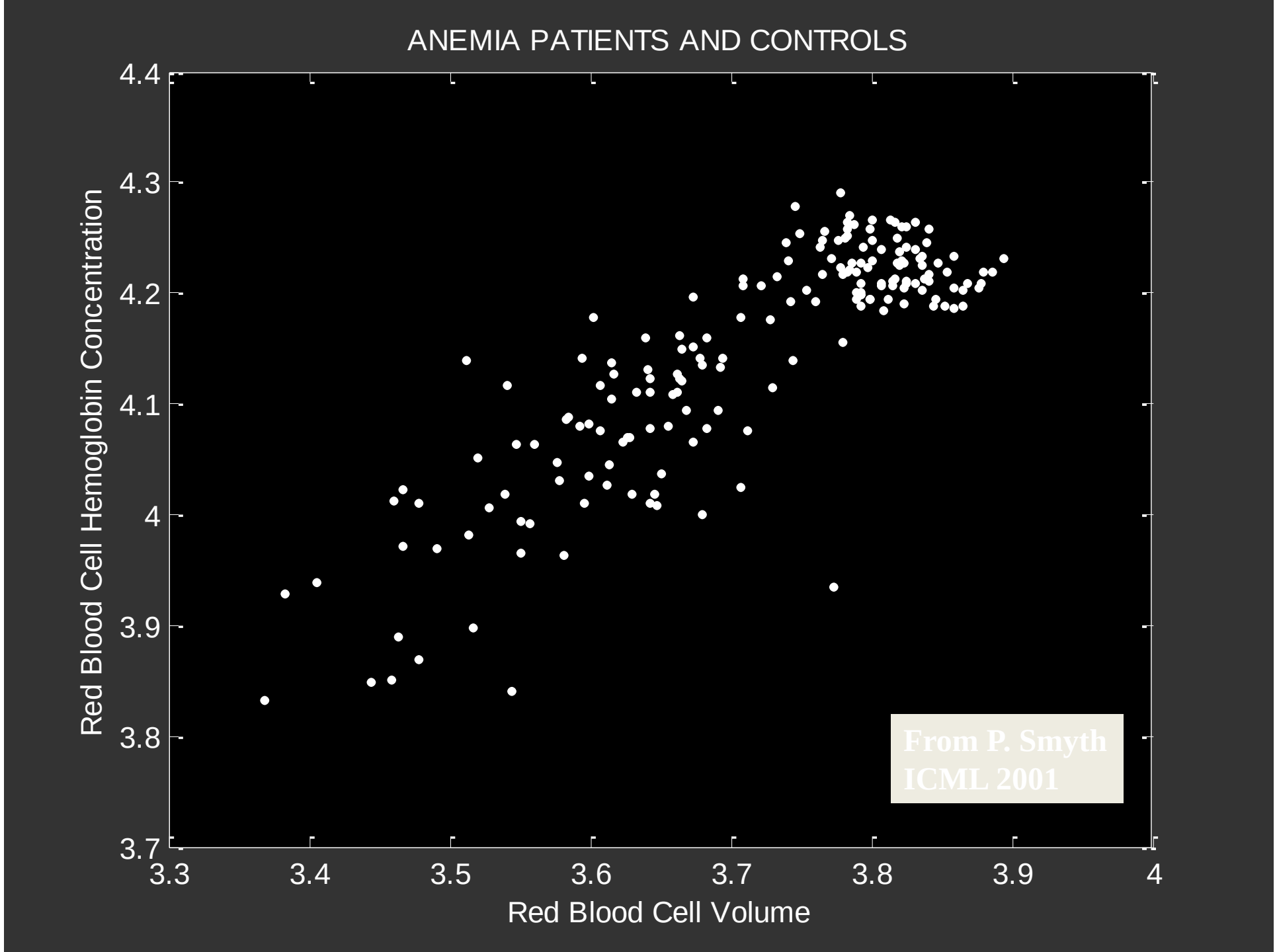
Expectation Maximization

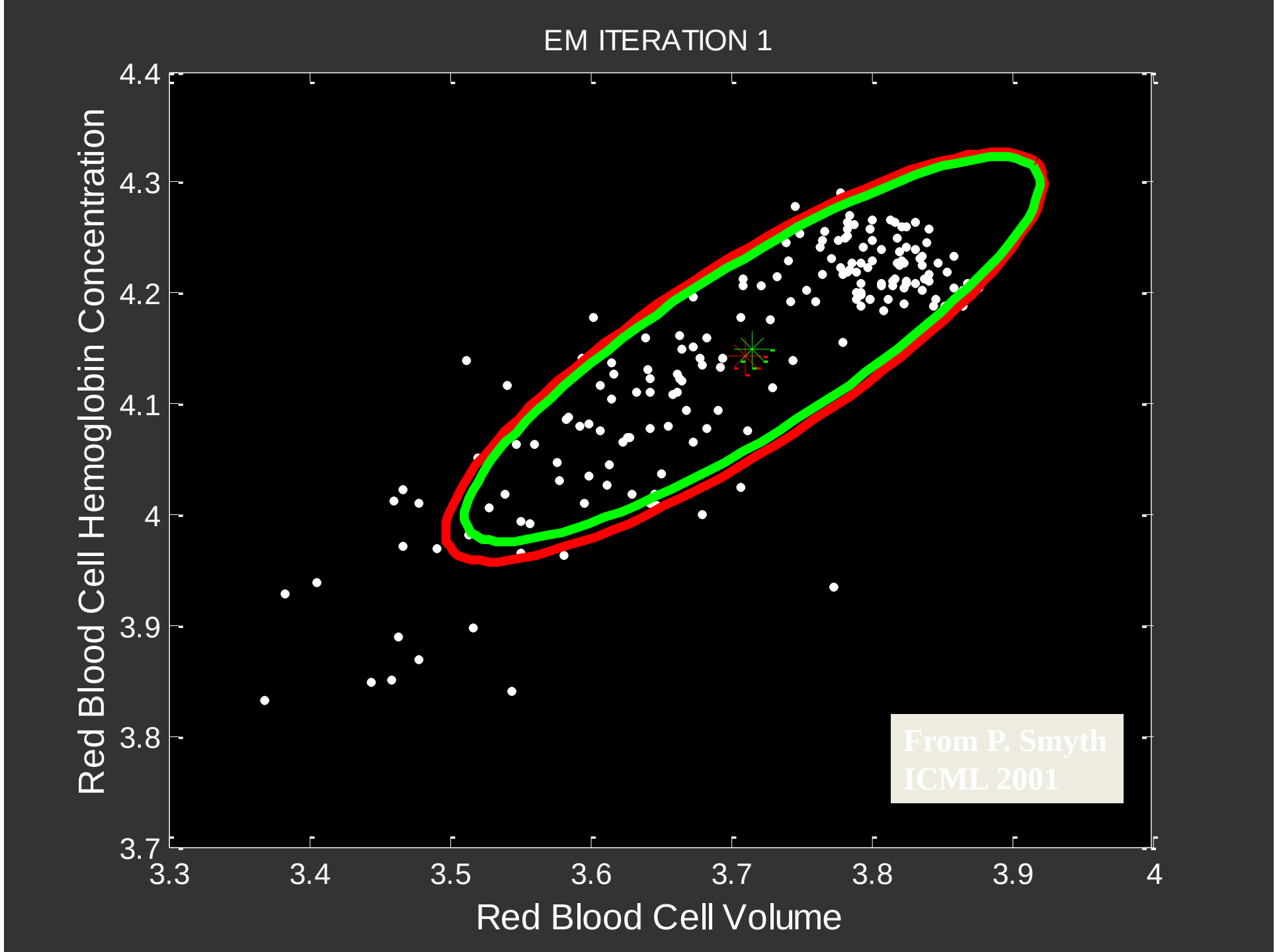
- Each step increases the log-likelihood of our model

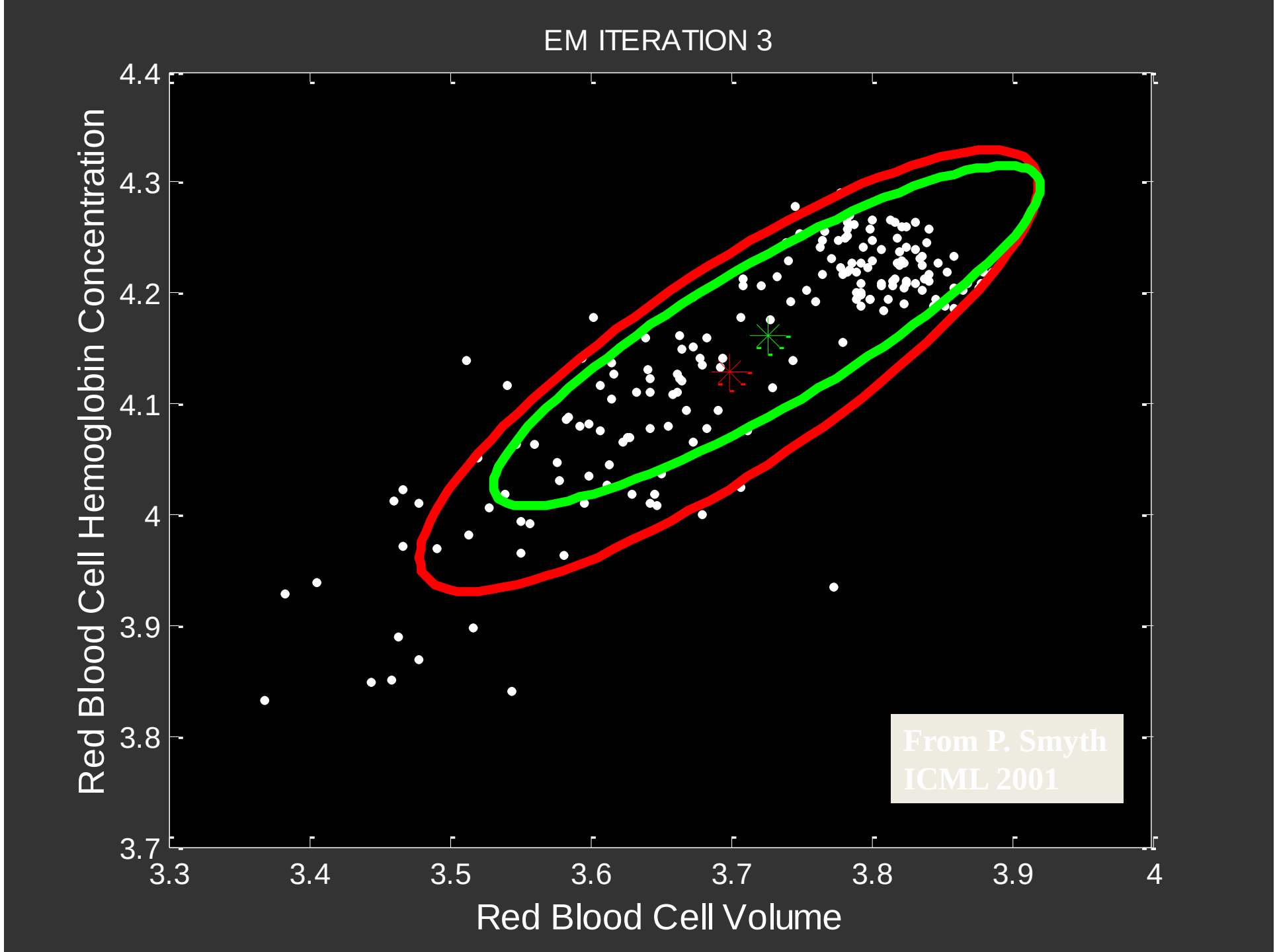
$$\log p(\underline{X}) = \sum_i \log \left[\sum_c \pi_c \, \mathcal{N}(x_i \; ; \; \mu_c, \Sigma_c) \right]$$

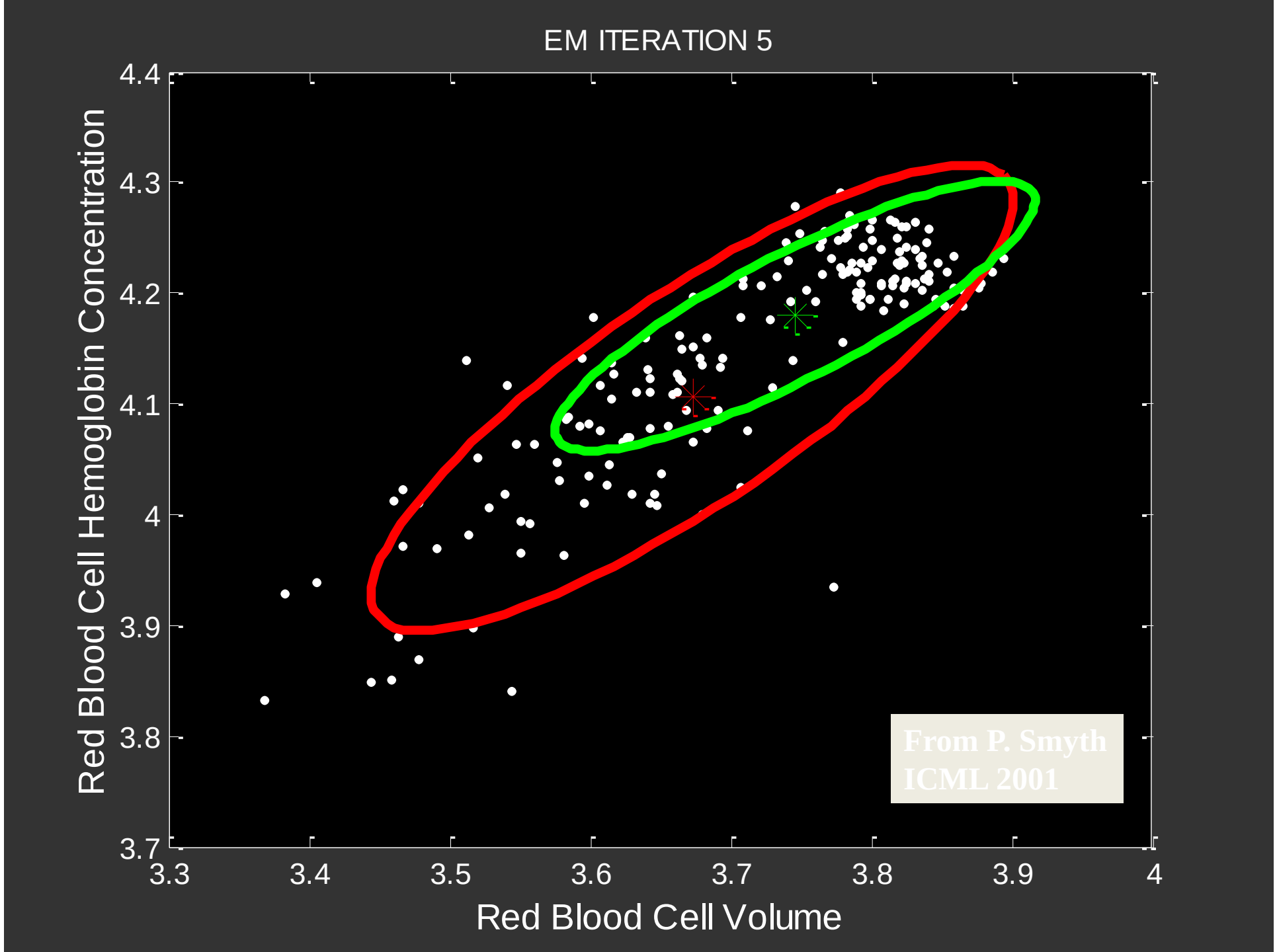
- Iterate until convergence
  - Convergence guaranteed – another ascent method

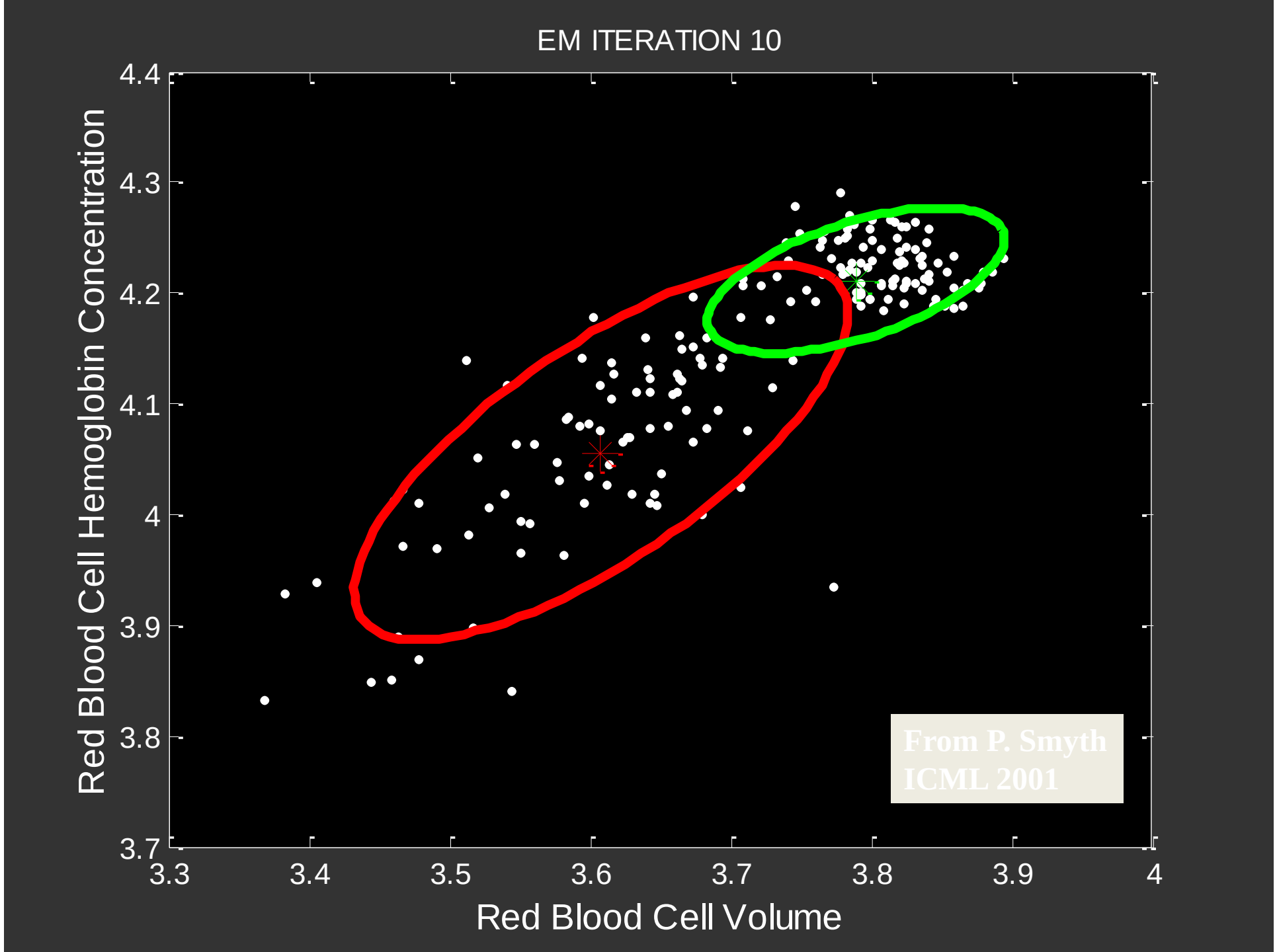
- What should we do
  - If we want to choose a single cluster for an “answer”?
  - With new data we didn’t see during training?

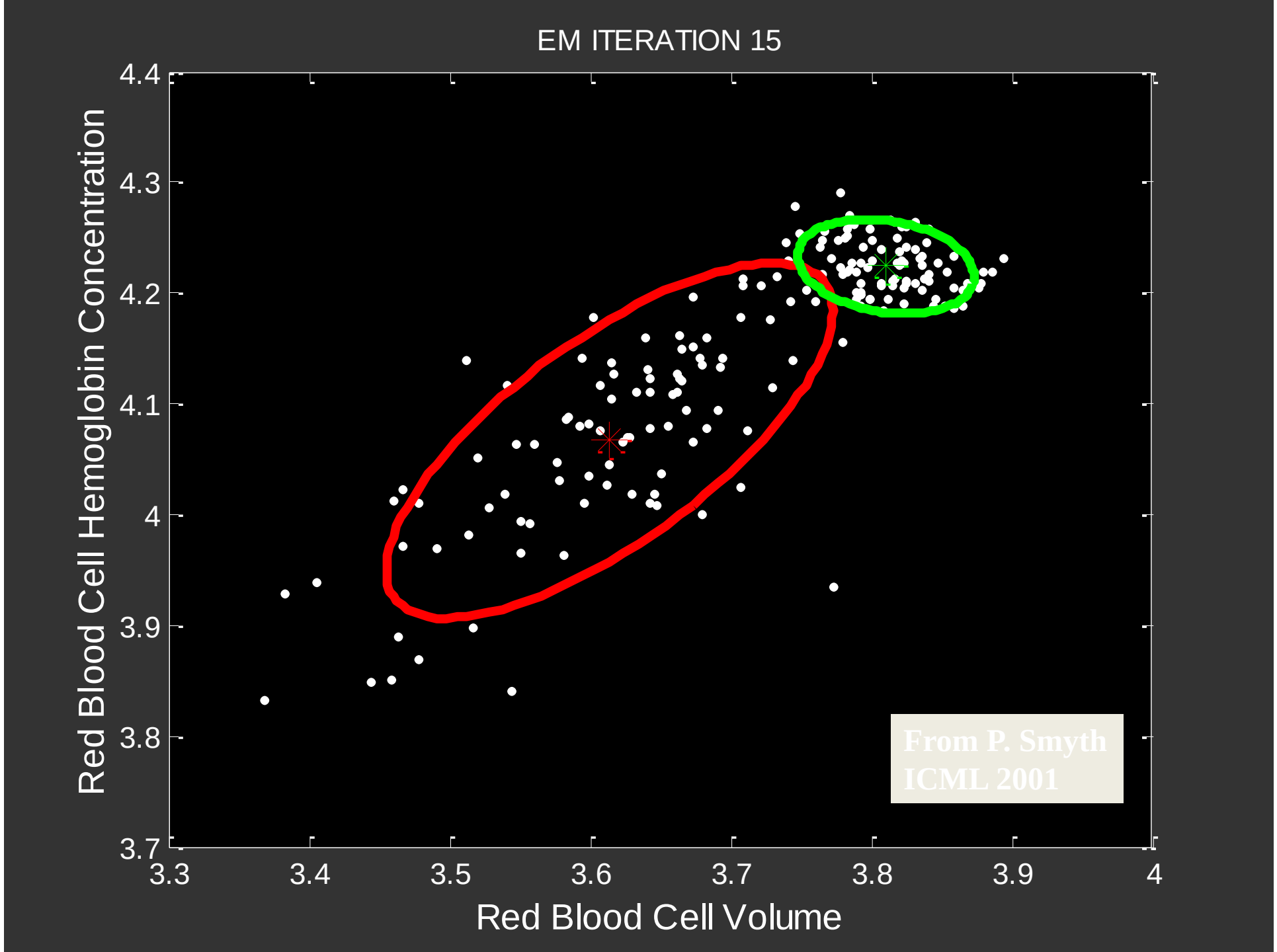

ANEMIA PATIENTS AND CONTROLS
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001

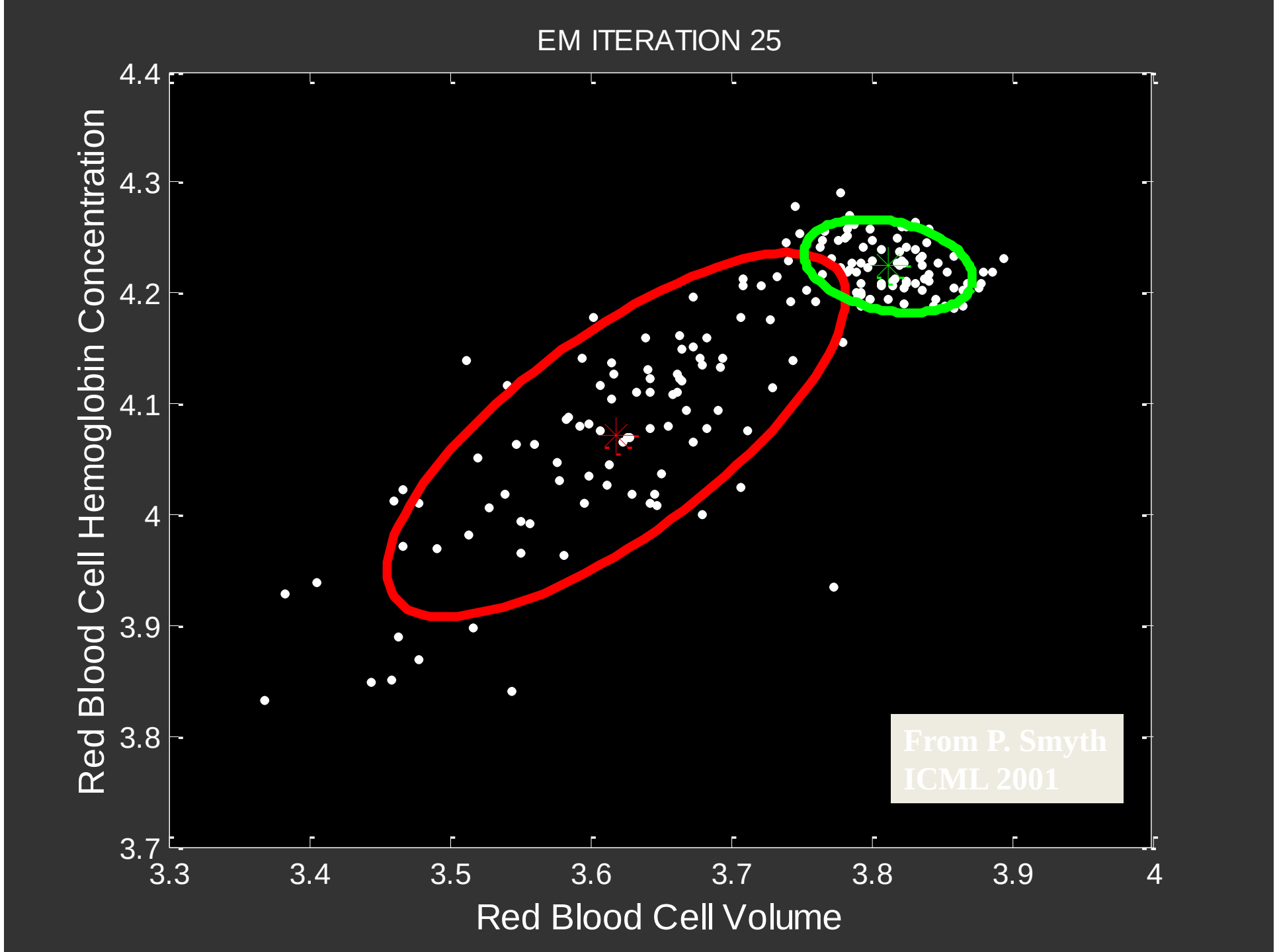

EM ITERATION 1
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001

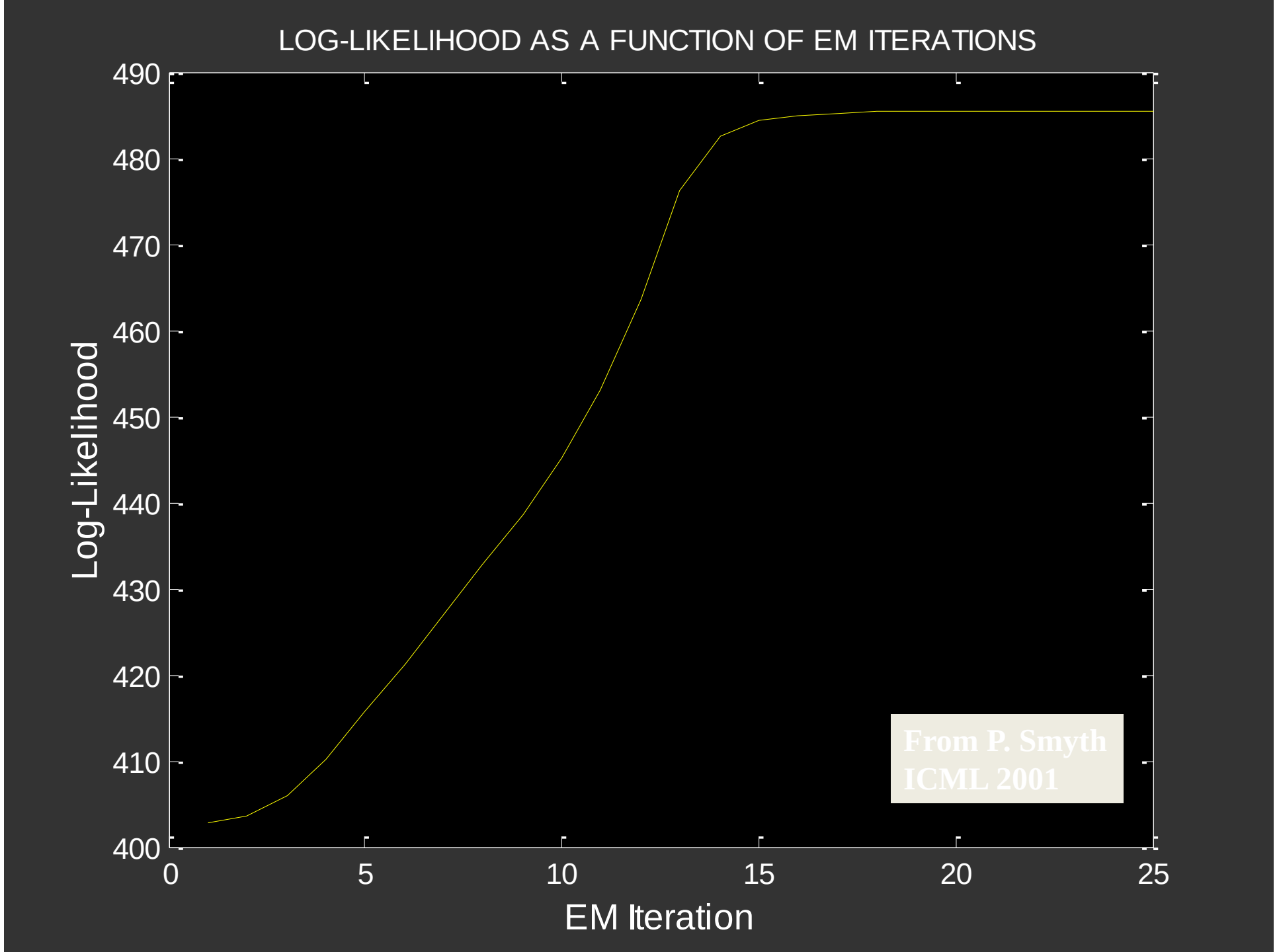

EM ITERATION 3
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001


EM ITERATION 5
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001

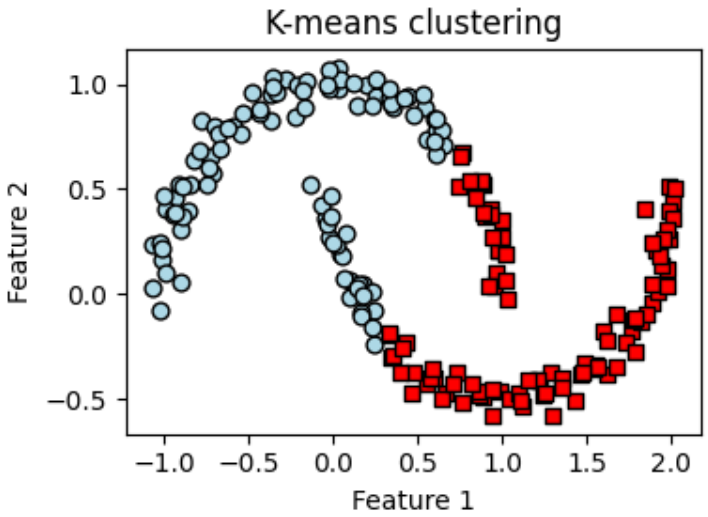

EM ITERATION 10
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001

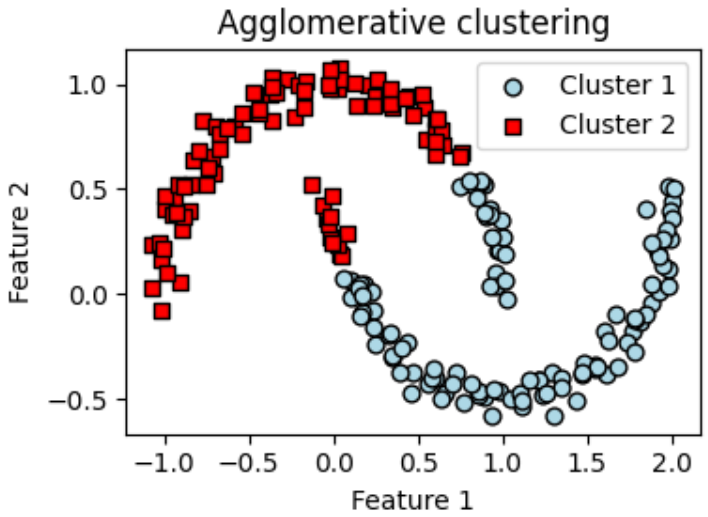

EM ITERATION 15
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001

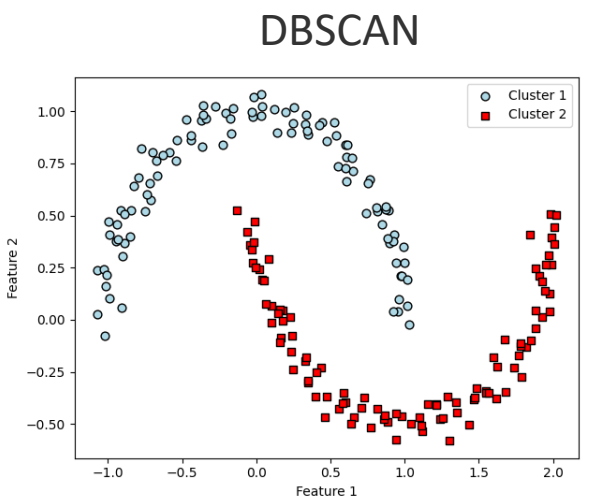

EM ITERATION 25
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001


LOG-LIKELIHOOD AS A FUNCTION OF EM ITERATIONS
Log-Likelihood
EM Iteration
490
480
470
460
450
440
430
420
410
400
0
5
10
15
20
25
From P. Smyth
ICML 2001

Density Based Clustering





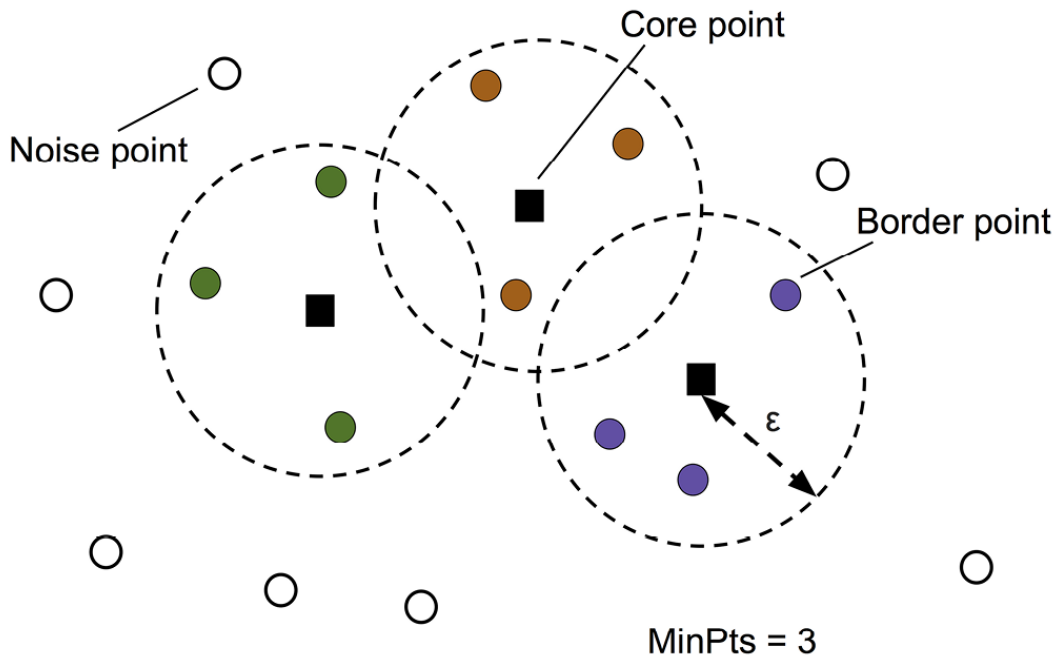


- Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points
- Discovers clusters of arbitrary shape in spatial databases with noise

Density Based Clustering

- Two parameters**:**
  - ***Eps***: Maximum radius of the neighbourhood
  - ***MinPts***: Minimum number of points in an Eps-neighbourhood of that point
- $N_{Eps}(p)$: $\{q \text{ belongs to } D \mid dist(p,q) <= Eps\}$
- Directly density-reachable**:** A point ***p*** is directly density-reachable from a point ***q*** wrt. ***Eps***, ***MinPts*** if
  - ***p*** belongs to $N_{Eps}(q)$
  - core point condition: $|N_{Eps}(q)| >= MinPts$

Density Based Clustering



- Arbitrary select a point ***p***
- Retrieve all points density-reachable from ***p*** wrt ***Eps*** and ***MinPts***.
- If ***p*** is a core point, a cluster is formed.
- If ***p*** is a border point, no points are density-reachable from ***p*** and DBSCAN visits the next point of the database.
- Continue the process until all of the points have been processed.

Summary

- In clustering, clusters are inferred from the data without human input (unsupervised learning)

- However, in practice, it is very domain specific:
  - Definition of distance in data space
  - Representation of data
  - Defining distance between clusters
  - Number of clusters
  - And so on.

- Practice, practice, practice!

Spectroscopic Imaging

Advancements in imaging led to a broad spectrum of the spectroscopic imaging techniques, in which response spectra are measured in each spatial location giving rise to 3- and higher dimensional data.

Scanning probe microscopy:

- Force-distance curve measurements
- Current-voltage measurements

Electron microscopy:

- Electron Energy Loss Spectroscopy

Optical microscopy:

- Hyperspectral imaging
- Time resolved measurements

Mass-spectrometry:

- Secondary ion MS imaging

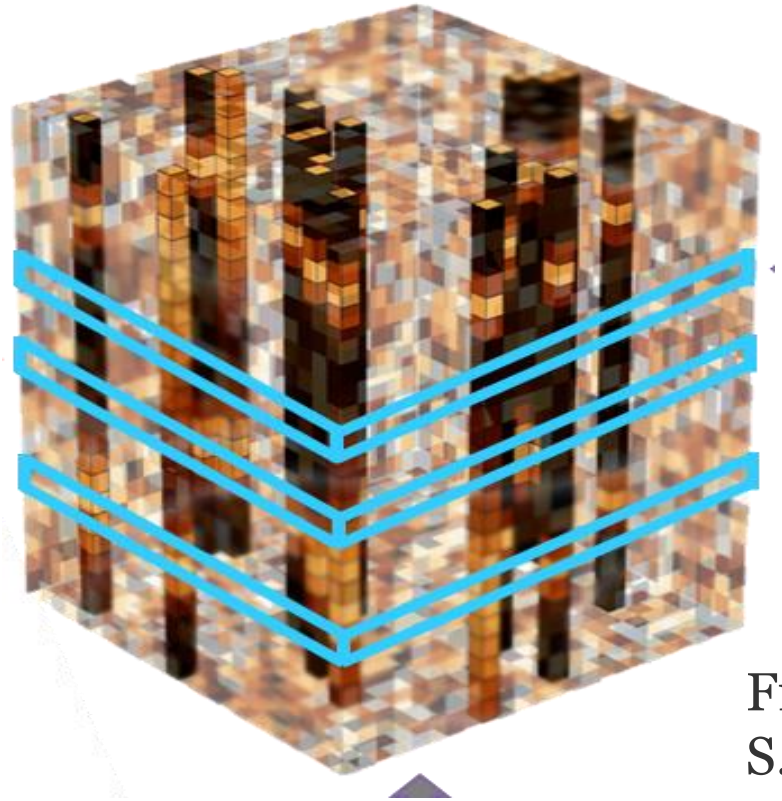
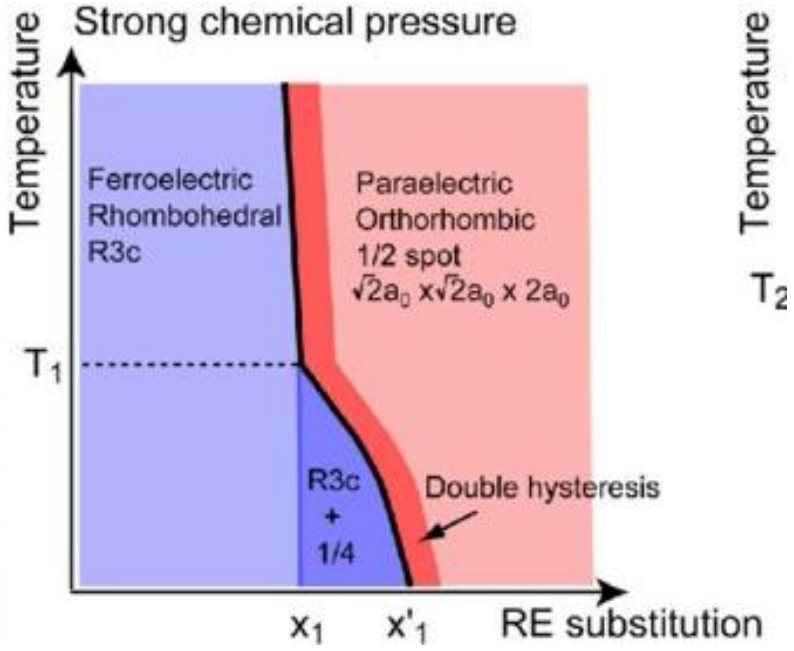
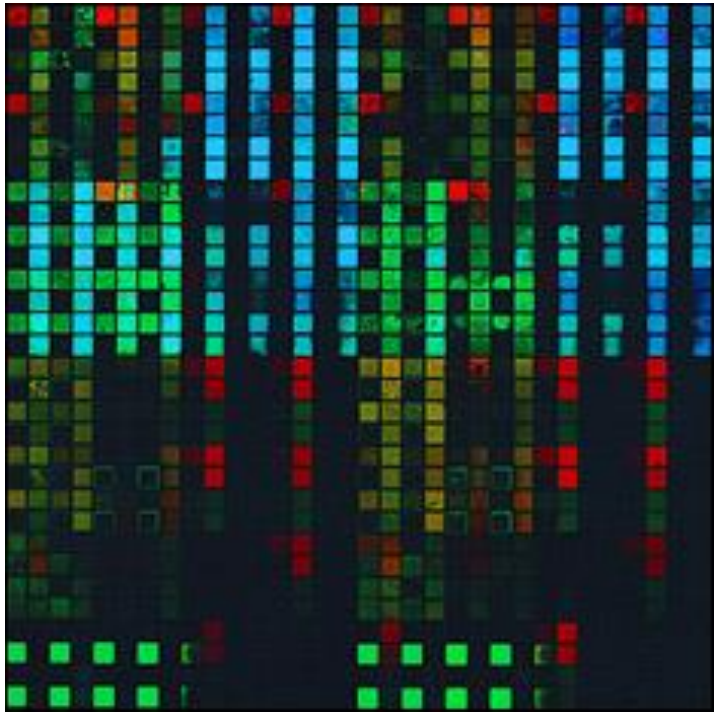


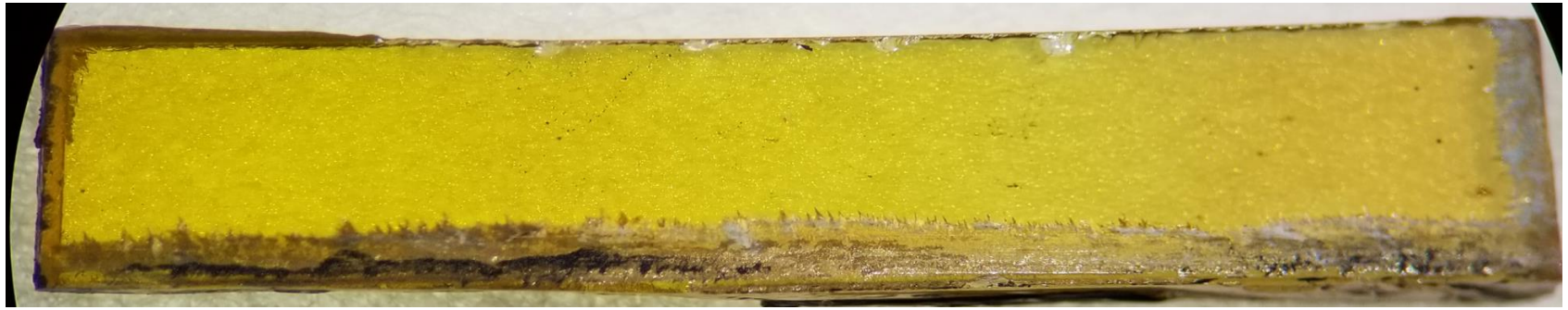
Figure by S. Jesse

Combinatorial Libraries



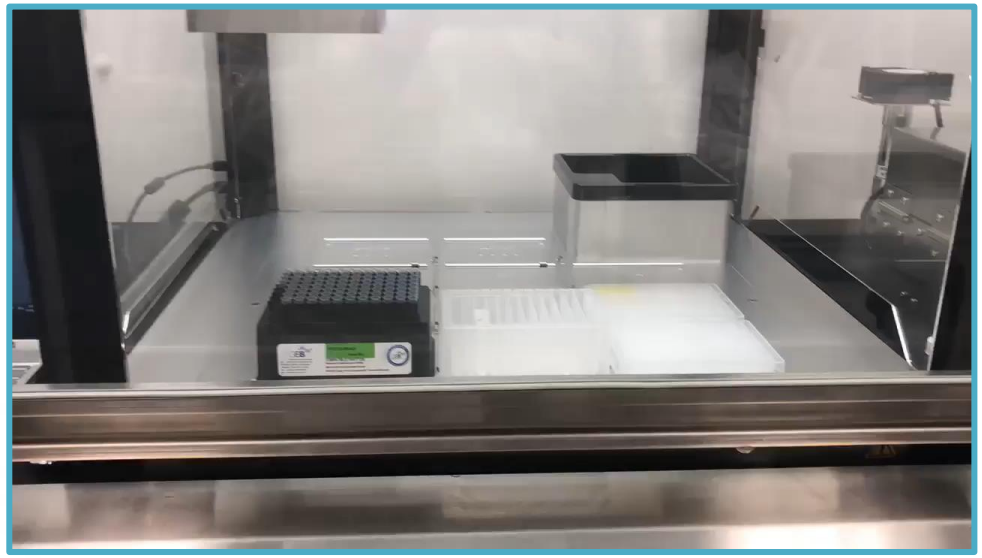


https://mse.umd.edu/research/spotlight/combinatorial

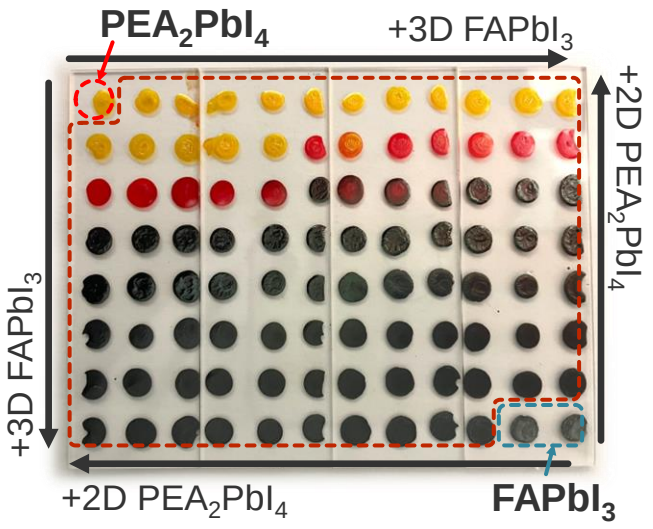


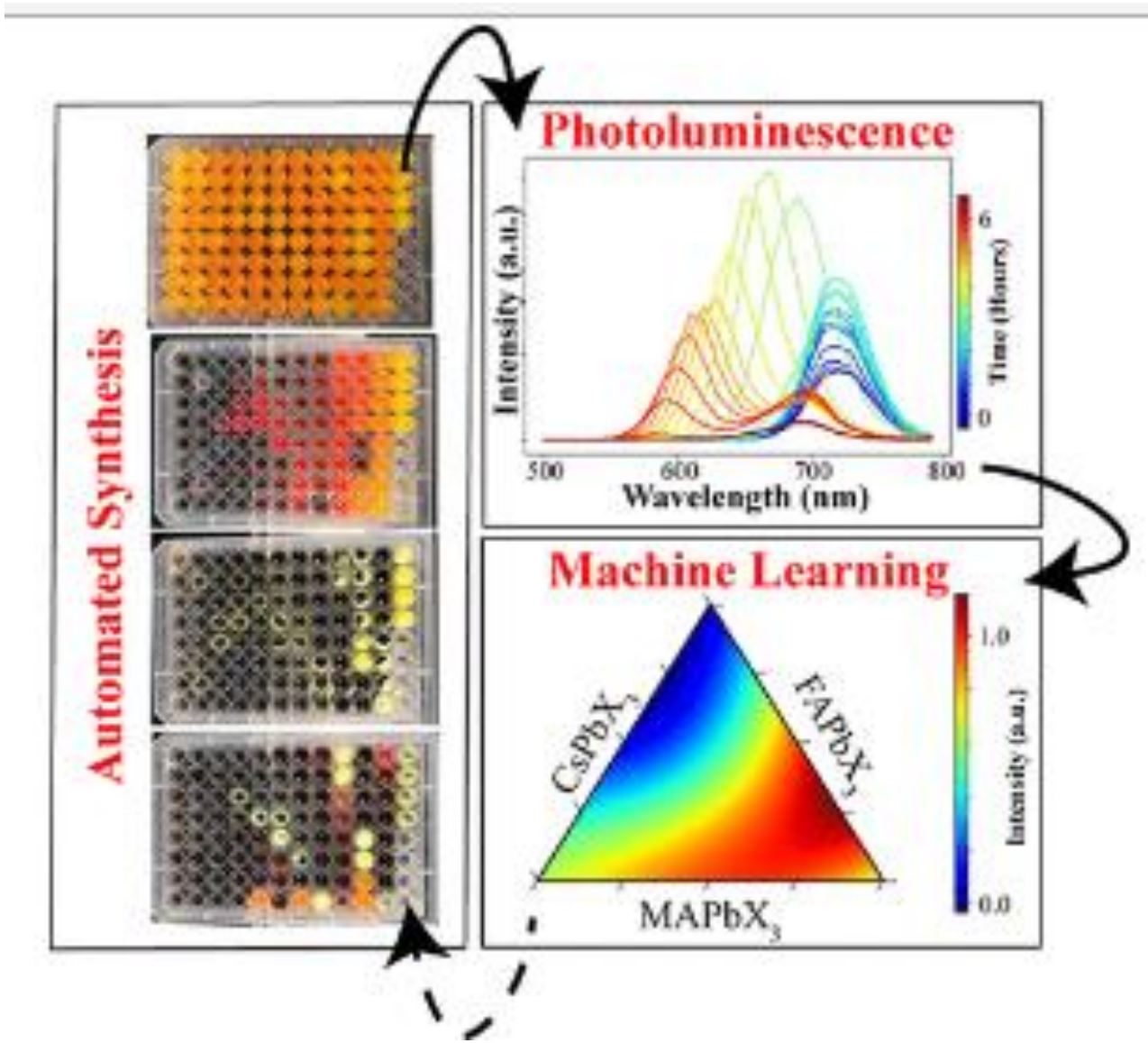
$BiFeO_3$ Linear est. 7%Sm $BiFeO_3$ 20%Sm $BiFeO_3$

Combinatorial Libraries

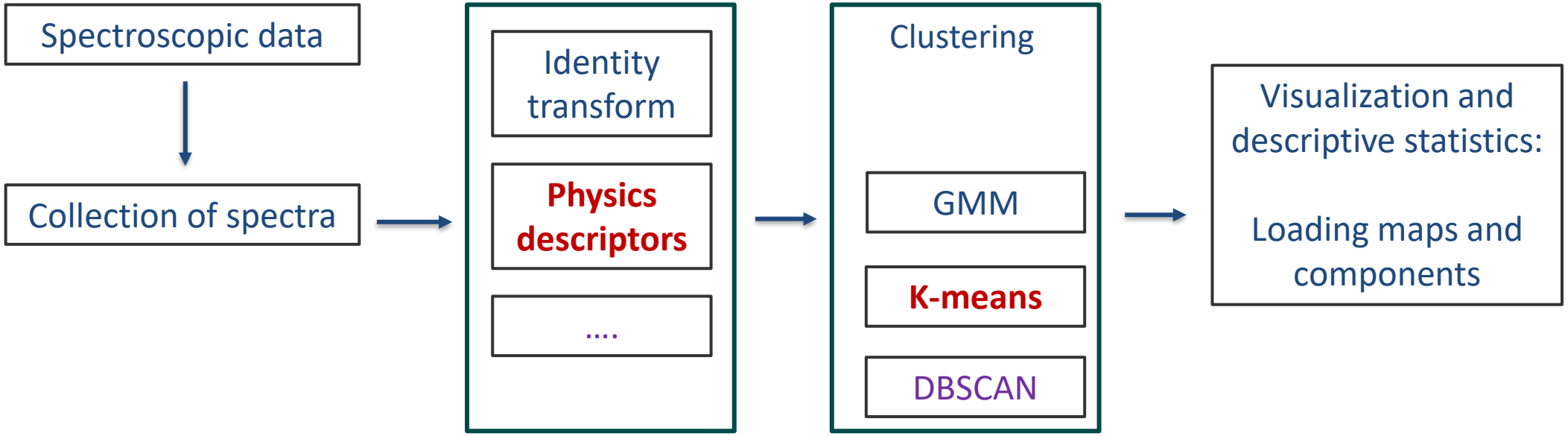


Quasi-2D MHP Film Array





Analysis pipeline

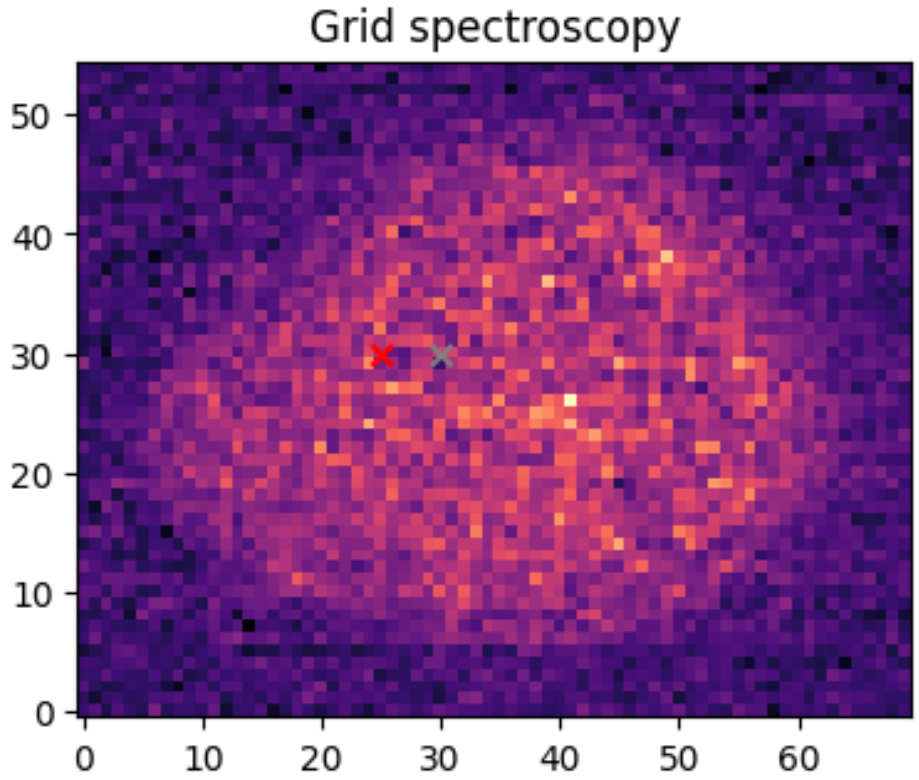


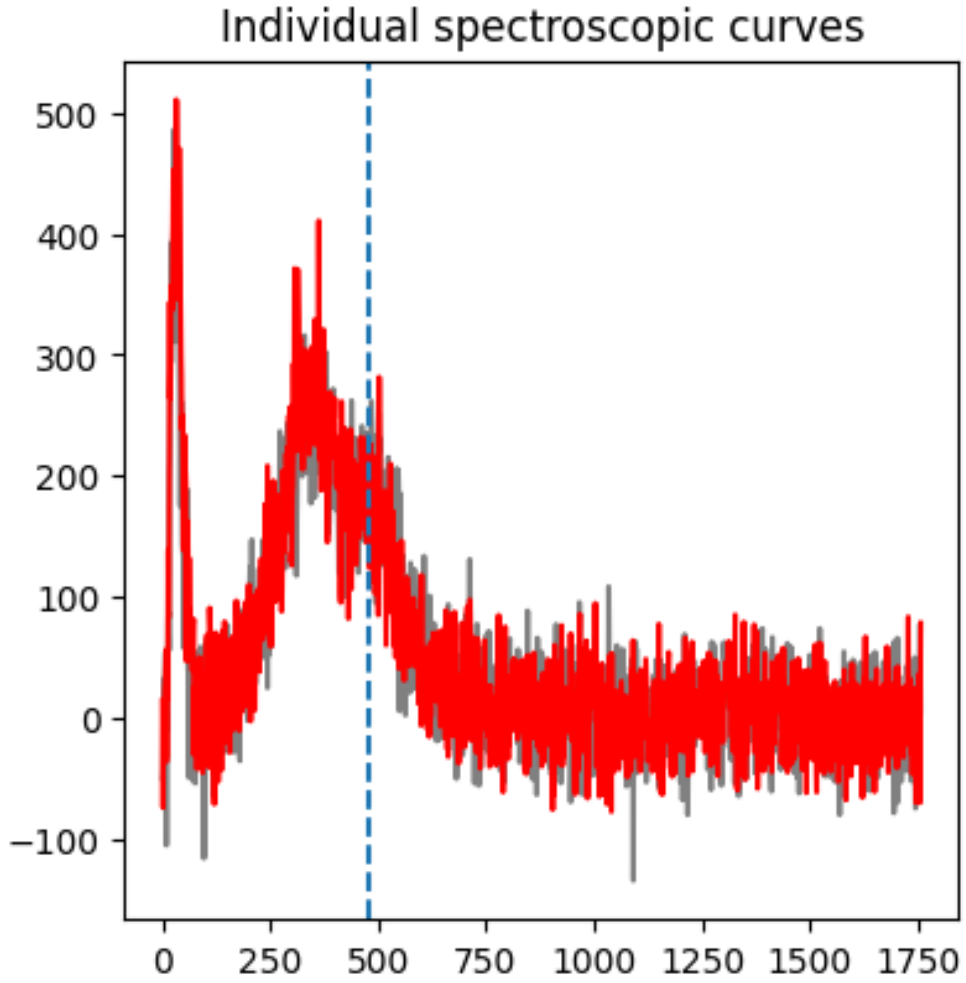
Pipelines are defined to

- Make analysis traceable, repeatable, explainable, and transferable
- Allow for hyperparameter tuning and optimization
- Efficiently use the memory

EELS Colab

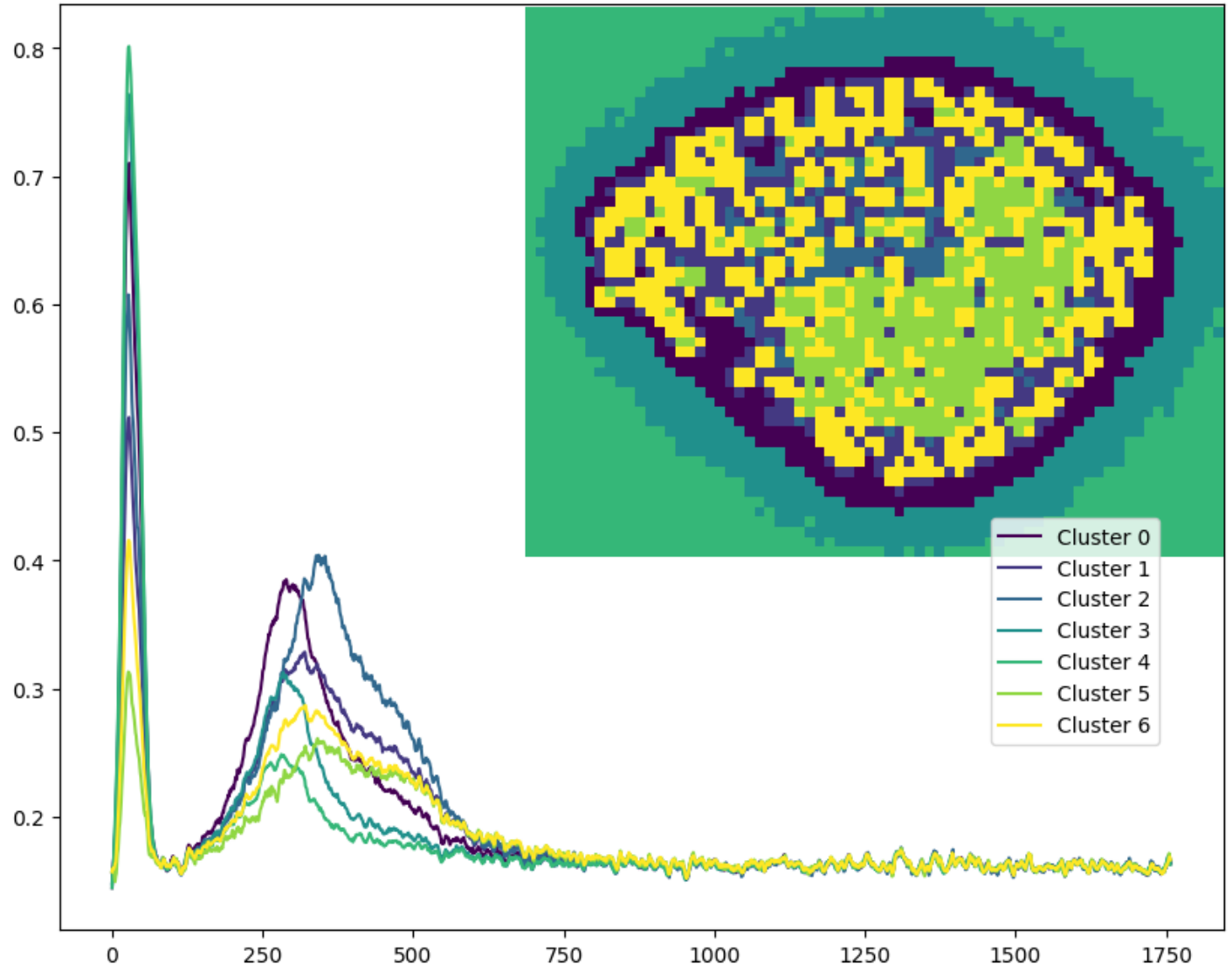
Clustering of spectral data





- The hyperspectral data set contains spectrum at each spatial position on the dense rectangular grid
- We use clustering to establish internal structure of this dataset

Clustering of spectral data

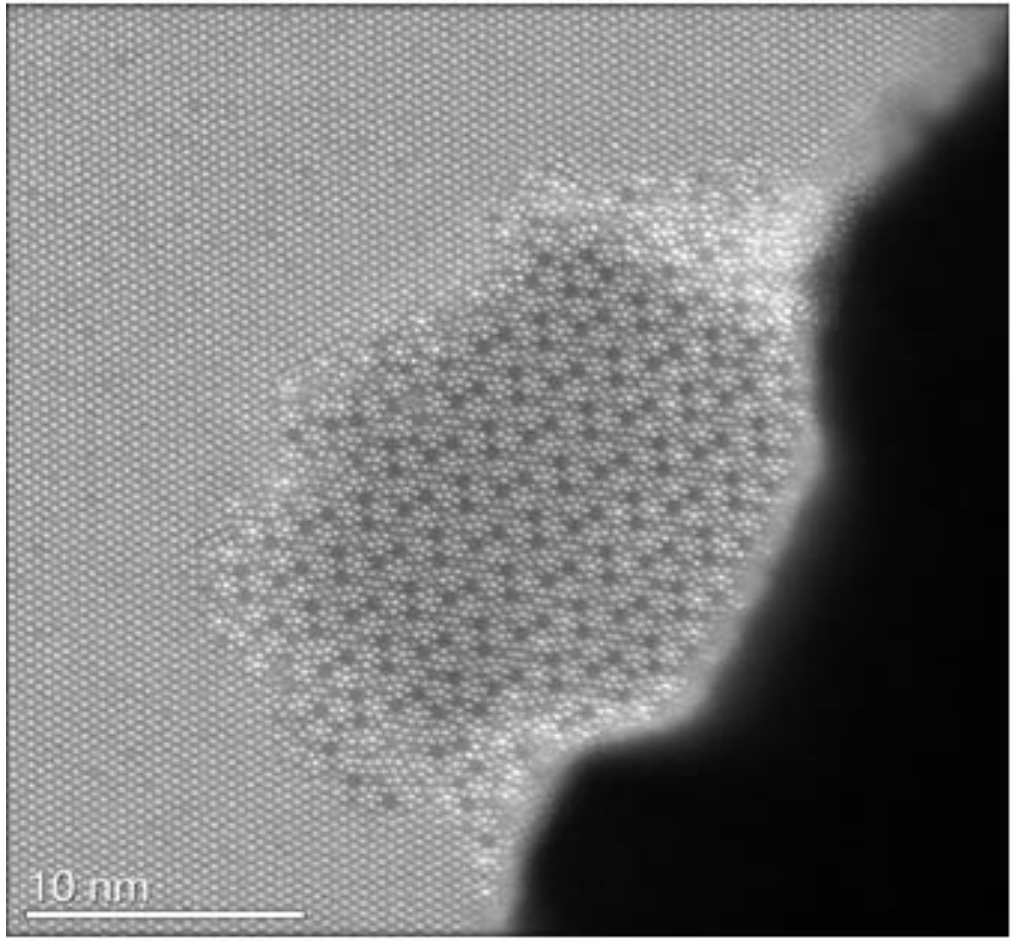


- Experiment with number of clusters
- Based on domain experience, explore the behavior of the components and images of class labels
- This is already “real” research

But what about images?

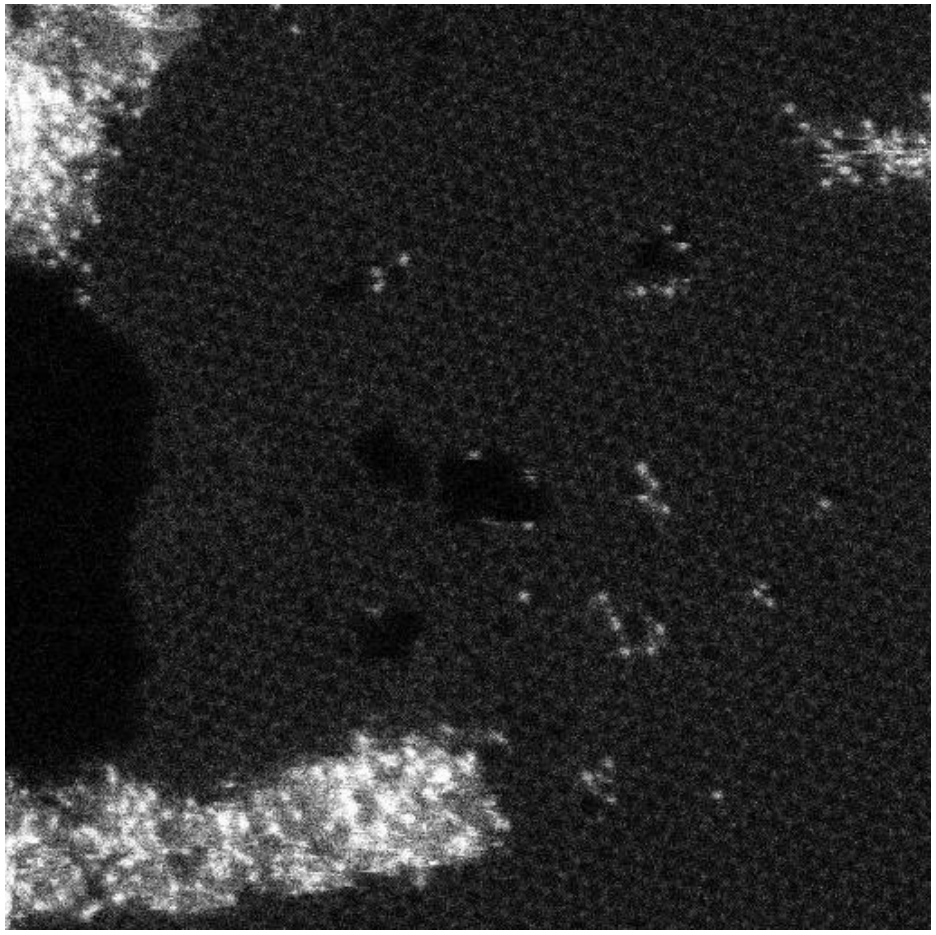
Chemically disordered systems

Mo-V-Ta complex oxide



Q. He et al, ACS Nano 9, 3470-3478

Si in graphene

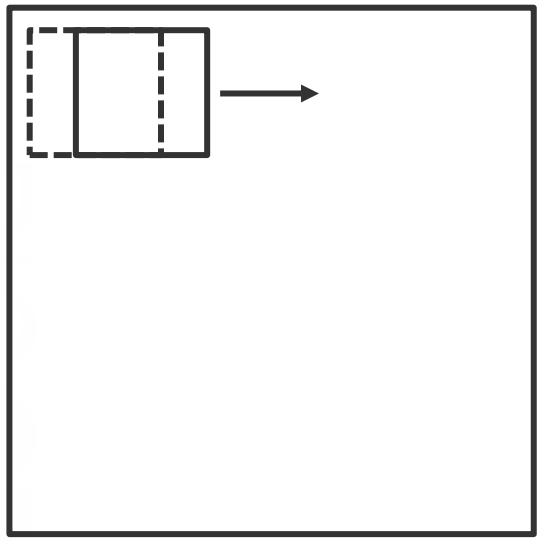


Data collected by O. Dyck (ORNL)

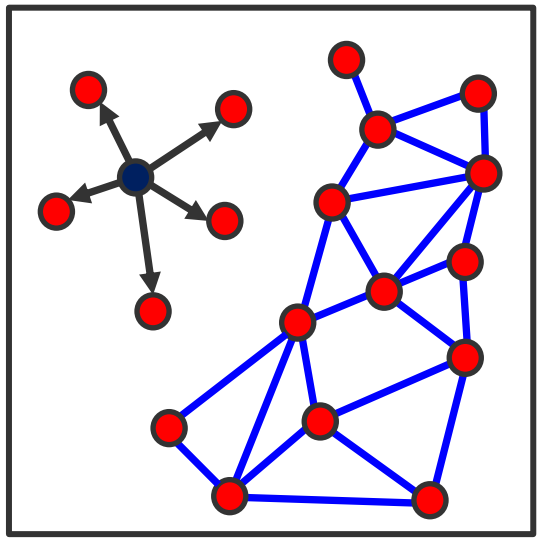
- What is the nature of the building blocks and relevant atomic configurations?
- Can we define single-phase regions and phase boundaries?

Constructing the descriptors

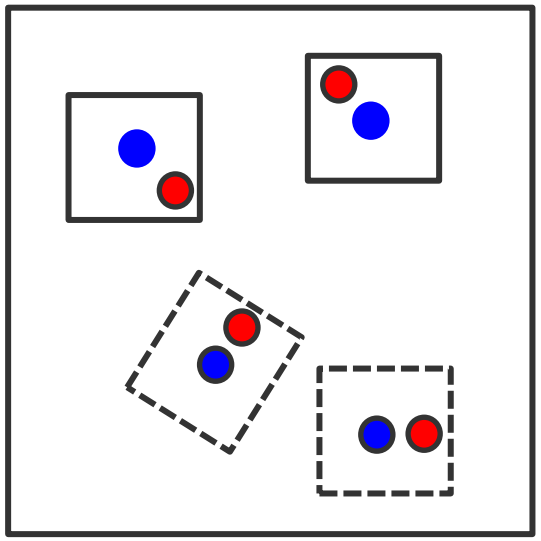
Continuous translational symmetry



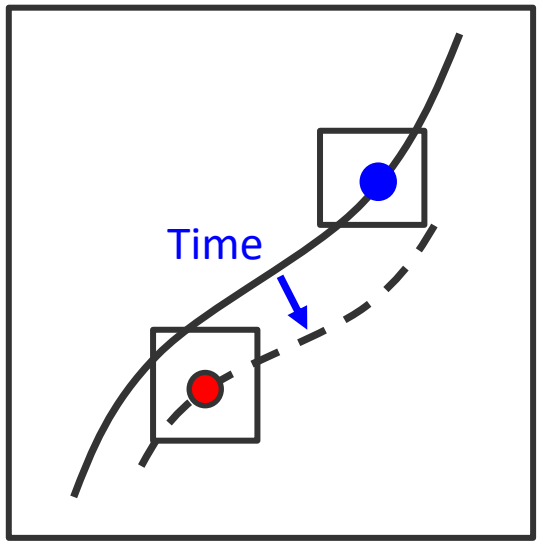
Atom based descriptions



Localized sub-images



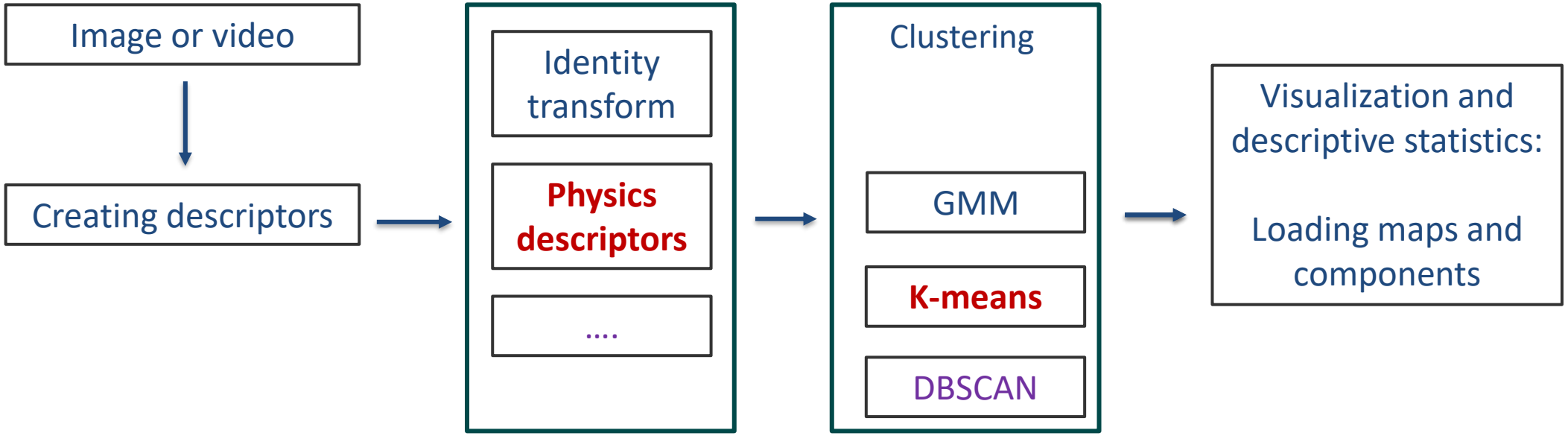
Time-delayed descriptors



The choice of the descriptor:

- Defines physical inferential biases and allows to introduce prior knowledge
- Determines the physical meaning of the analysis
- Establishes the analysis pipeline

Example of analysis pipeline

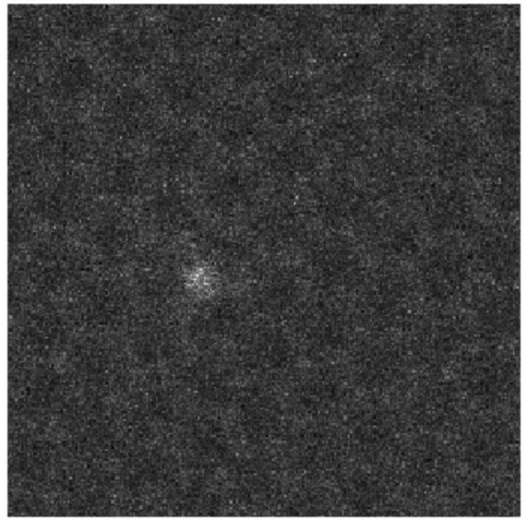


Pipelines are defined to

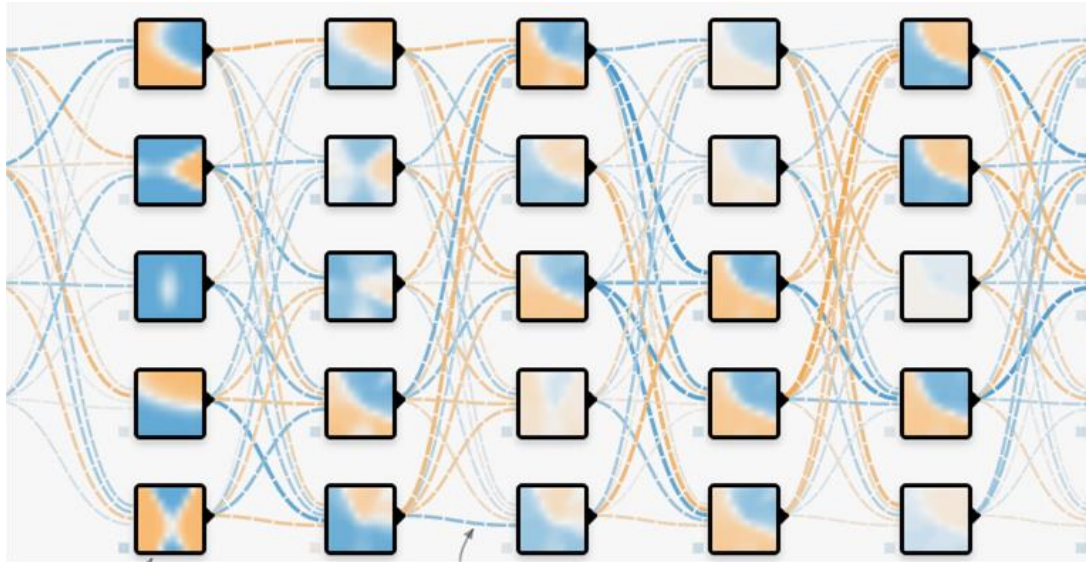
- Make analysis traceable, repeatable, explainable, and transferable
- Allow for hyperparameter tuning and optimization
- Efficiently use the memory

Machine learning can help!

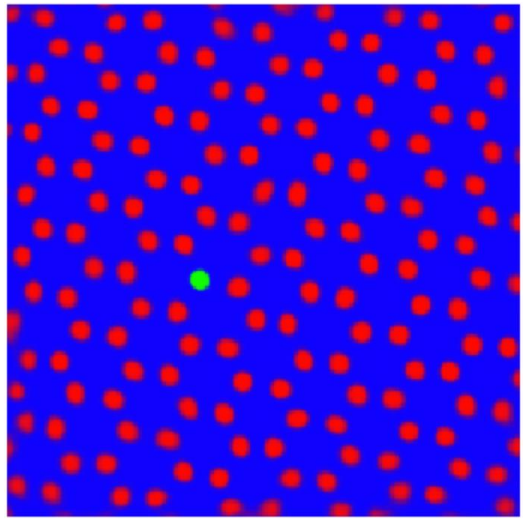
Input



Deep Neural Network (aka deep learning)



Output

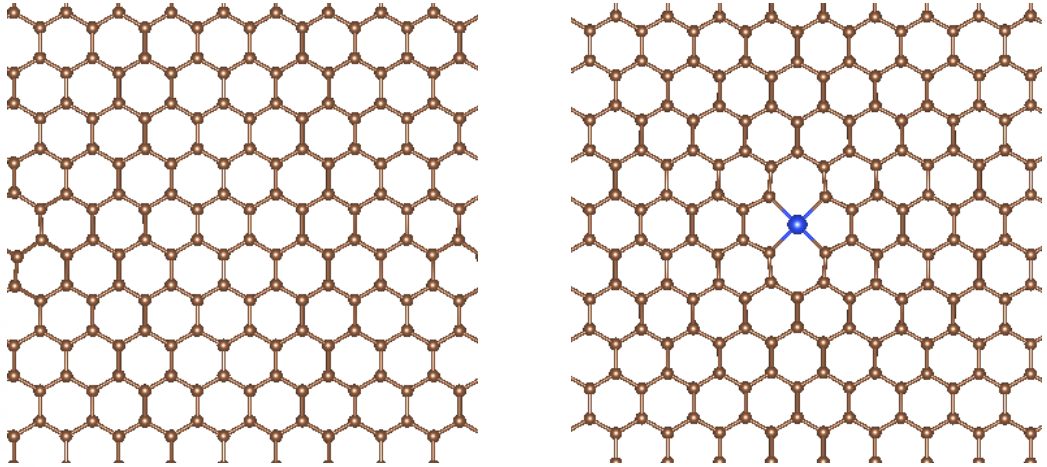


The unique aspect of data in high resolution electron microscopy is the presence of well-defined ground truth (atomic positions) and reasonably well understood physics of the imaging process (allows a creation of a training set with problem-specific physical constraints.

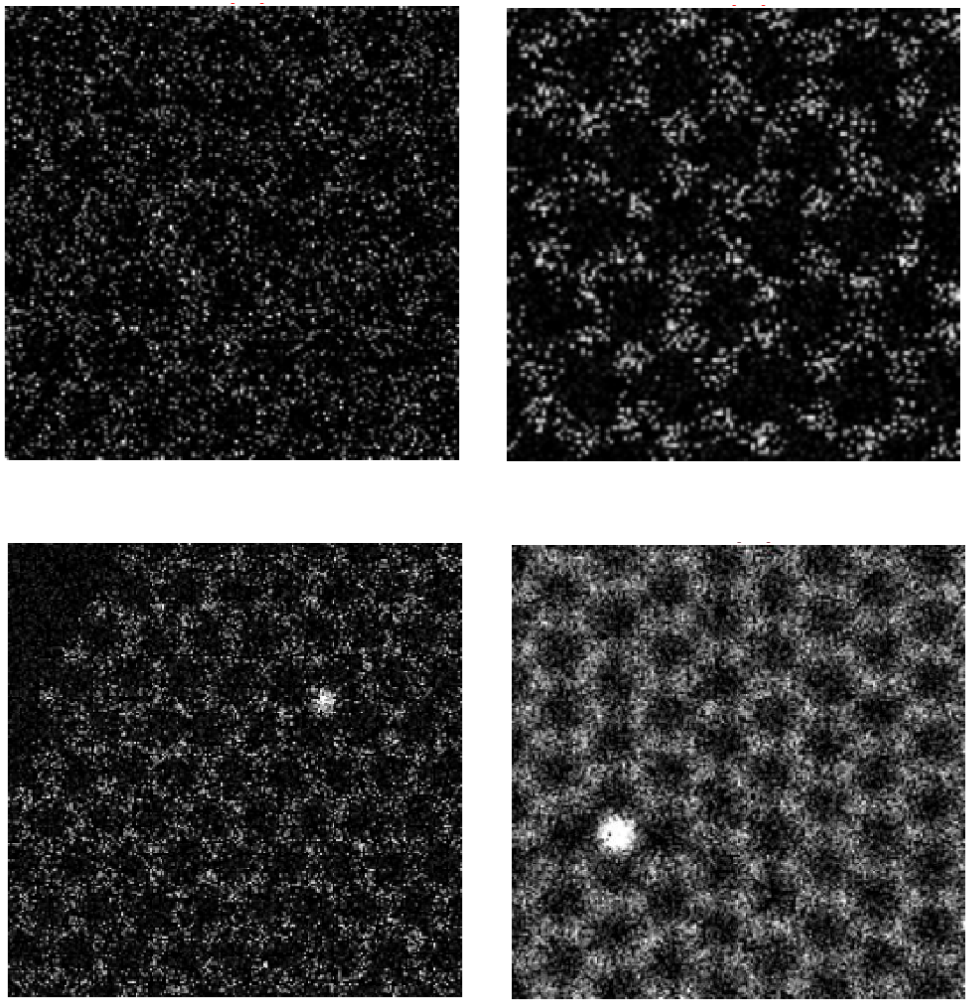
Slide by M. Ziatdinov

Generating training data

Theoretical structures



Simulated images (Training data)

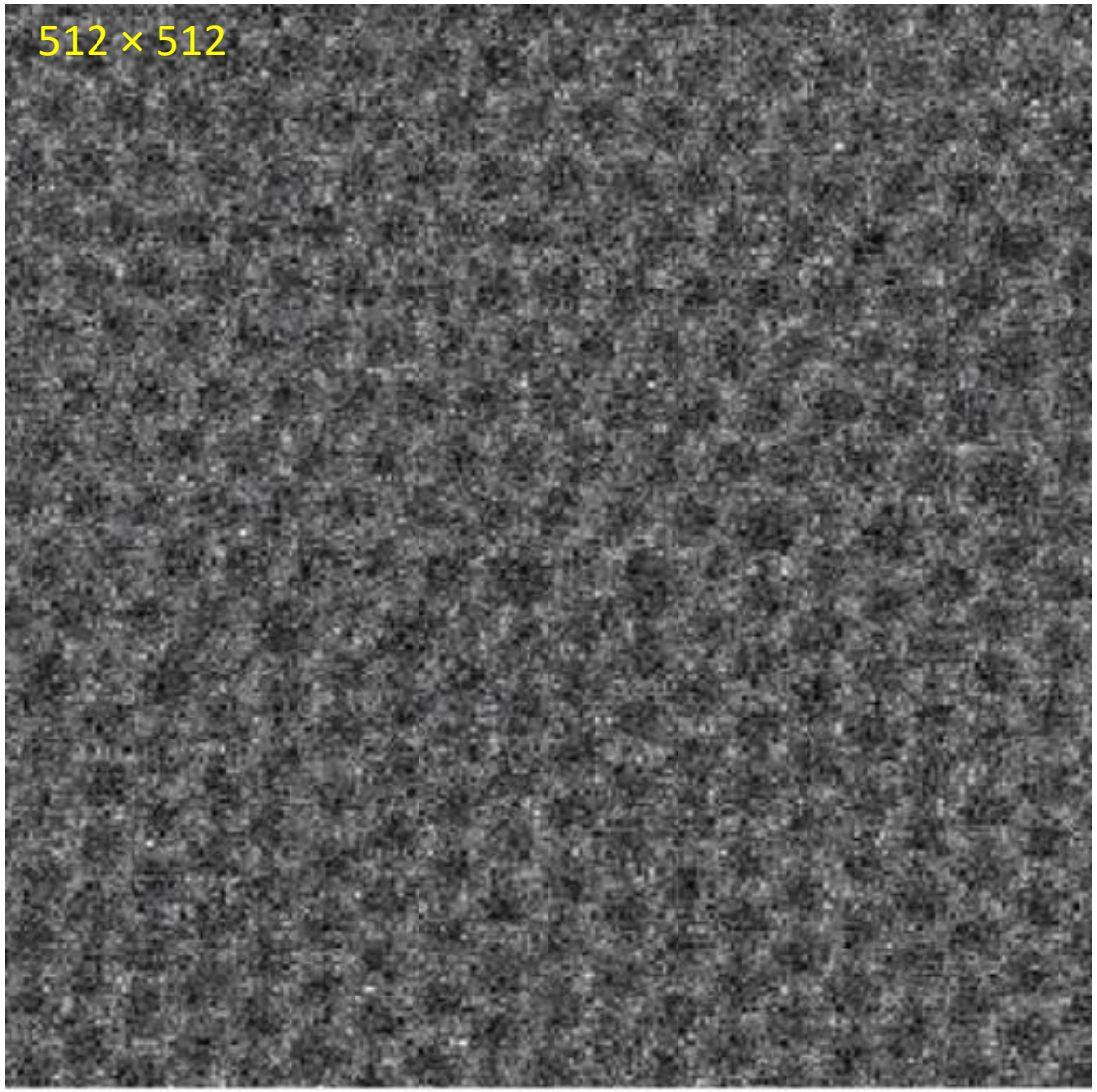


- Atomic coordinates from DFT or MD
- Each structure “augmented” by applying different strain
- Image simulation:
  - Multislice algorithm
  - Atoms as 2D Gaussians
- Image augmentation to account for instrumental factors

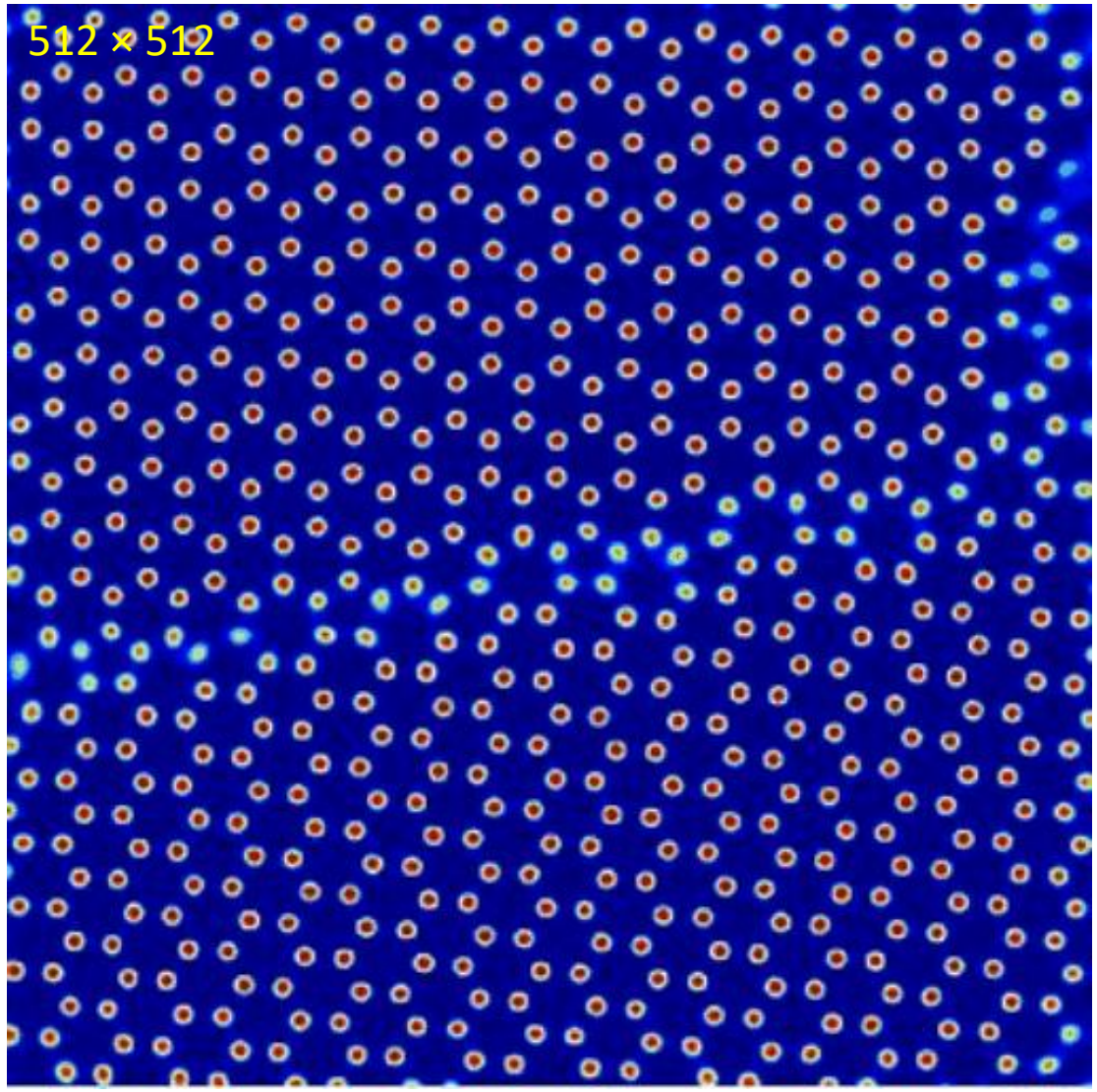
Slide by M. Ziatdinov

Application to electron microscopy data

Experiment



Neural Network Output



Slide by M. Ziatdinov

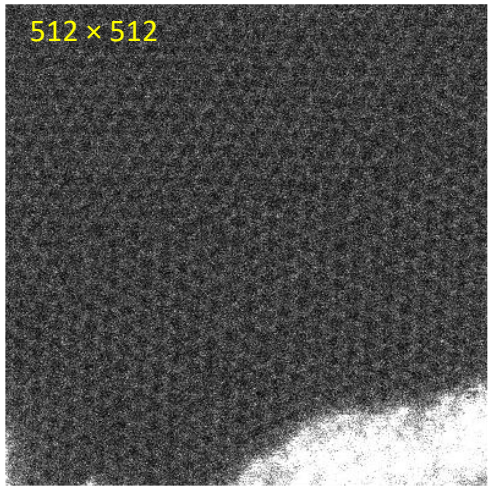
Application to electron microscopy data

Avoids surface "junk"

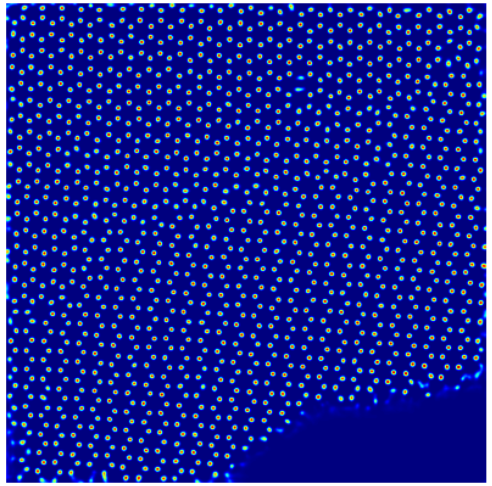
Distinguishes between point impurities and amorphous

Robust to changes in image dimensions

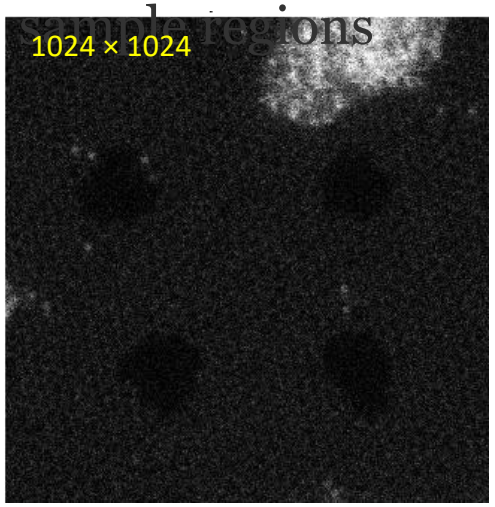
Experimental



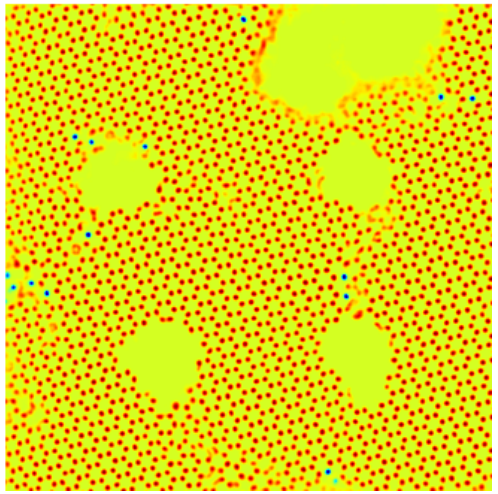
DL output



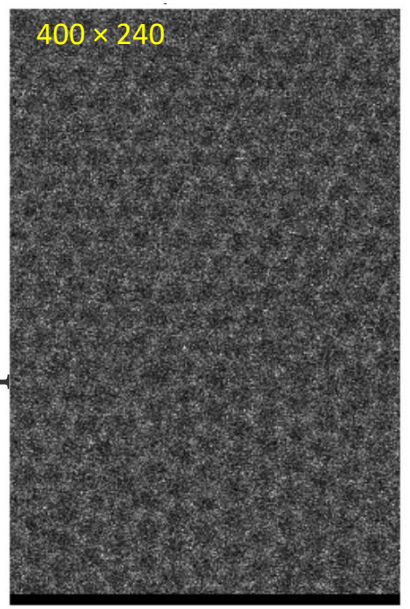
Experimental



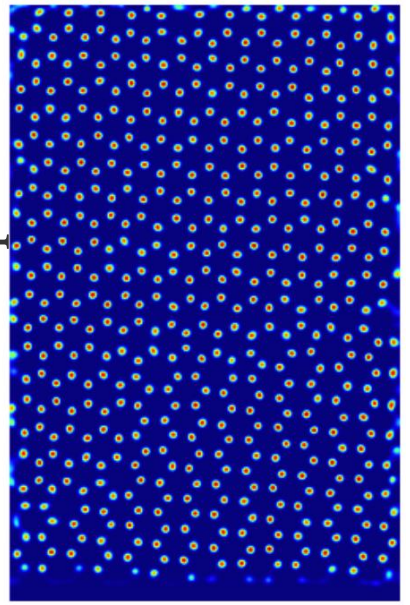
DL output



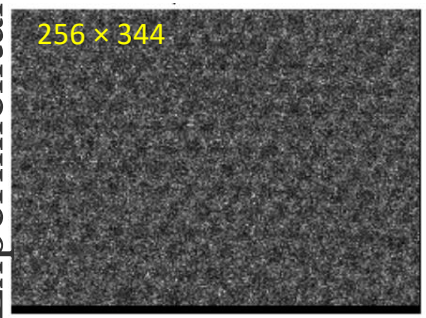
Experimental



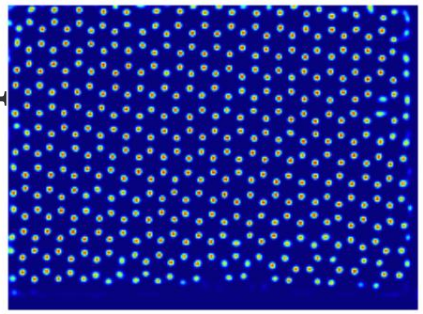
DL output



Experimental

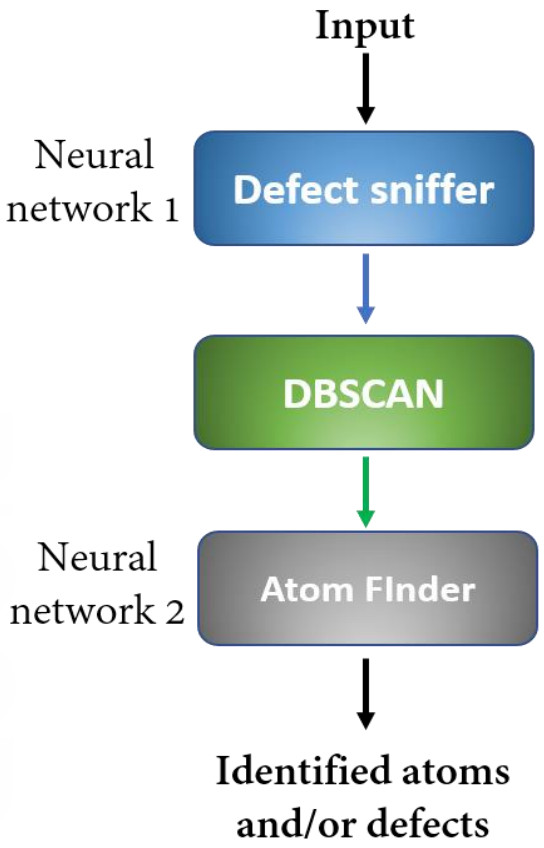


DL output



Slide by M. Ziatdinov

Practically: pipelines of simpler NNs



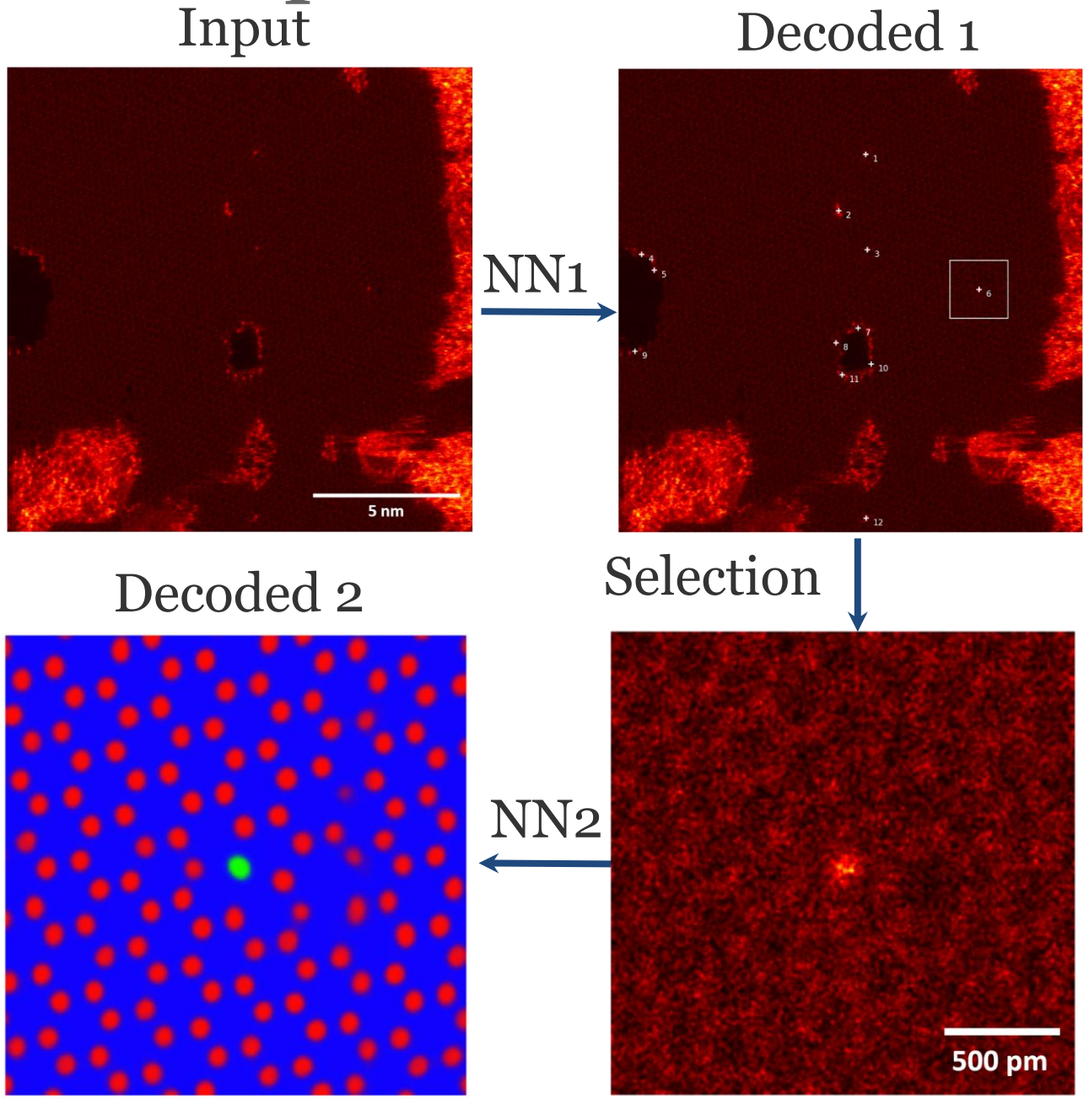
Finding needle in a haystack

We replace categorical cross entropy (CE) with focal loss (FL) function

$$CE(p_t) = -\log(p_t)$$

$$FL(p_t) = -(1-p_t)^{\gamma}\log(p_t)$$

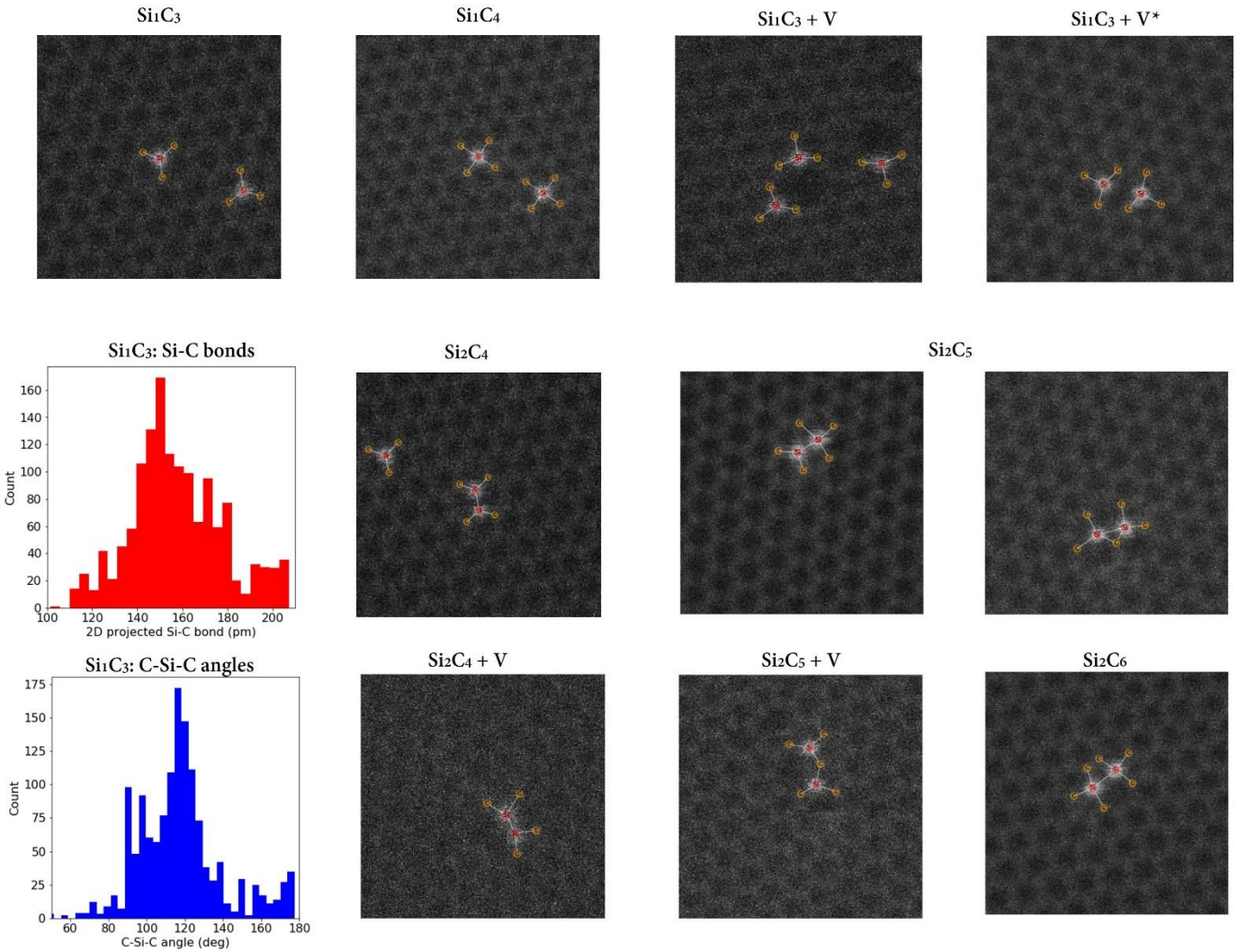
T.-Y. Lin et al., arXiv 1708.02002 (2018)



Opportunities: Building Libraries of Defects

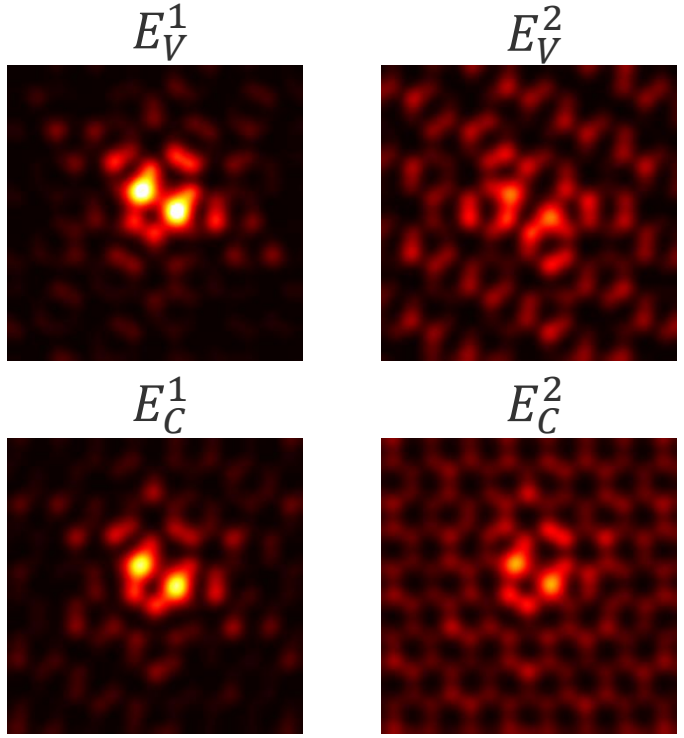
Total number of images analyzed: ~600 (ranging from 256 × 256 to 2048 × 2048 and from 2 nm × 2 nm to 16 nm × 16 nm)

Creation of Si-vacancy libraries from STEM data



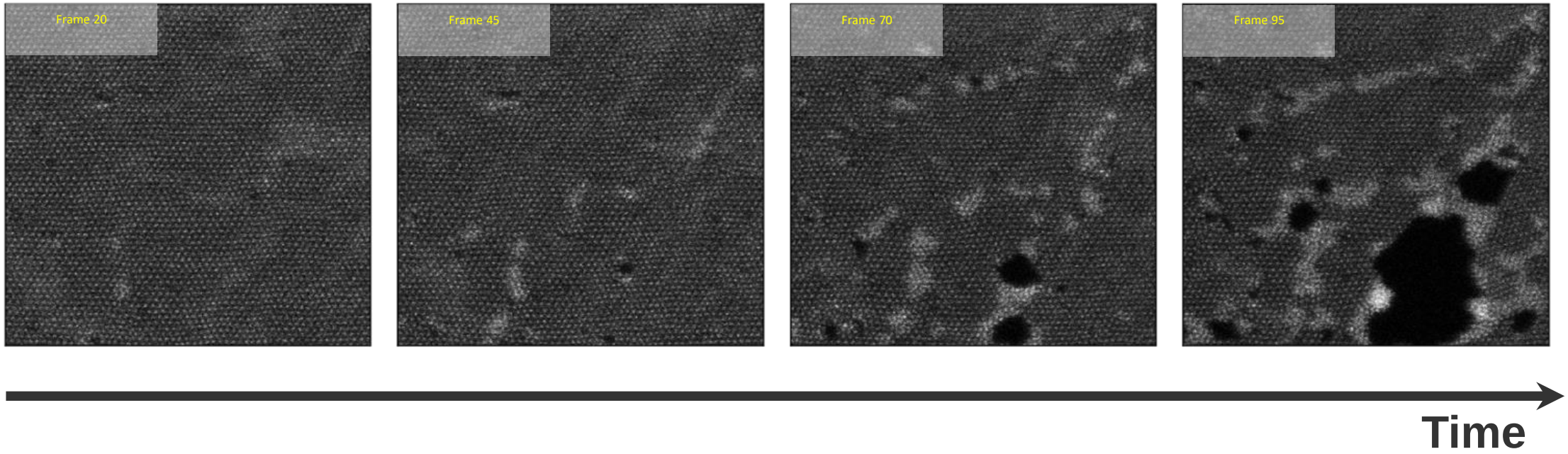
Adding electronic structure calculations

Si2C6



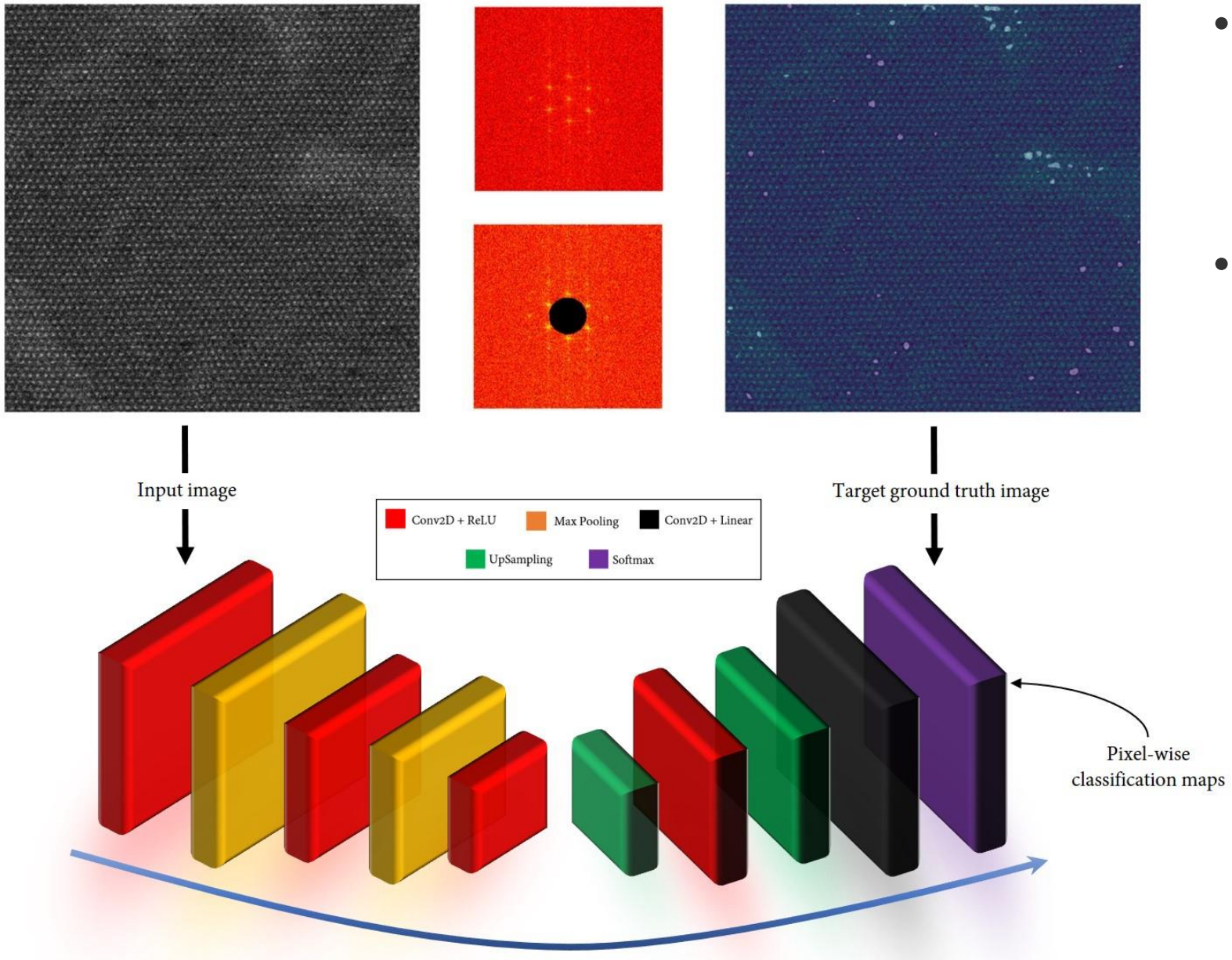
M. Ziatdinov *et al.* ArXiv:1809.04256 (2018).

Exploring defect dynamics: ws_2



- To train a deep learning model, we exploit the fact that each defect is associated with violation of ideal periodicity of the lattice
- We train a model using a single image at the early stage of the beam-induced transformation, when macroscopic periodicity is still maintained, and each defect can be readily discovered, providing the "ground truth" for network training.
- The extracted defect structures are then classified using unsupervised clustering and unmixing techniques

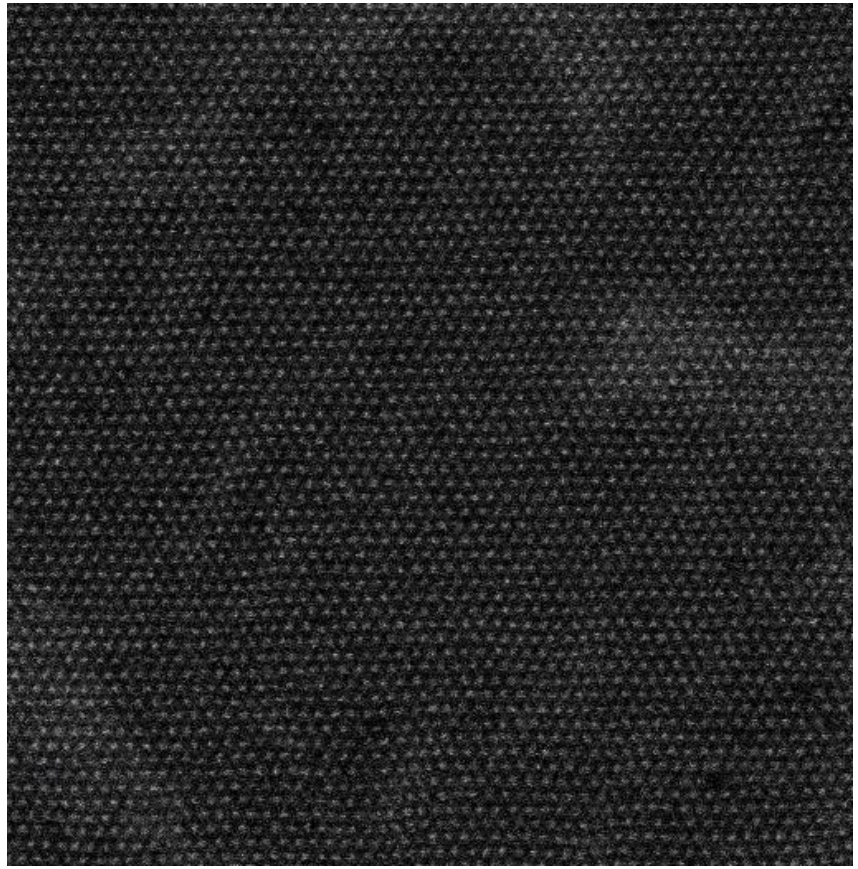
Exploring defect dynamics



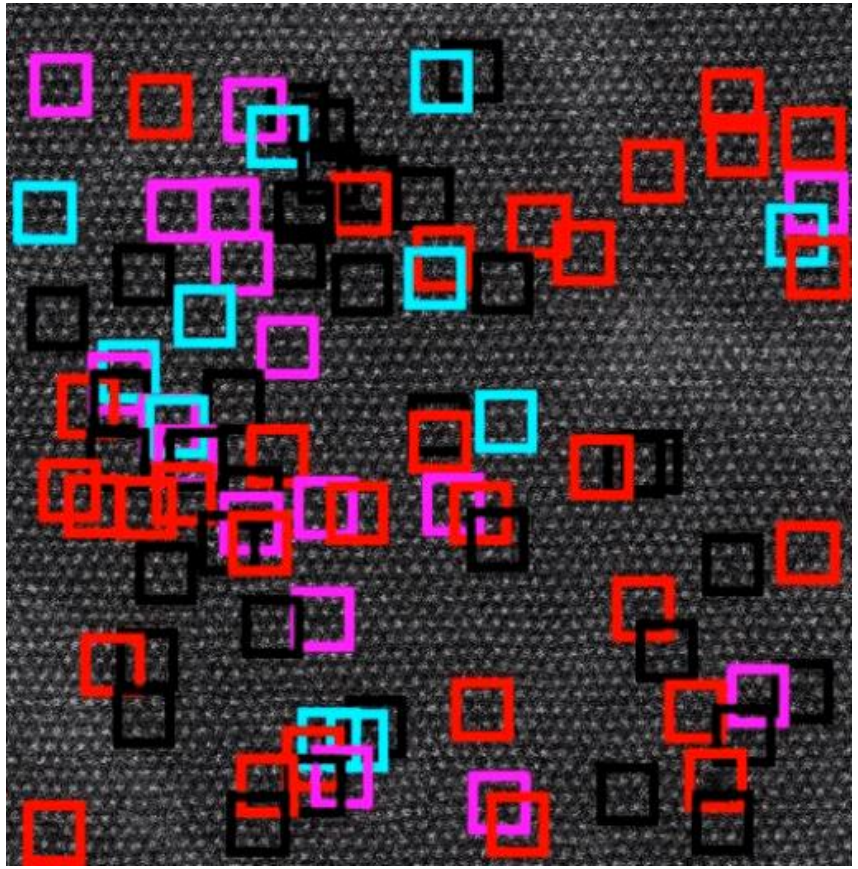
- Training a CNN model (defect identification)
- Gaussian mixture model (defect classification)

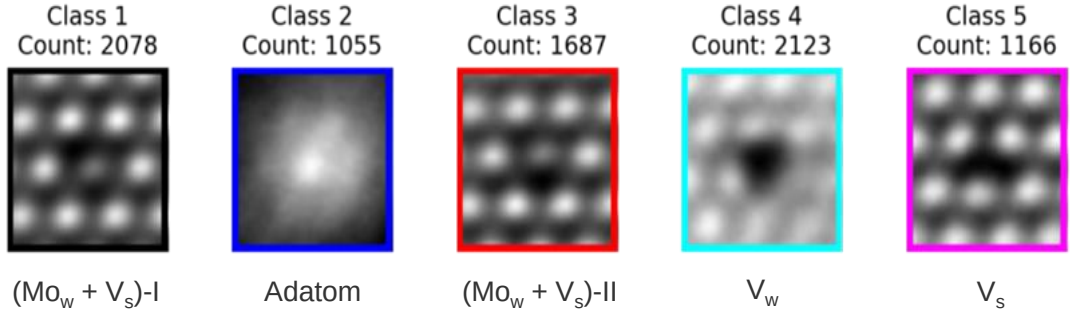
Exploring defect dynamics

Experimental



Decoded

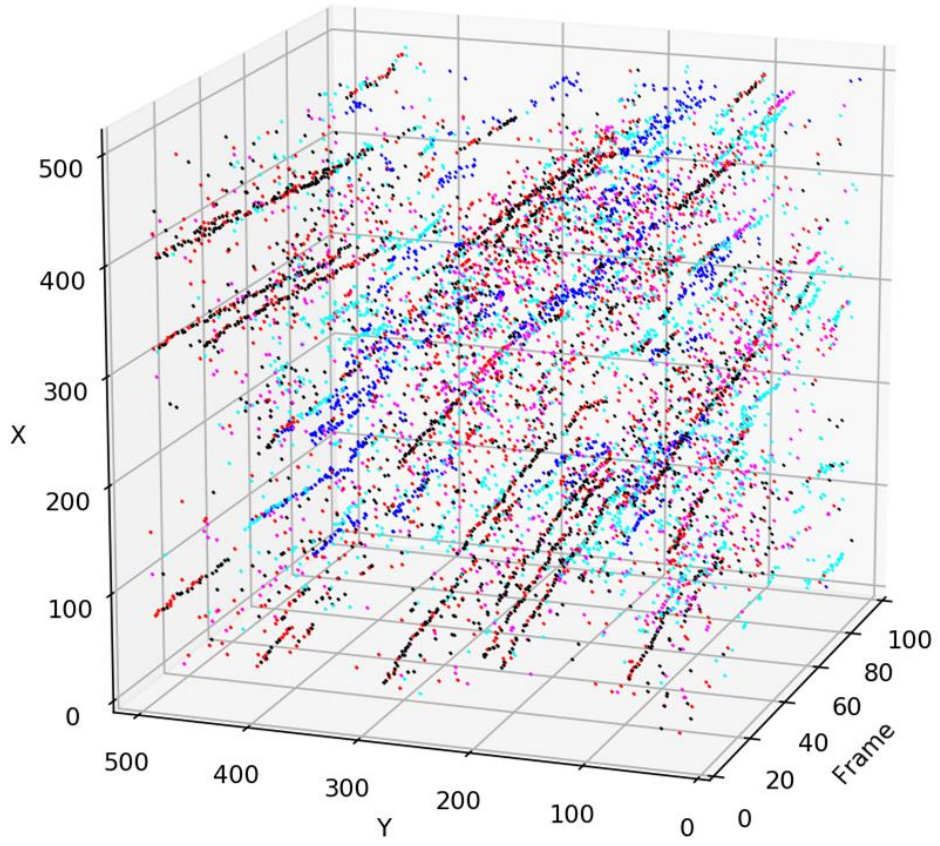




Sample: WS2
E-beam energy: 60 kV
Data by Ondrej Dyck (CNMS/ORNL)

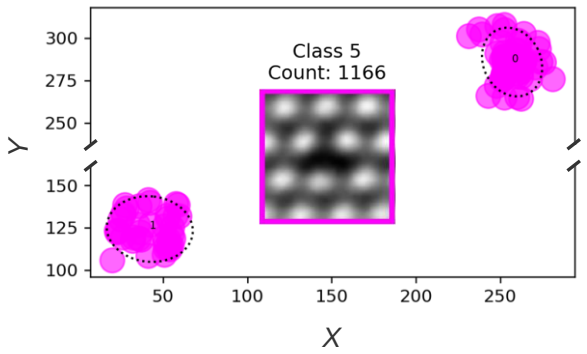
Learning Physics of Defects

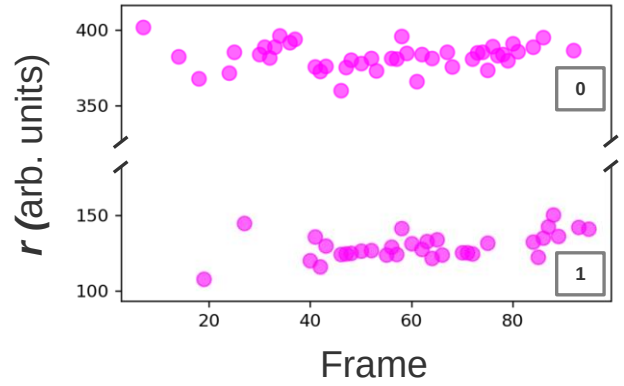
Spatio-temporal trajectories



Maksov *et al.*, npj Computational Materials 5, 12 (2019)

Diffusion parameters



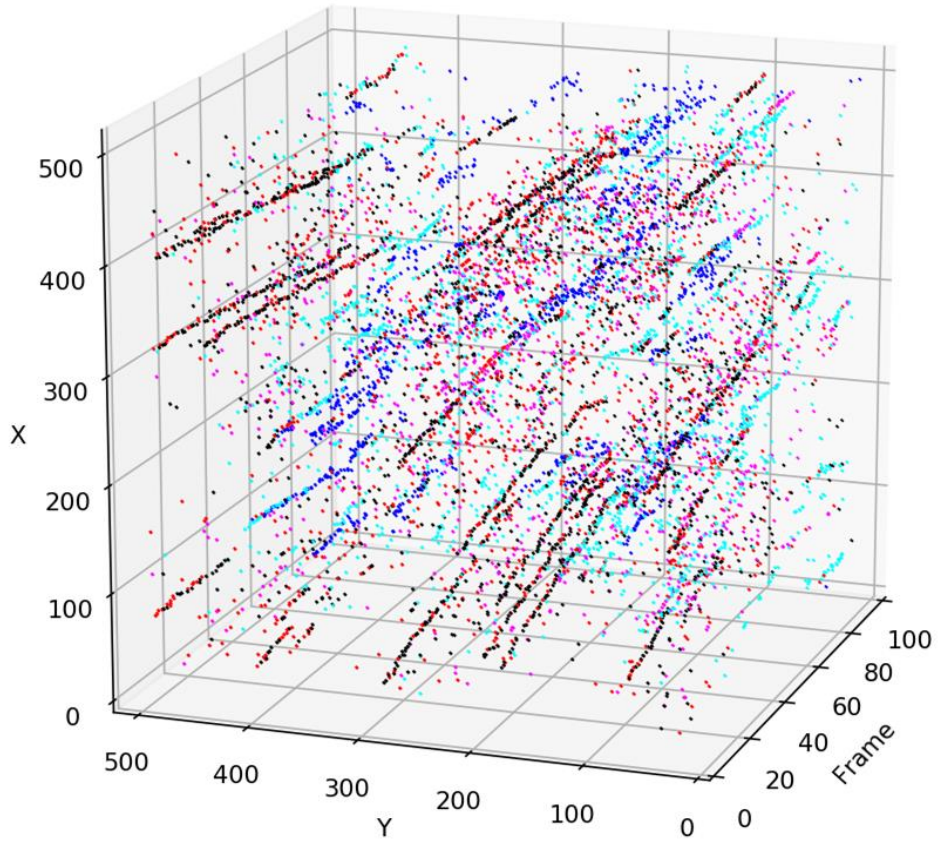


Diffusion coefficient: (3-6)×10^{-4} nm^2/s within 2D random walk approximation

- Identification of dominant point defects and their characteristic statistical behaviors
- Analysis of diffusion parameters for the selected defect species
- Study of transformation pathways and transition probabilities for composite defects

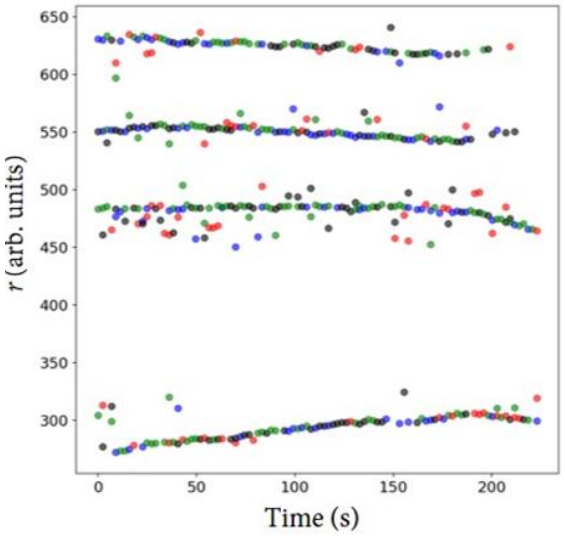
Learning Physics of Defects

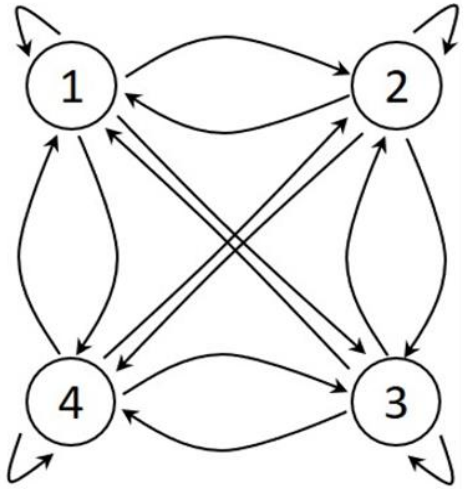
Spatio-temporal trajectories

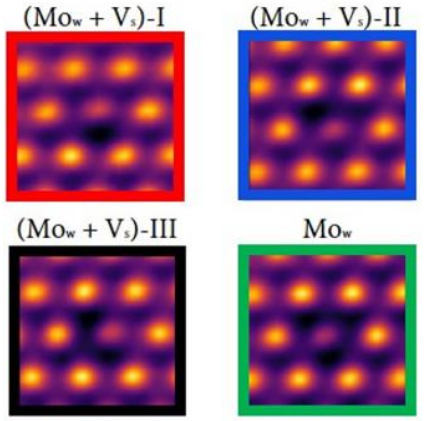


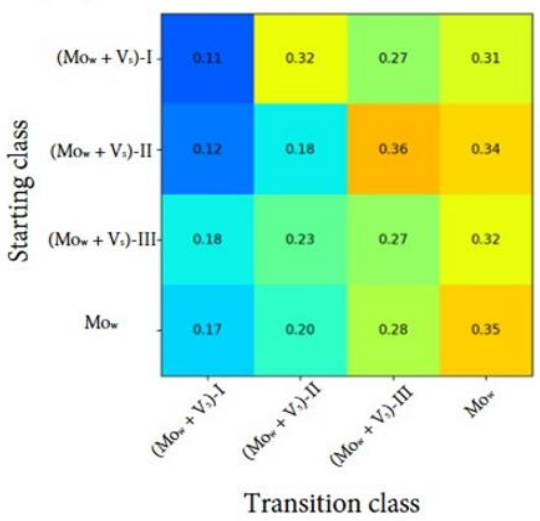
Maksov *et al.*, npj Computational Materials 5, 12 (2019)

Evolution of defects



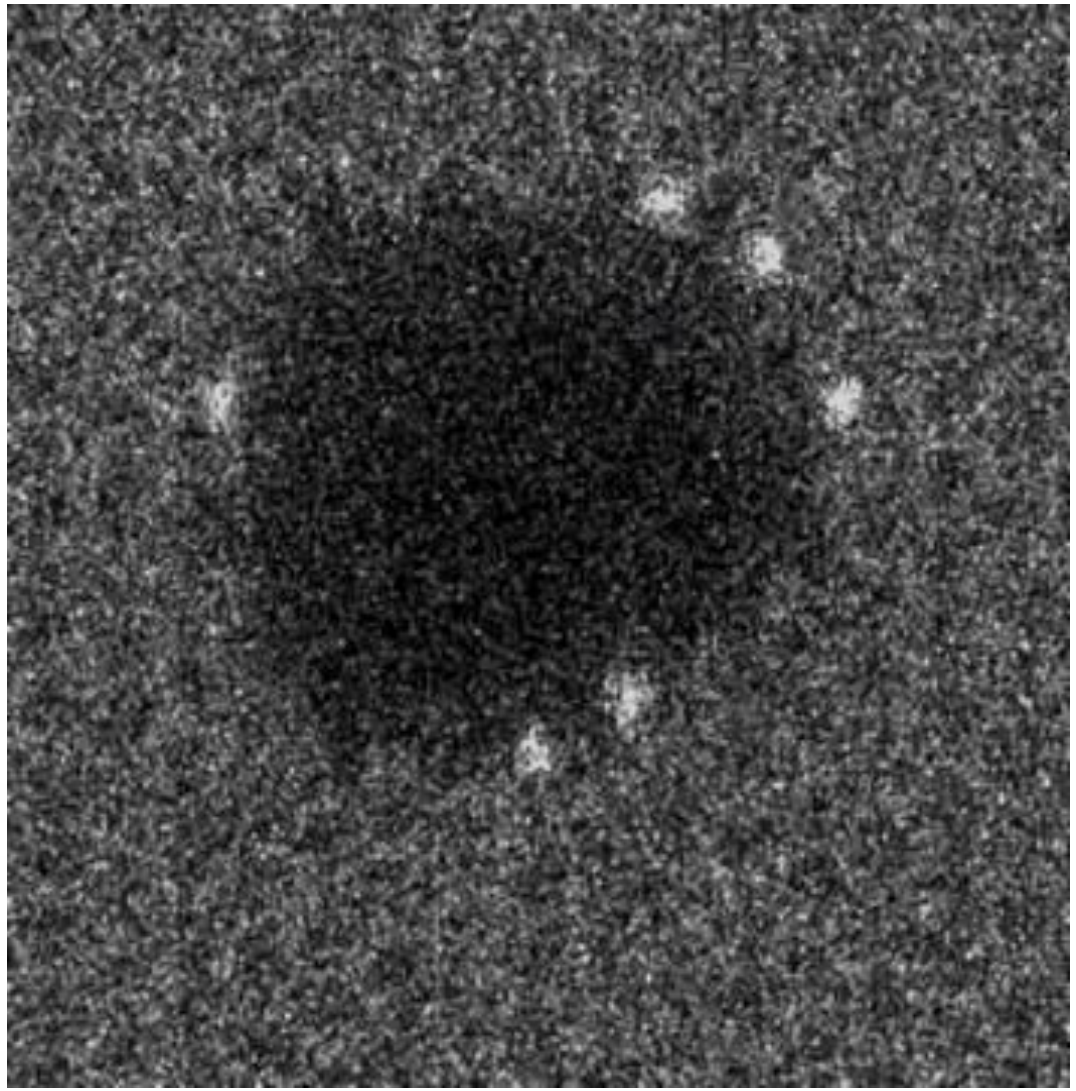




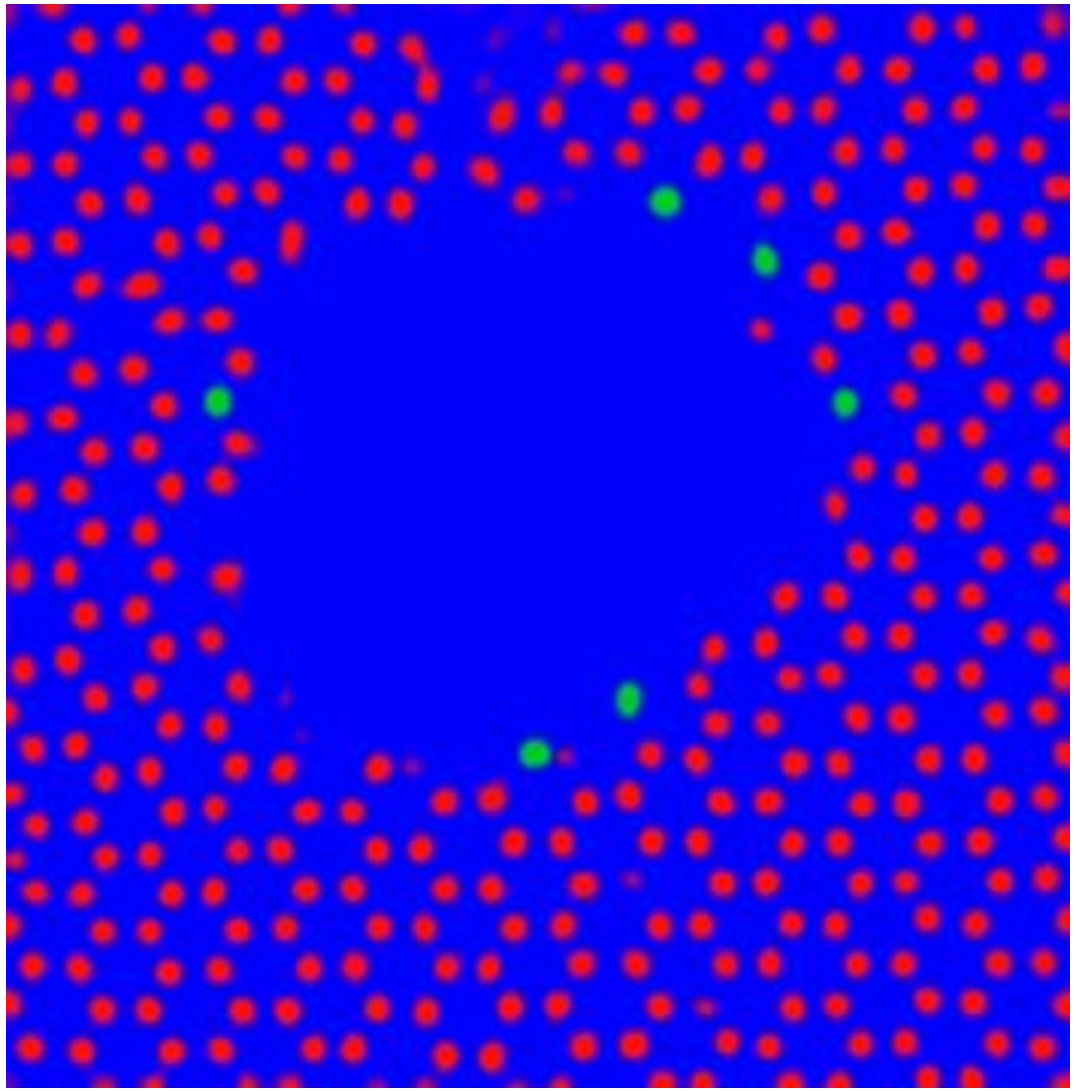


Si impurities on the graphene edge

Experimental data



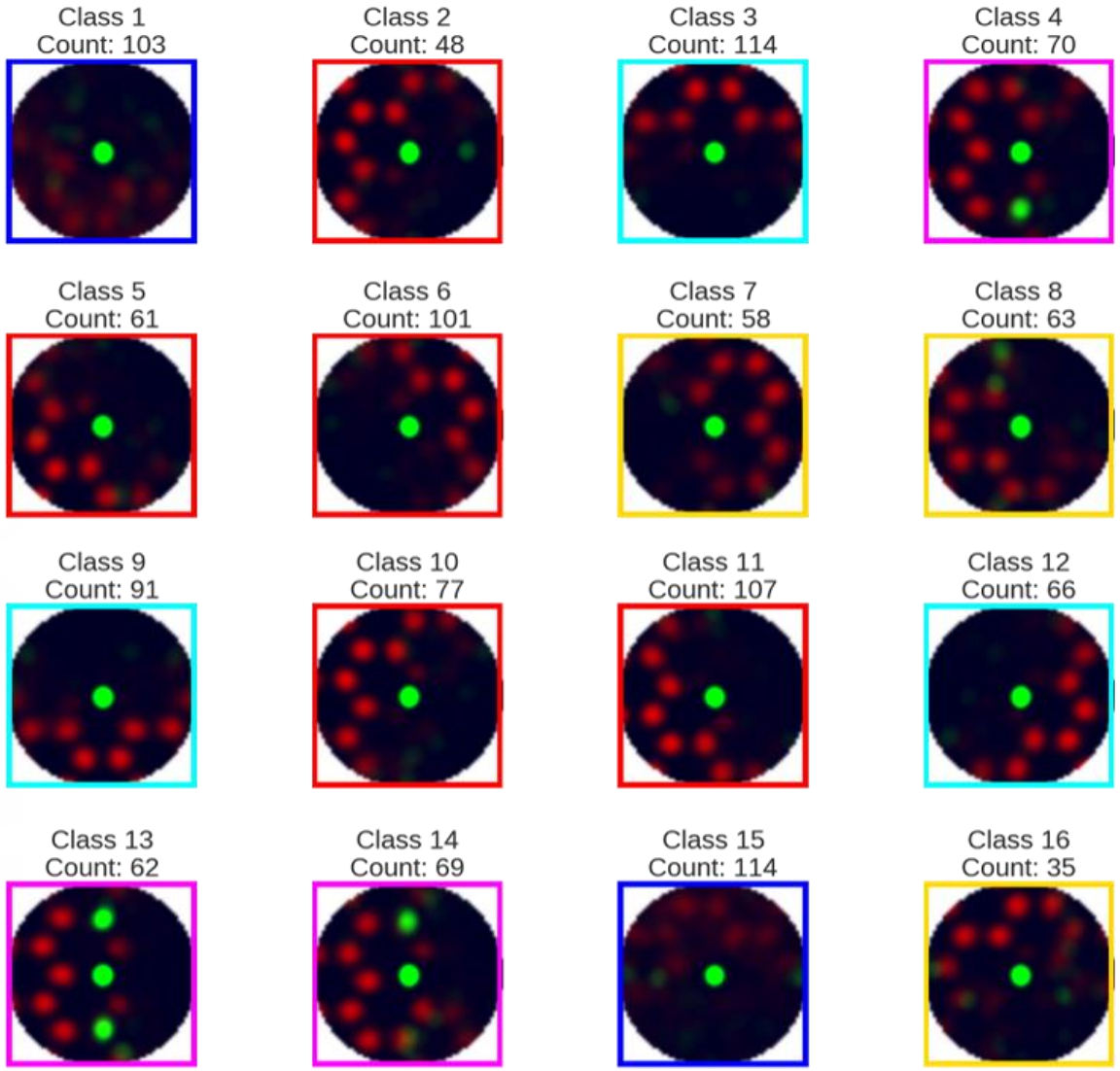
Network's output



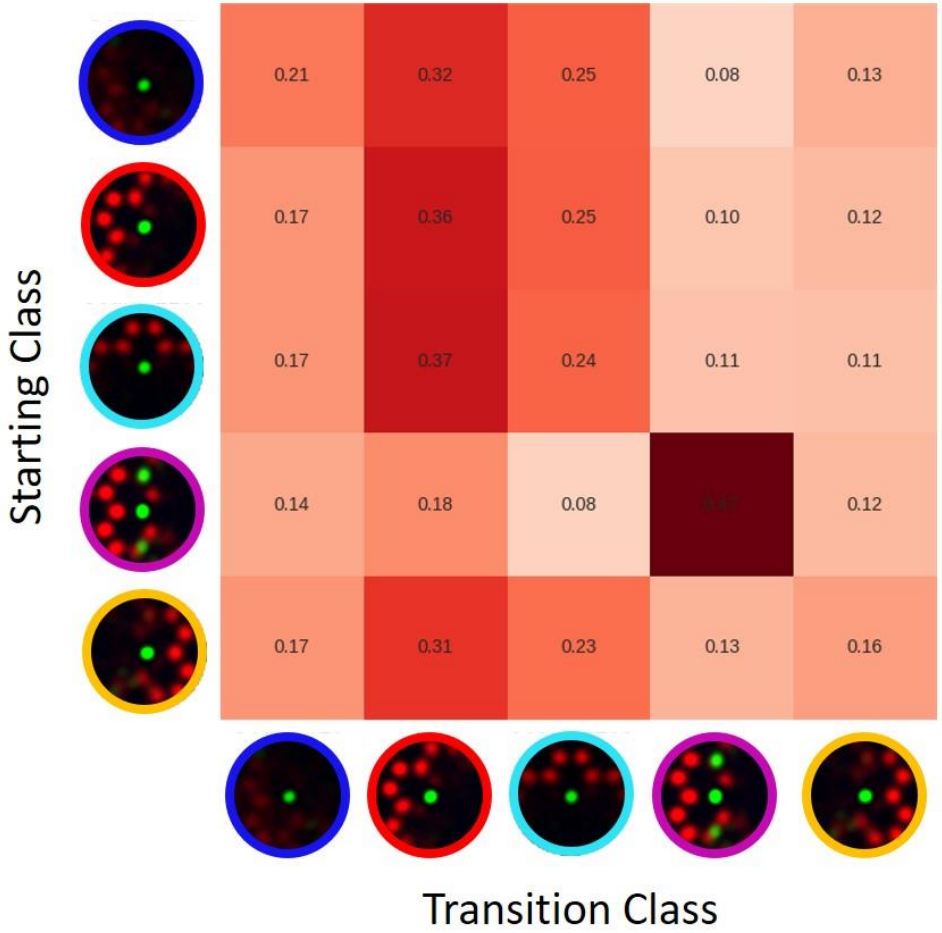
Ziatdinov *et al.*, ArXiv:1901.09322

Classification of edge states

Derived classes of Si-C edge configurations



Transition probabilities matrix



- Gaussian mixture model
- Discrete rotation symmetry
- Markov state analysis

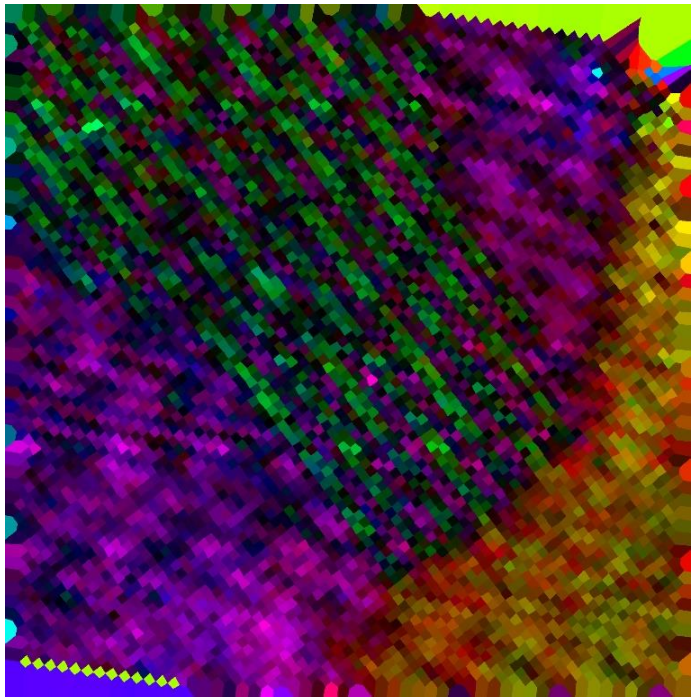
Ziatdinov *et al.*, ArXiv:1901.09322 (2019)

Crystalline materials

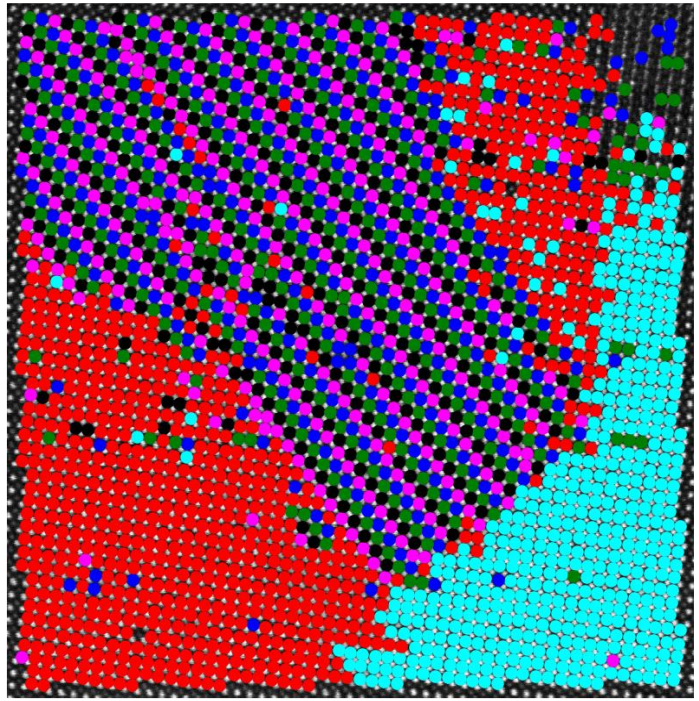
Experimental data (Ferroelectric LBFO)

Data by C. Nelson (MSTD ORNL)

Domain expert analysis



DCNN + k-means (~ 1-3 minutes)



CNN models work with patch by patch and sliding window approaches to image analysis. This allowed us to analyze high resolution images. CNN models were trained using Multislice simulations *and* simulations when atoms are modelled simply as 2D gaussians.

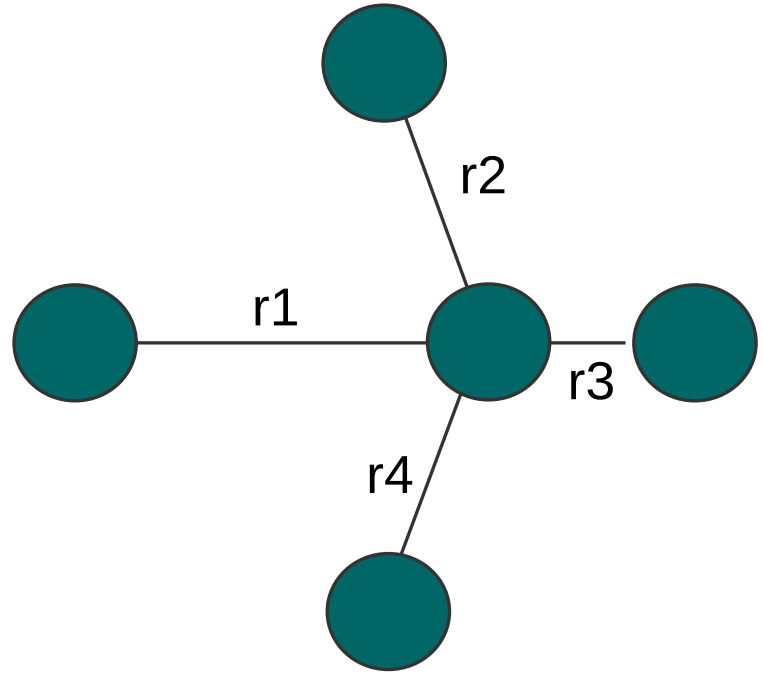
Local crystallography

For each atom, define nearest neighbors and generate array of the corresponding radius-vectors of the form

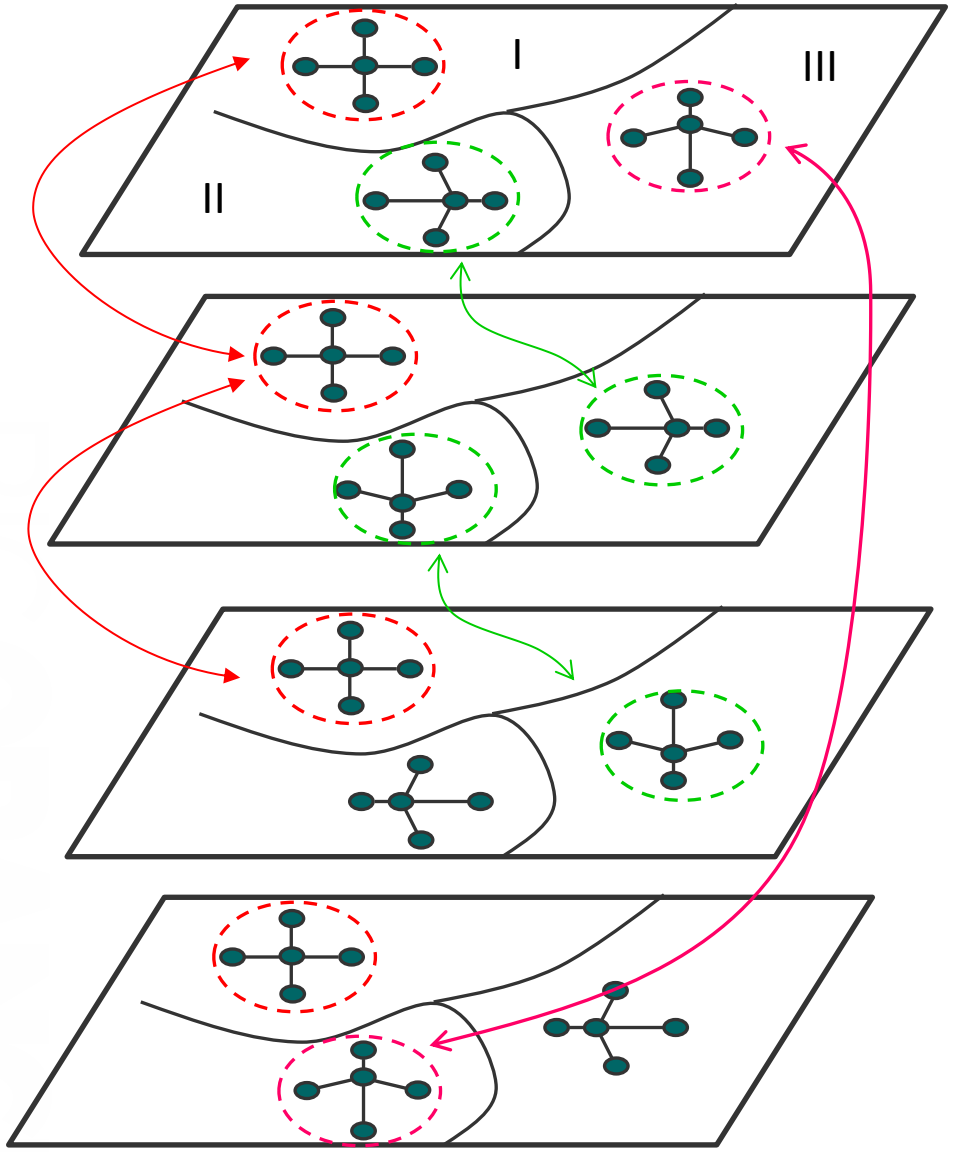
$NA_{ij} = (rx_1, ry_1, rx_2, ry_2, rx_3, ry_3, rx_4, ry_4)_{ij}$

Indexes 1,2,3,4 are chosen in the same sense for all atoms (generalization for different lattice and/or next coordination sphere obvious)

Then, phase/ferroic variant identification problem can be reduced to finding equivalent (in statistical sense) groups of nearest neighbors (for limited sense, we use point groups, for general sense, we use the spatial group and add translation symmetry operations, i.e. i -> i+1 and j -> j+1 for lattice doubling)



Local crystallography



A

A’

A’’

Same cluster in all replicas:
Non-ferroic phase

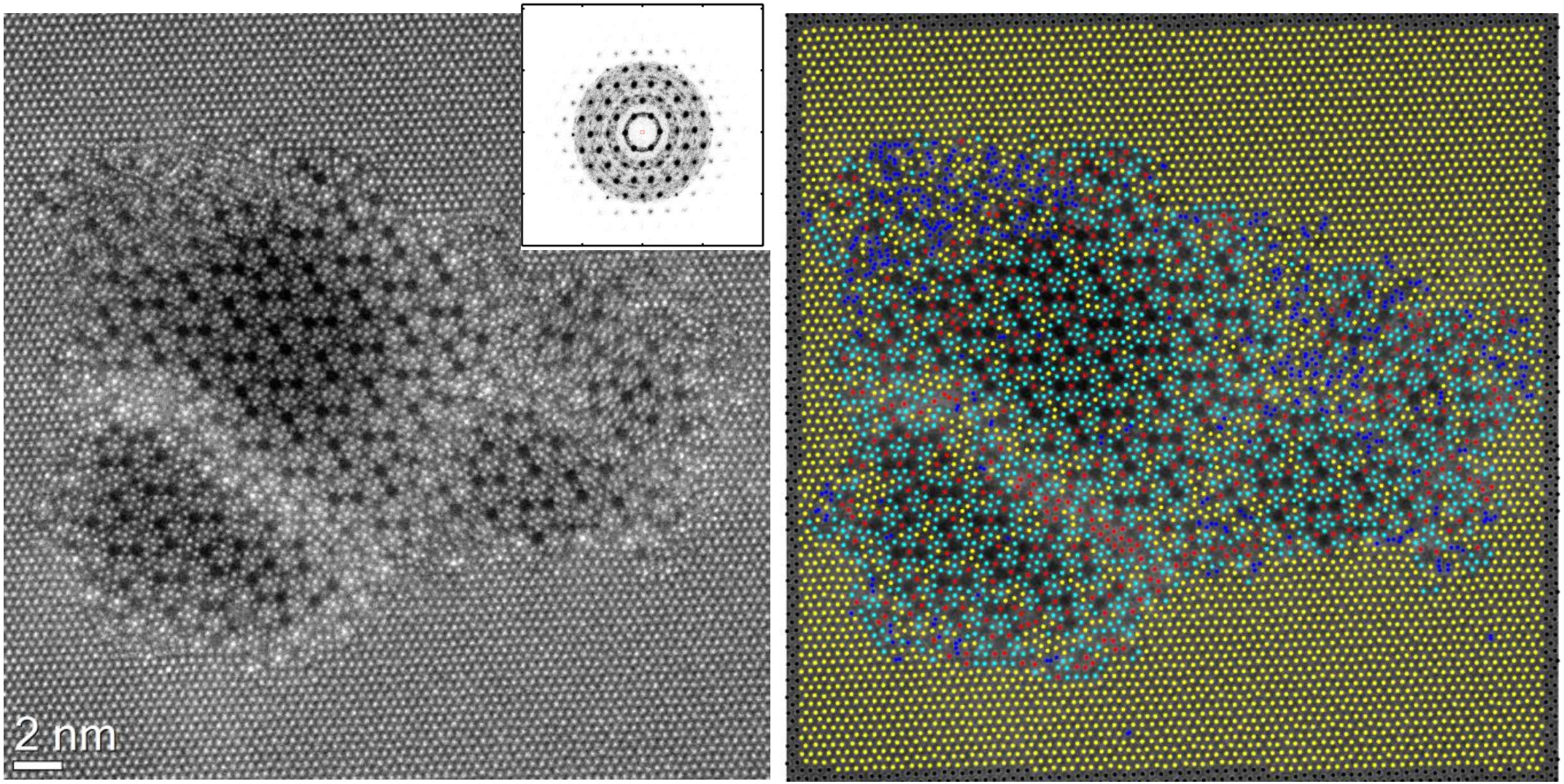
Form cluster with different regions
in different replicas:
Ferroic phase

Only some of the correspondences
are shown (but these are obvious)

A. Belianinov, Q. He, M. Kravchenko, S. Jesse, A. Borisevich, and S.V. Kalinin, *Identification of phases, symmetries, and defects through local crystallography*, Nat. Comm. **6**, 7801 (2015).

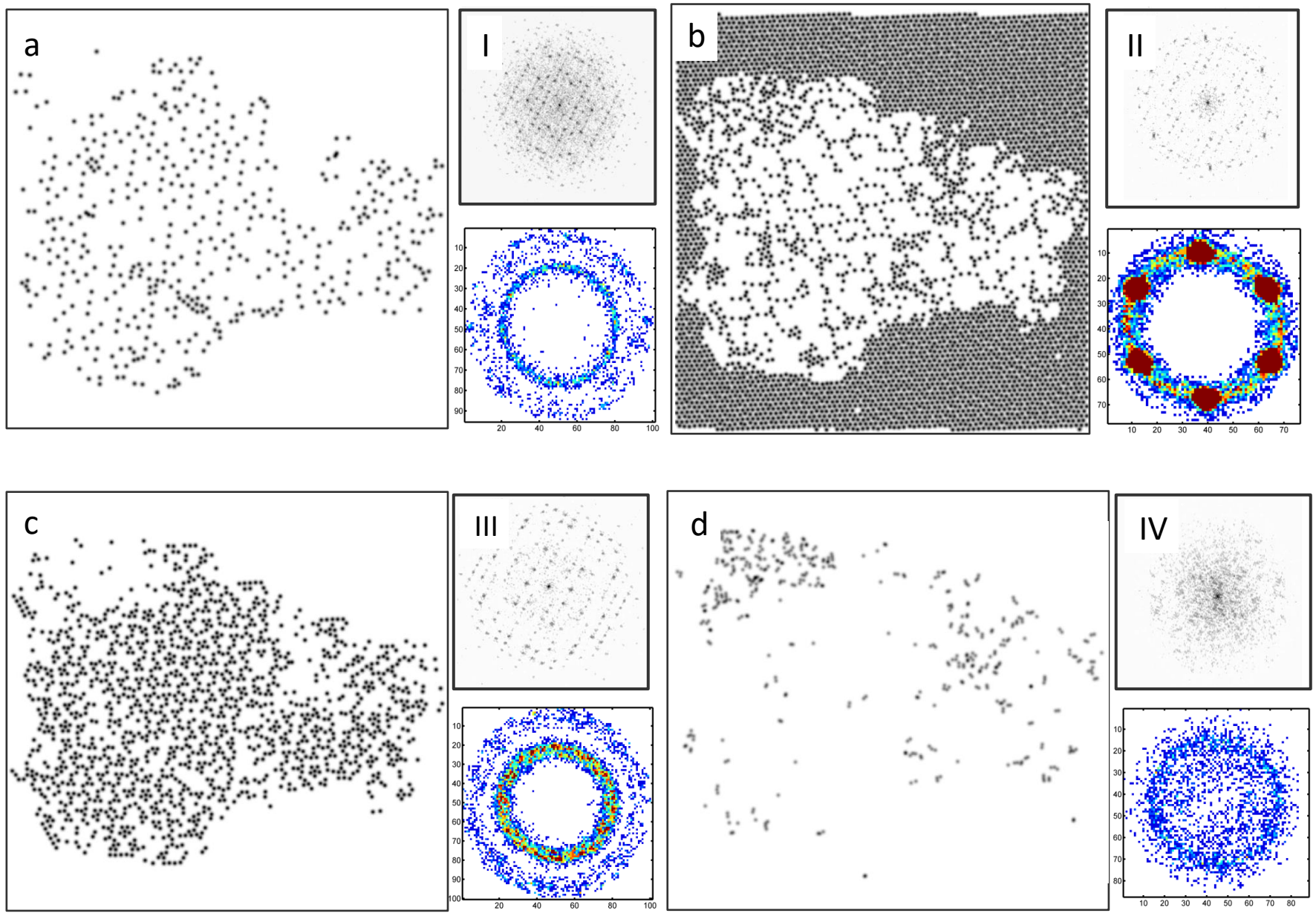
W. Lin, Q. Li, A. Belianinov, B.C. Sales, A. Sefat, Z. Gai, A.P. Baddorf, M. Pan, S. Jesse, and S.V. Kalinin, *Local crystallography analysis for atomically resolved scanning tunneling microscopy images*, Nanotechnology **24**, 415707 (2013).

Local crystallography: k-means

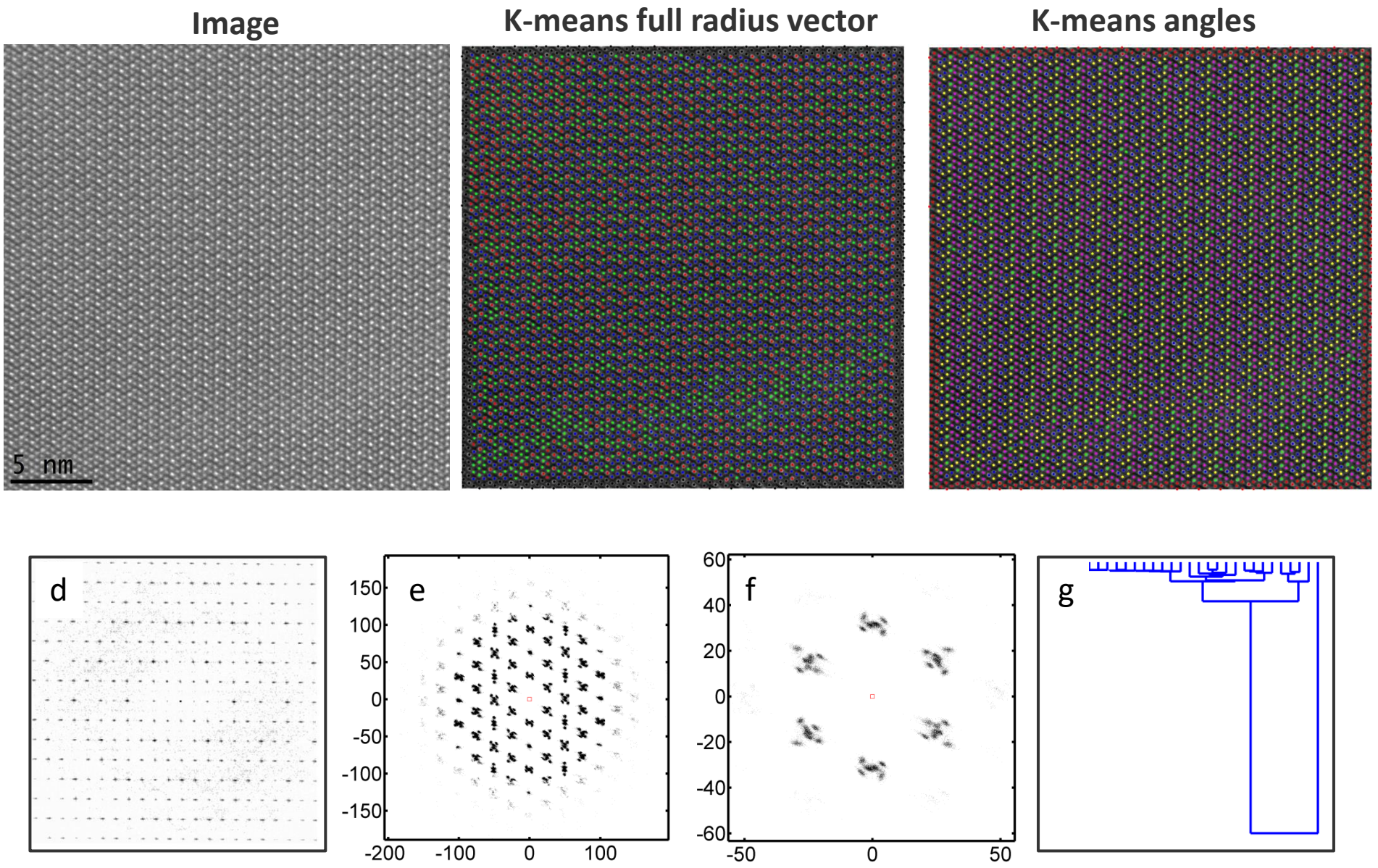


A. Belianinov, Q. He, M. Kravchenko, S. Jesse, A. Borisevich, and S.V. Kalinin, *Identification of phases, symmetries, and defects through local crystallography*, Nat. Comm. **6**, 7801 (2015).

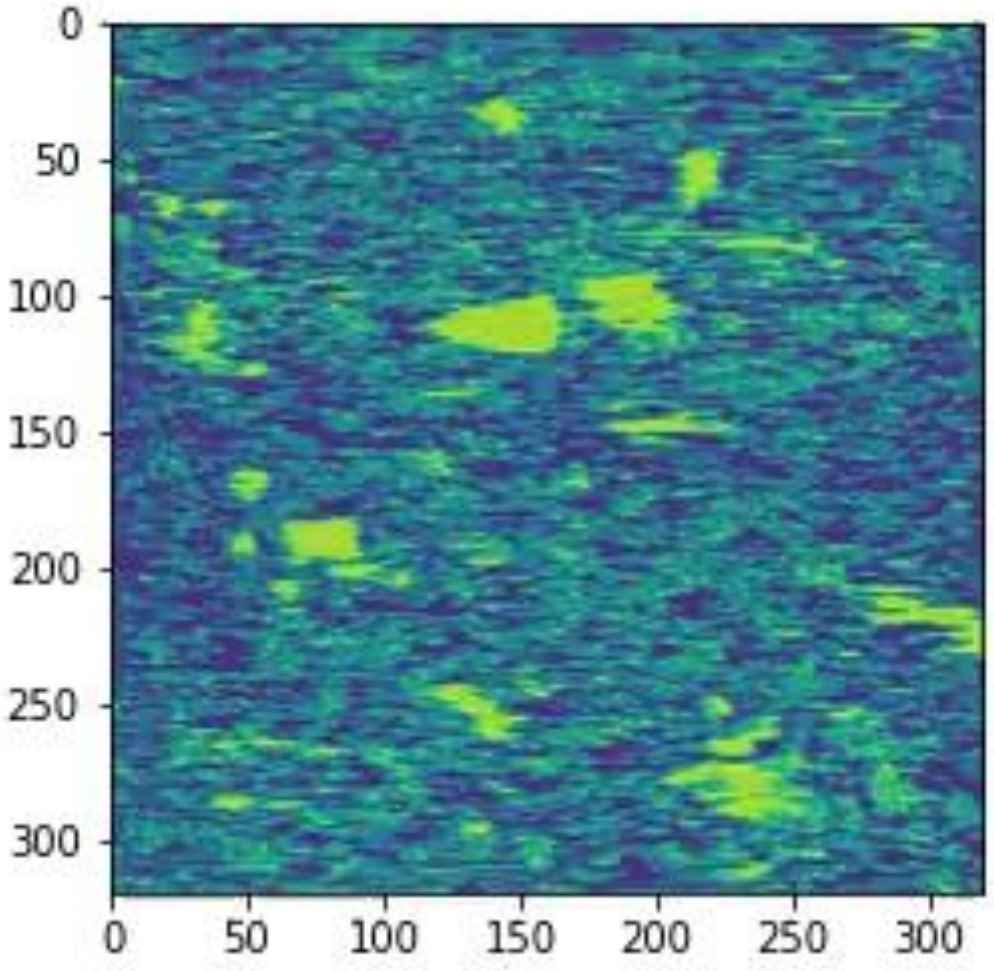
Local crystallography

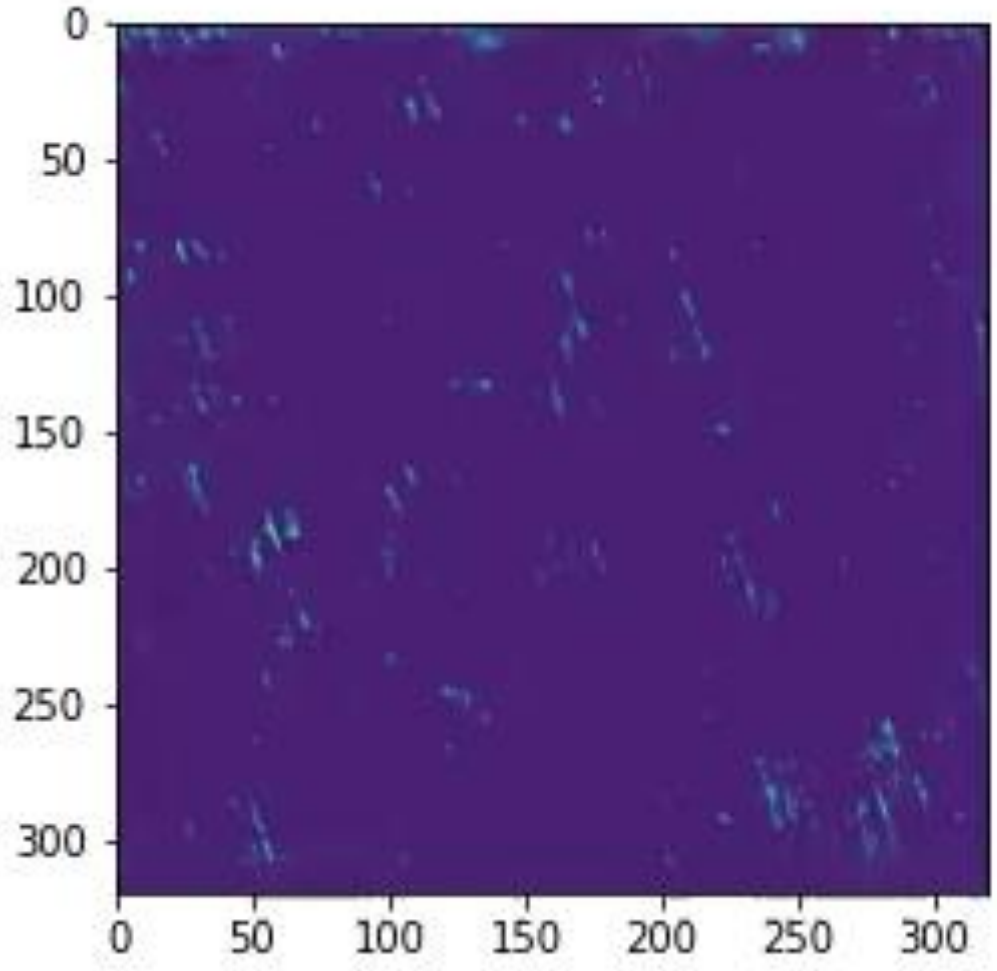


Local crystallography



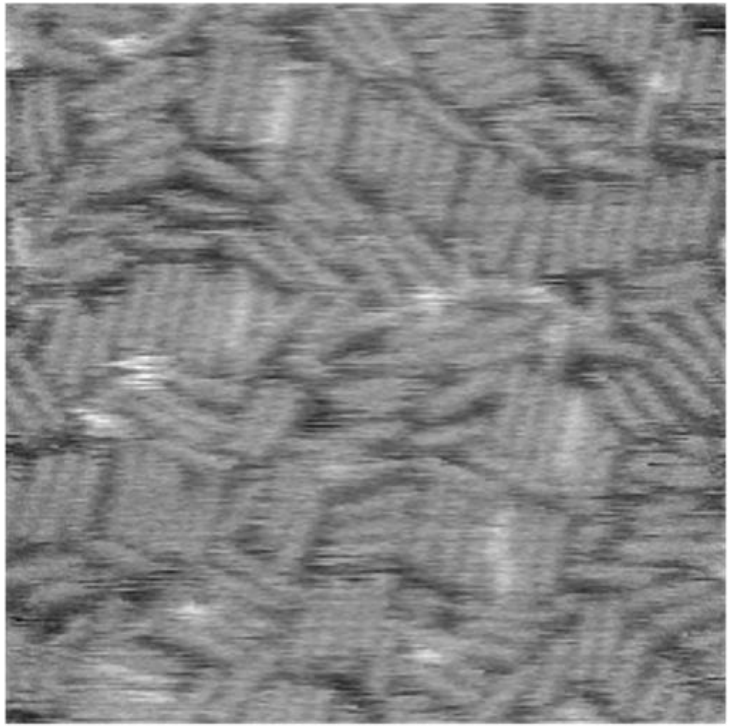
Protein assembly



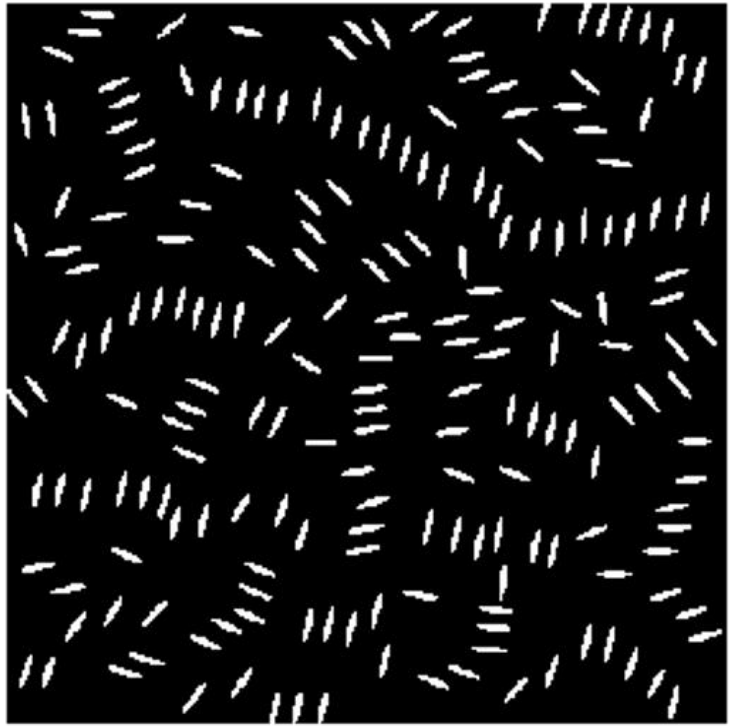


Protein assembly

Experimental image

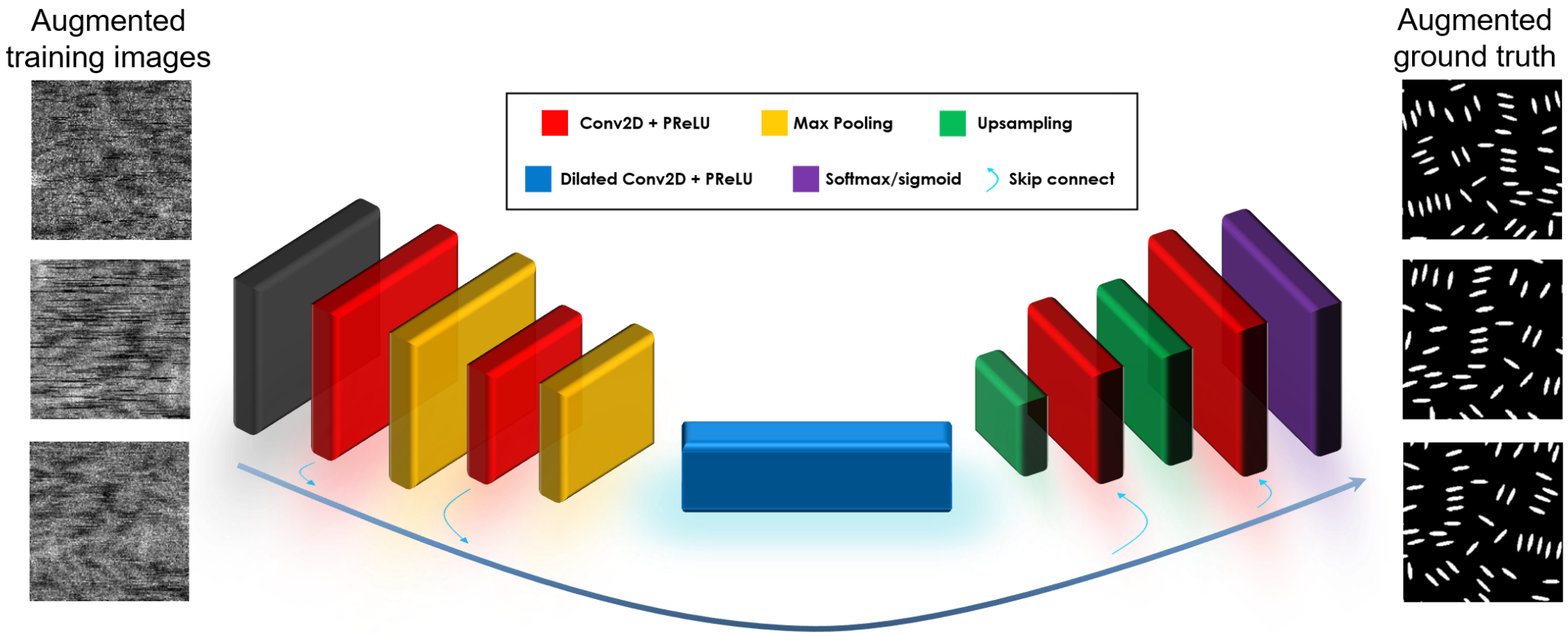


Ground truth



- Select an image where the position and orientation of all particles is well-defined (usually the last frame of an AFM movie)
- Perform manual labelling. The trained network should allow extraction of position and orientation of each particle.
- Generate a training set on-the-fly via data augmentation by random cropping, adding Gaussian and Poisson noises, artificial scan “scars”, zooming and resizing.

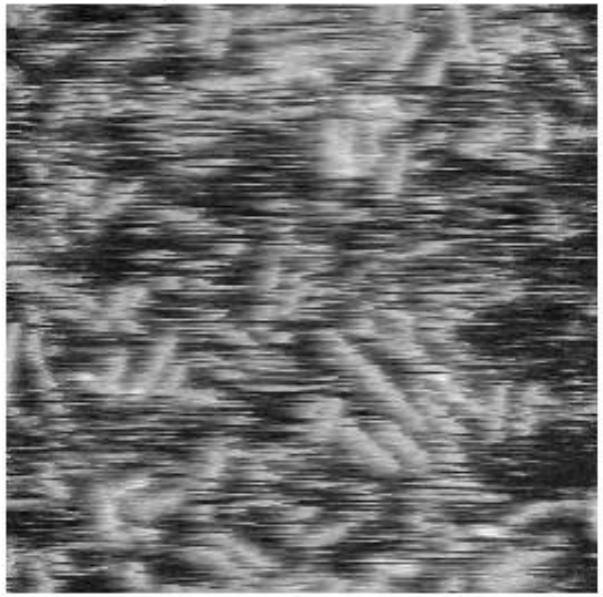
Protein assembly



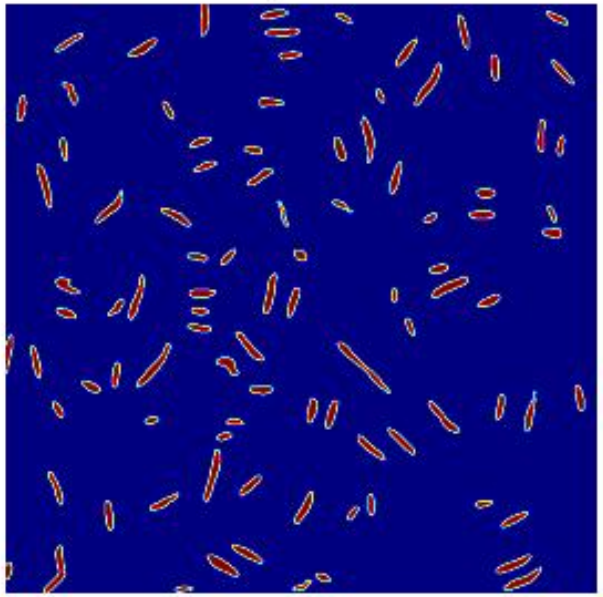
- Encoder-decoder type of architecture with spatial pyramid of dilated convolutions in the bottleneck layer
- Skip connections between encoder and decoder parts for mixing global and local information

Protein assembly

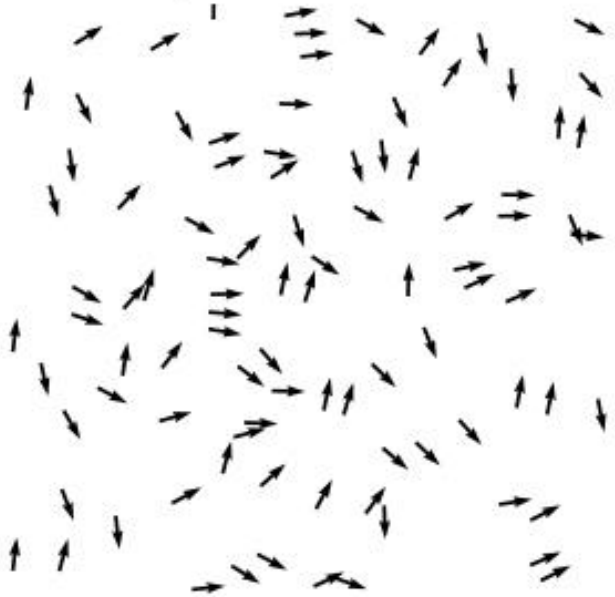
Experimental image



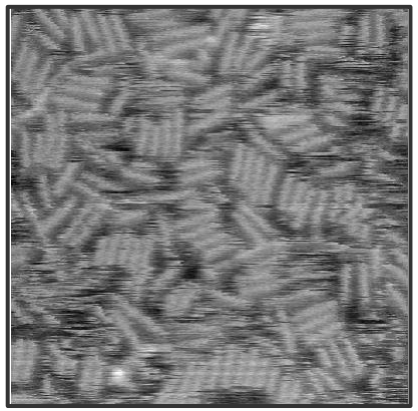
Model prediction

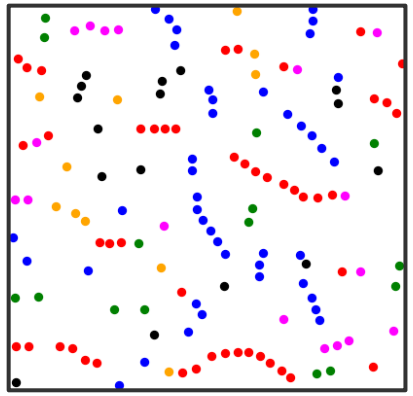


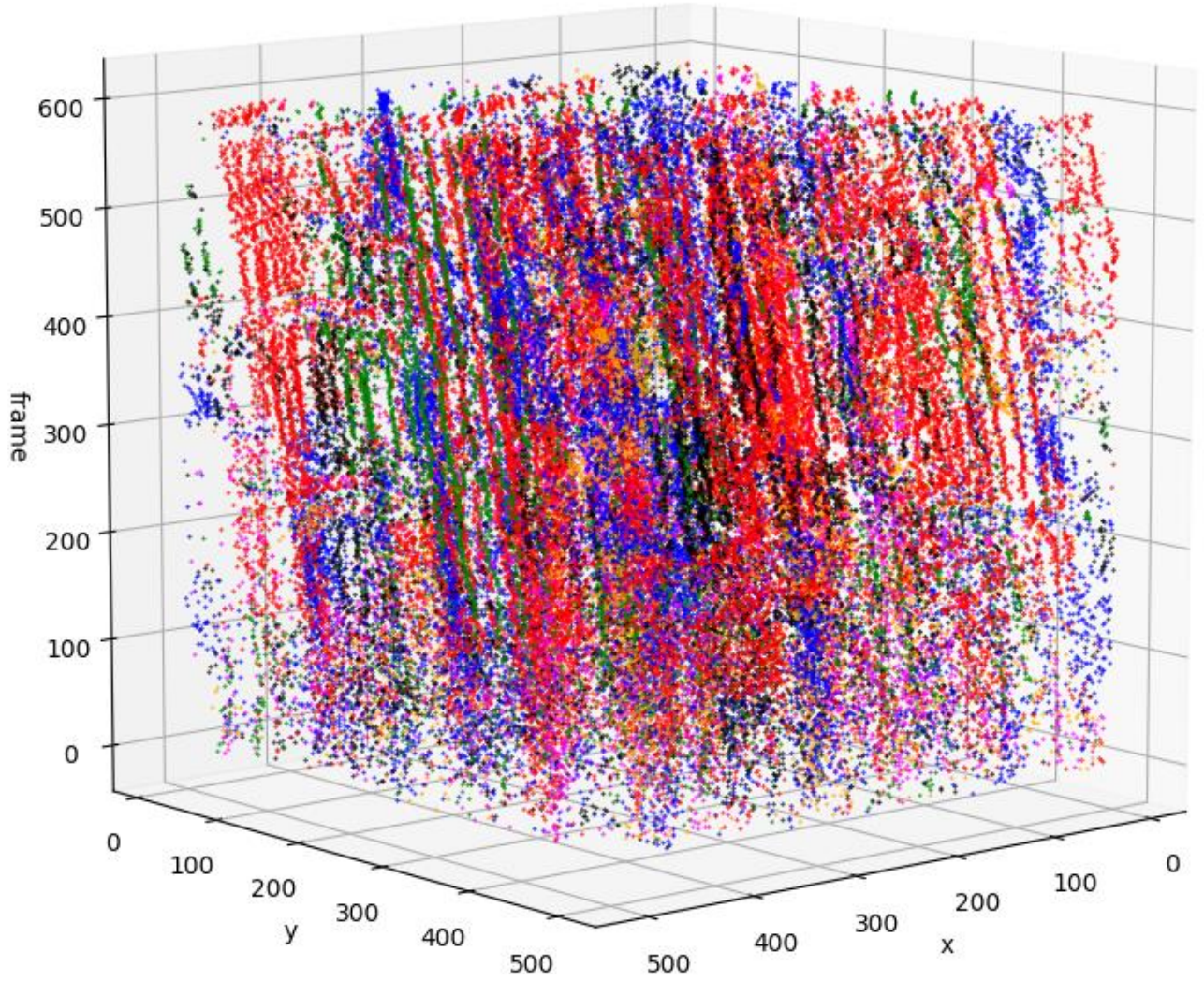
Proteins position and orientation



Protein assembly







Reconstructing spatio-temporal trajectories for all the detected particles in the movie, which show the evolution of different domains in time.

Learning programming:

1. Learn to program per se
2. Writing code used in project
3. Making code that others will use
4. Using codes written by others
5. Being a part of the team developing code
6. Leading the team developing code

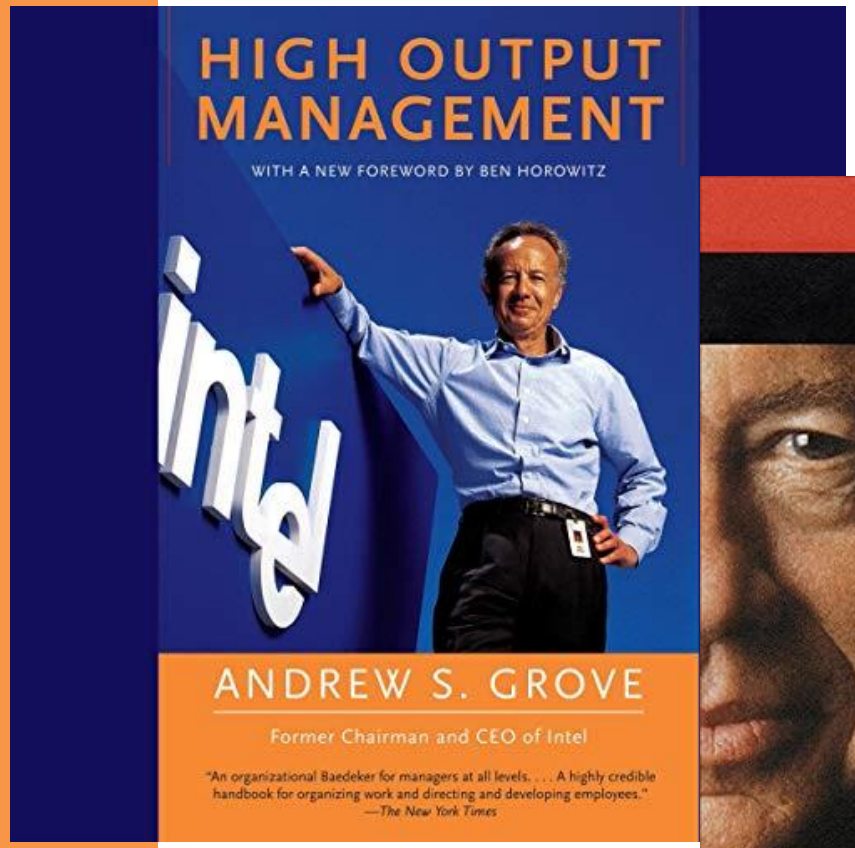
Not a good classification:

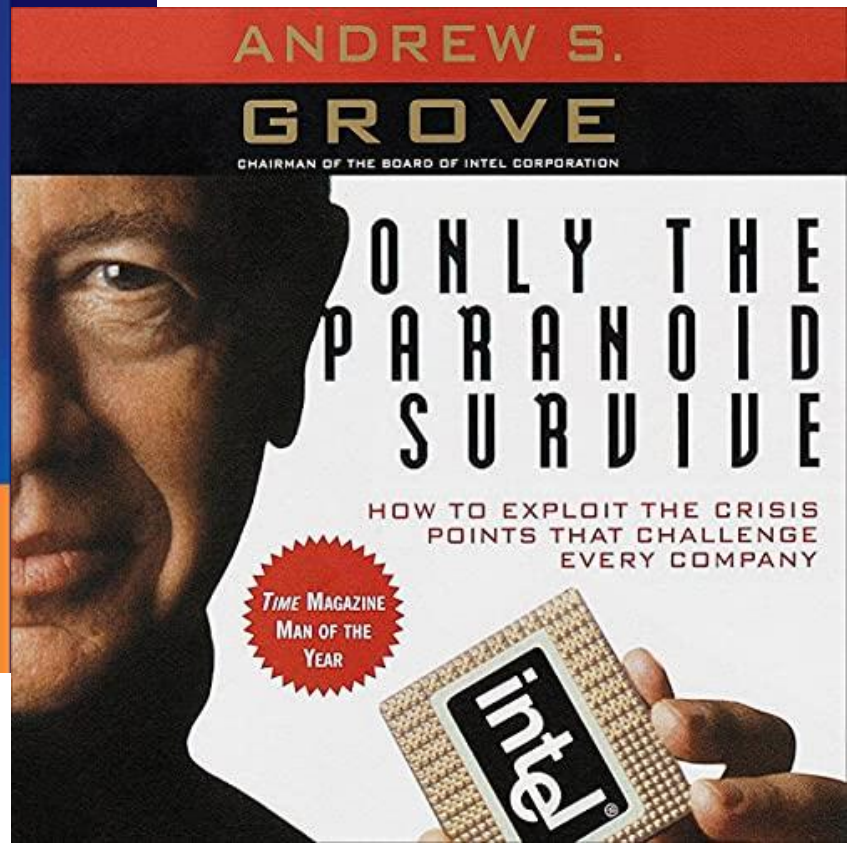
Animals are divided into

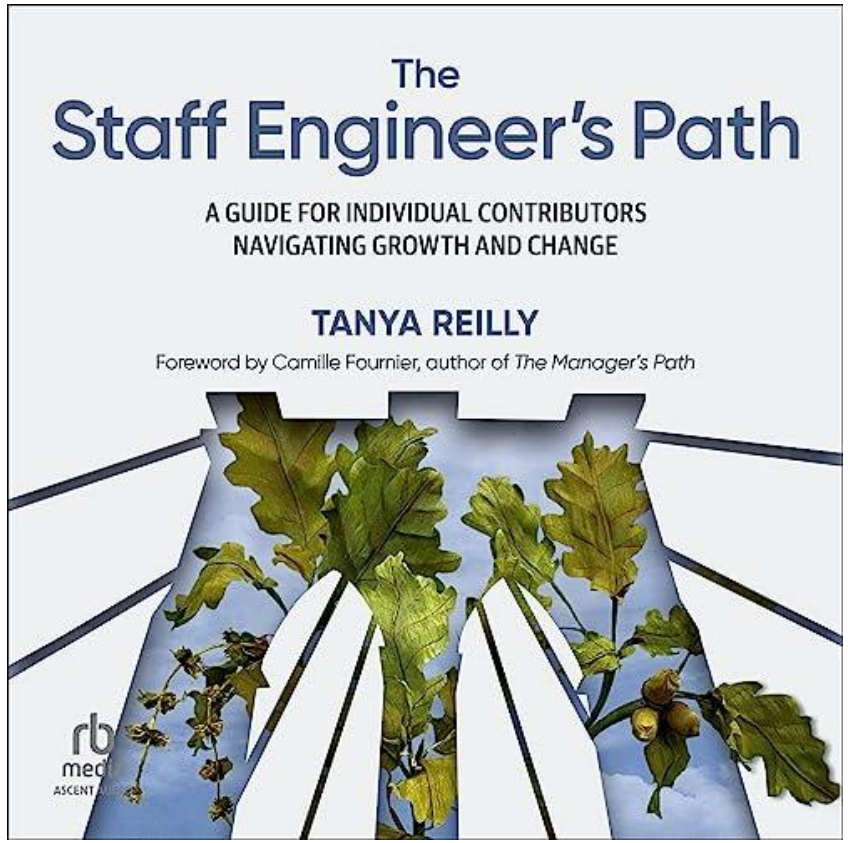
1. those that belong to the Emperor,
2. embalmed ones,
3. those that are trained,
4. suckling pigs,
5. mermaids,
6. fabulous ones,
7. stray dogs,
8. those that are included in this classification,
9. those that tremble as if they were mad,
10. innumerable ones,
11. those drawn with a very fine camel's hair brush,
12. others,
13. those that have just broken a flower vase,
14. those that resemble flies from a distance.

Celestial Emporium of Benevolent Knowledge – Jorge Luis Borges's fictional taxonomy of animals from his 1942 short story *The Analytical Language of John Wilkins*.

What's after grad school?







Amazon Principal Engineers Tenets

As Amazon's most senior individual contributors, Principal Engineers work on Amazon's hardest problems. As they navigate these complex and ambiguous challenges, the Principal Engineering Community uses the following set of tenets to guide them. Tenets are a key part of Amazon's peculiar culture helping to bring Amazonians together and focused on achieving our mission and vision to be Earth's most customer-centric company.

EXEMPLARY PRACTITIONER

Principal Engineers are hands-on and lead by example. We deliver artifacts that set the standard for engineering excellence, from designs to algorithms to implementations. Only by being close to the details can we earn the respect needed to be effective technical leaders.

TECHNICALLY FEARLESS

Amazon’s startup culture does not admit the luxury of conservatism. Principal Engineers tackle intrinsically hard problems, venturing beyond comfortable approaches when necessary. We acquire expertise as needed, pioneer new spaces, and inspire others as to what’s possible.

Amazon Principal Engineers Tenets

LEAD WITH EMPATHY

Principal Engineers shape an inclusive engineering culture where others are heard, feel respected, and are empowered. We are conscious of how our words and demeanor impact others, especially those with less influence; we take responsibility for that impact, intentional or otherwise. Our work builds productive relationships across teams and disciplines, and across a wide range of life experiences.

BALANCED AND PRAGMATIC

Principal Engineers are pragmatic problem solvers. We apply judgment and experience to balance trade-offs between competing interests. We simplify processes and technologies while advocating a long-term view.

ILLUMINATE AND CLARIFY

Principal Engineers bring clarity to complexity and demonstrate smart ways to simplify. We frame each problem in its customer and business context and boil it down to its essence. We probe assumptions, illuminate pitfalls, and foster shared understanding. We accelerate progress by driving crisp and timely decisions.

Amazon Principal Engineers Tenets

FLEXIBLE IN APPROACH

Principal Engineers adapt our approach to meet the needs of the team, project, and product. We solicit differing views and are willing to change our minds as we learn more. We recognize there are often many viable solutions, and that sometimes the best solution is to solve a different problem, or to not solve the problem at all.

RESPECT WHAT CAME BEFORE

Principal Engineers are grateful to our predecessors. We appreciate the value of working systems and the lessons they embody. We understand that many problems are not essentially new.

LEARN, EDUCATE, AND ADVOCATE

Principal Engineers are constantly learning. We seek technical knowledge and educate the entire organization about trends, technologies, and approaches. We combine vision and discretion to drive fruitful and even game-changing technology choices.

HAVE RESOUNDING IMPACT

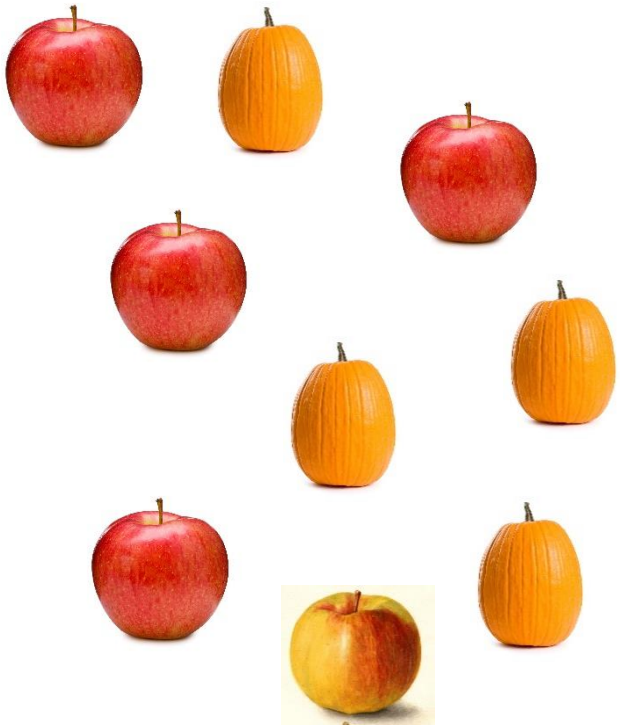
“Deliver Results” is a low bar for a Principal Engineer. Without seeking the spotlight, Principal Engineers make a lasting impact that echoes through the technology, the product, and the company. We amplify our impact by aligning teams toward coherent architectural strategies.

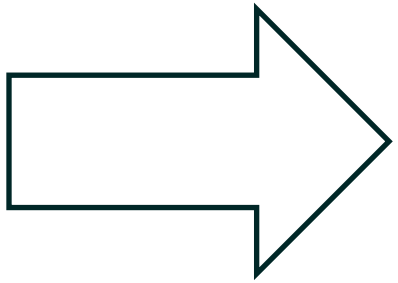
Classification Workflow

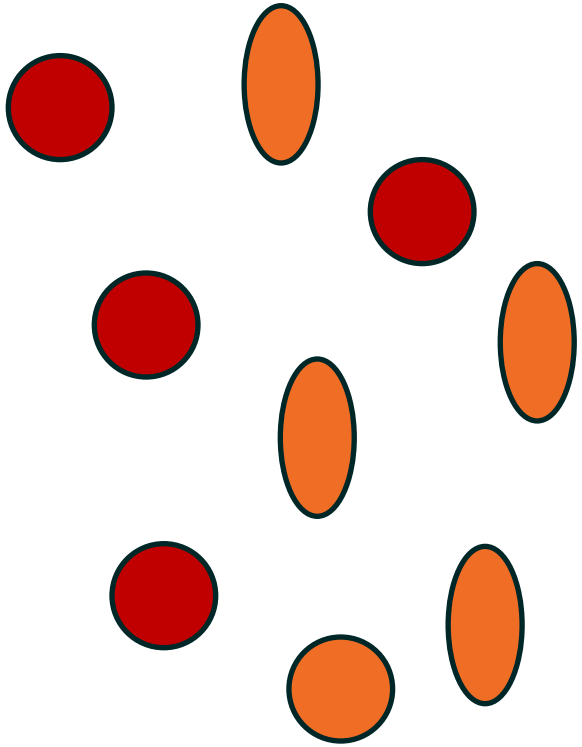
1. Selecting features and collecting labeled training examples
2. Choosing a performance metric
3. Choosing a learning algorithm and training a model
    - Can we understand how it works?
    - Does it have any hyperparameters
    - How well does it generalize to new data?
    - How expensive is it?
4. Evaluating the performance of the model
5. Changing the settings of the algorithm and tuning the model.

From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

Feature Engineering



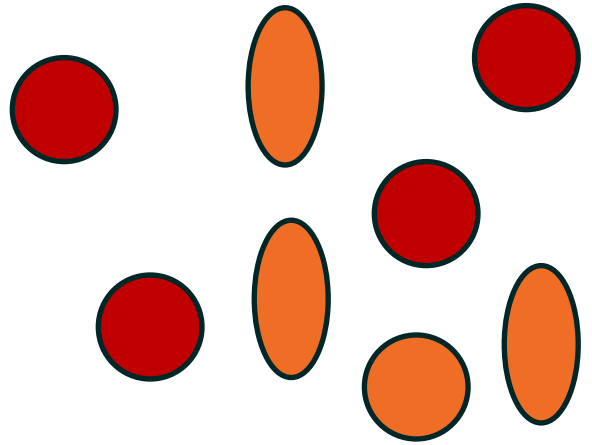


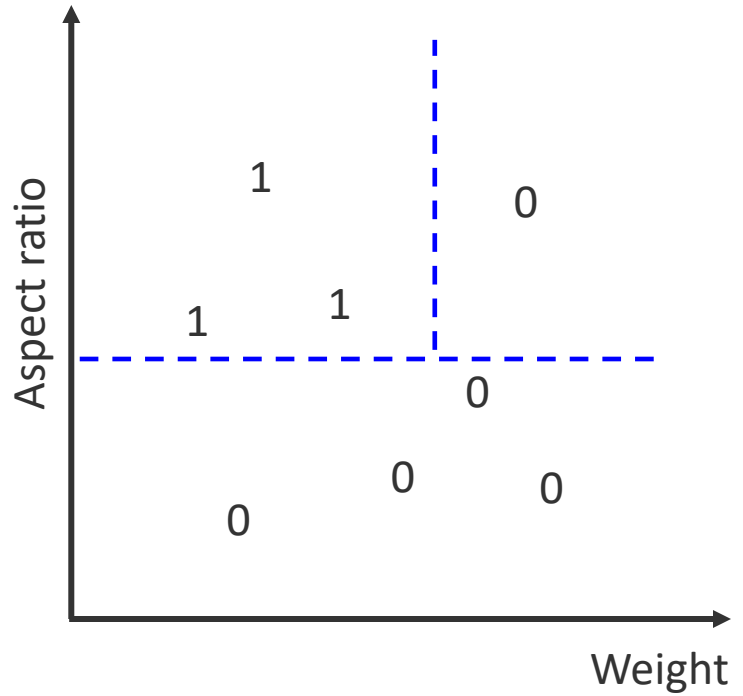


Features: color, shape, weight, aspect ratio

Classification and Regression Tree (CART)

- Create a set of features (shape, color, weight)
- Select a splitting criterion (likelihood):
  - Initialization: create a tree with one node containing all the training data
  - Splitting: find the best way for splitting each terminal node. Split the one terminal node that results in the greatest increase in the likelihood
  - Stopping: if each leaf node contains sample from the same class, stop. Otherwise, continue splitting
  - Pruning: use an independent test set or cross-validation to prune the tree.





How do we split?

Information gain:

$$IG(D_p, f) = I(D_p) - \sum_{j=1}^{m} \frac{N_j}{N_p} I(D_j)$$

- f is the feature to perform the split
- D_p and D_j are the dataset of the parent and jth child node
- I is our **impurity** measure
- N_p is the total number of training examples at the parent node
- N_j is the number of examples in the jth child node

Binary split:

$$IG(D_p, f) = I(D_p) - \frac{N_{left}}{N_p} I(D_{left}) - \frac{N_{right}}{N_p} I(D_{right})$$

From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

What are impurity measures?

Entropy: $$I_H(t) = -\sum_{i=1}^{c} p(i|t) \log_2 p(i|t)$$

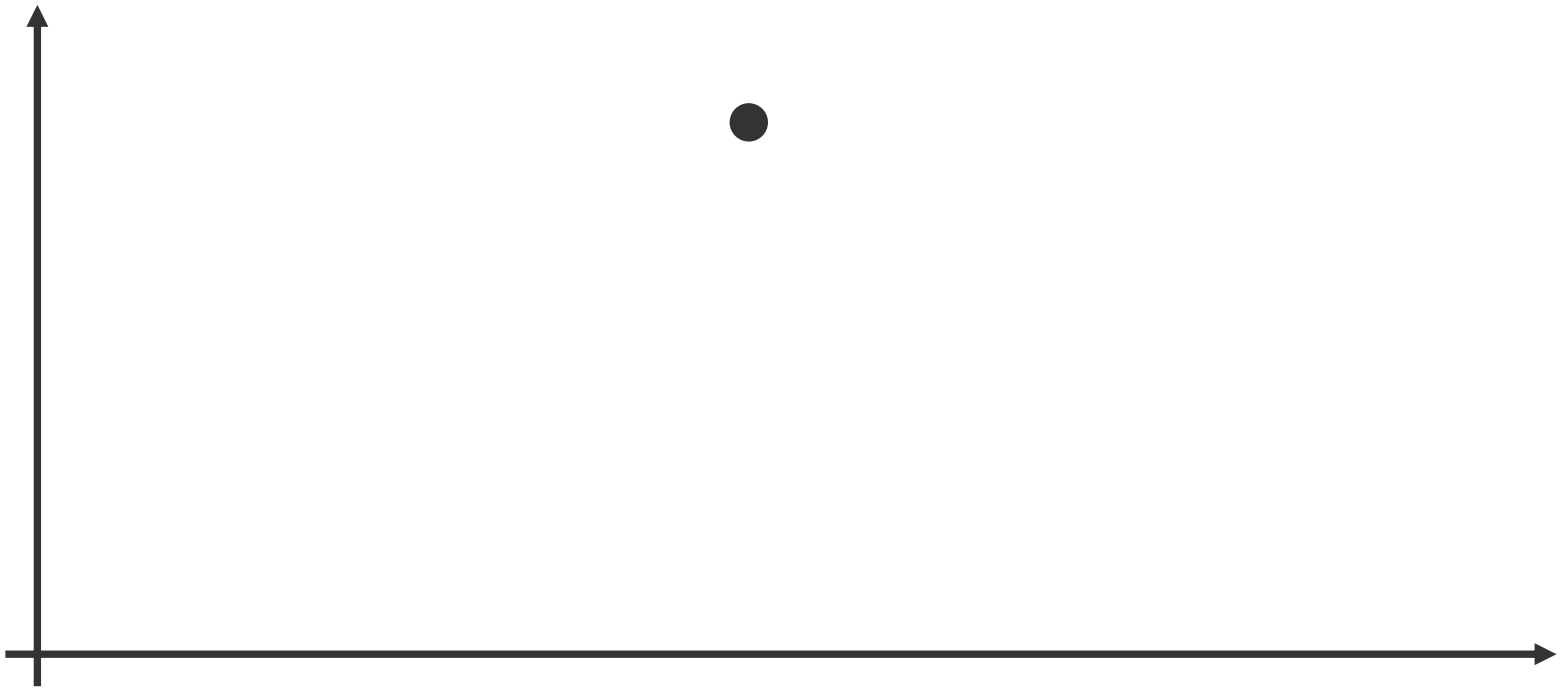
Gini impurity: $$I_G(t) = \sum_{i=1}^{c} p(i|t)\,(1 - p(i|t)) = 1 - \sum_{i=1}^{c} p(i|t)^2$$

Classification error: $$I_E(t) = 1 - \max\{p(i|t)\}$$

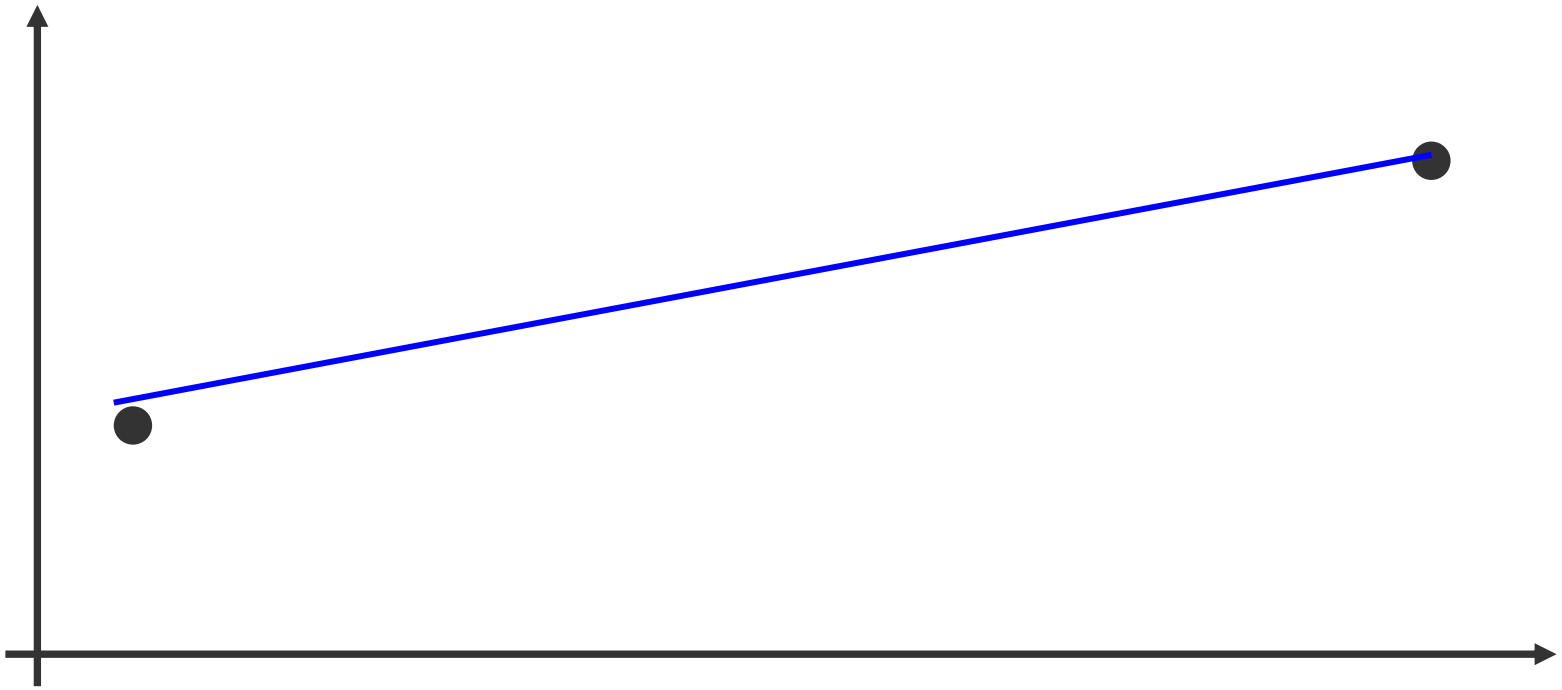
- $p(i|t)$ is the proportion of the examples that belong to class i for a node, t.

From S. Raschka, Machine Learning with PyTorch and Scikit-Learn

Physics vs. data science

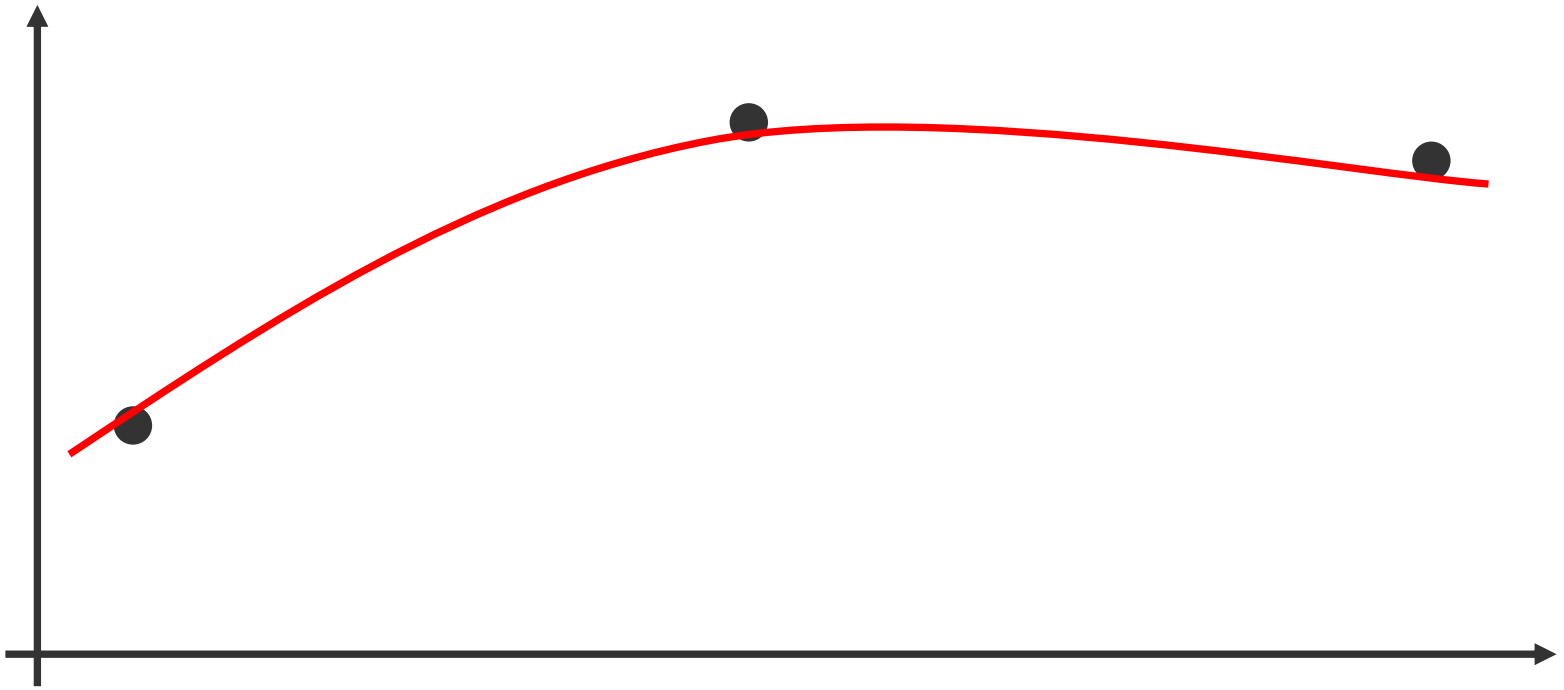


Physics vs. data science



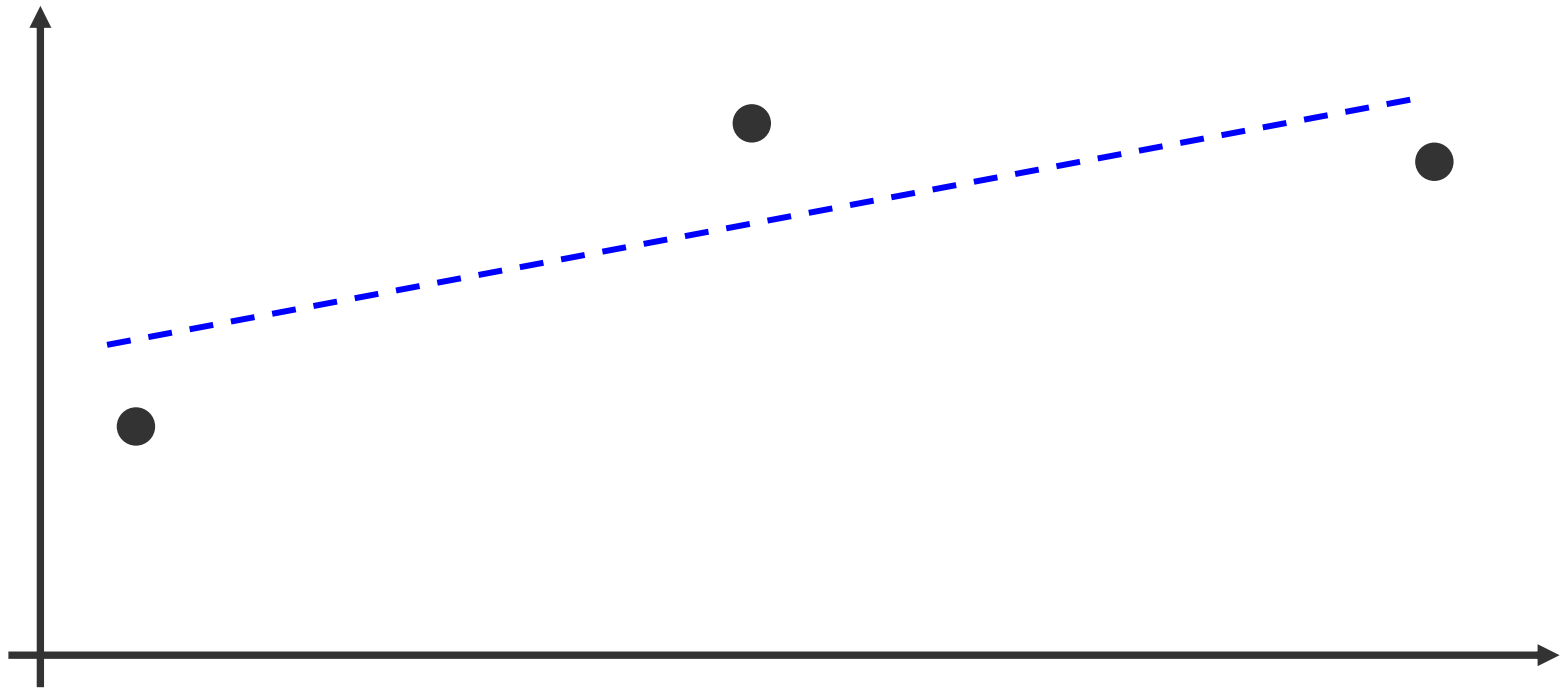
- If we have 2 data points, we “naturally” use linear model
- What should we use if we have three data points? Parabola or linear?
- What if we have one data point?

Physics vs. data science



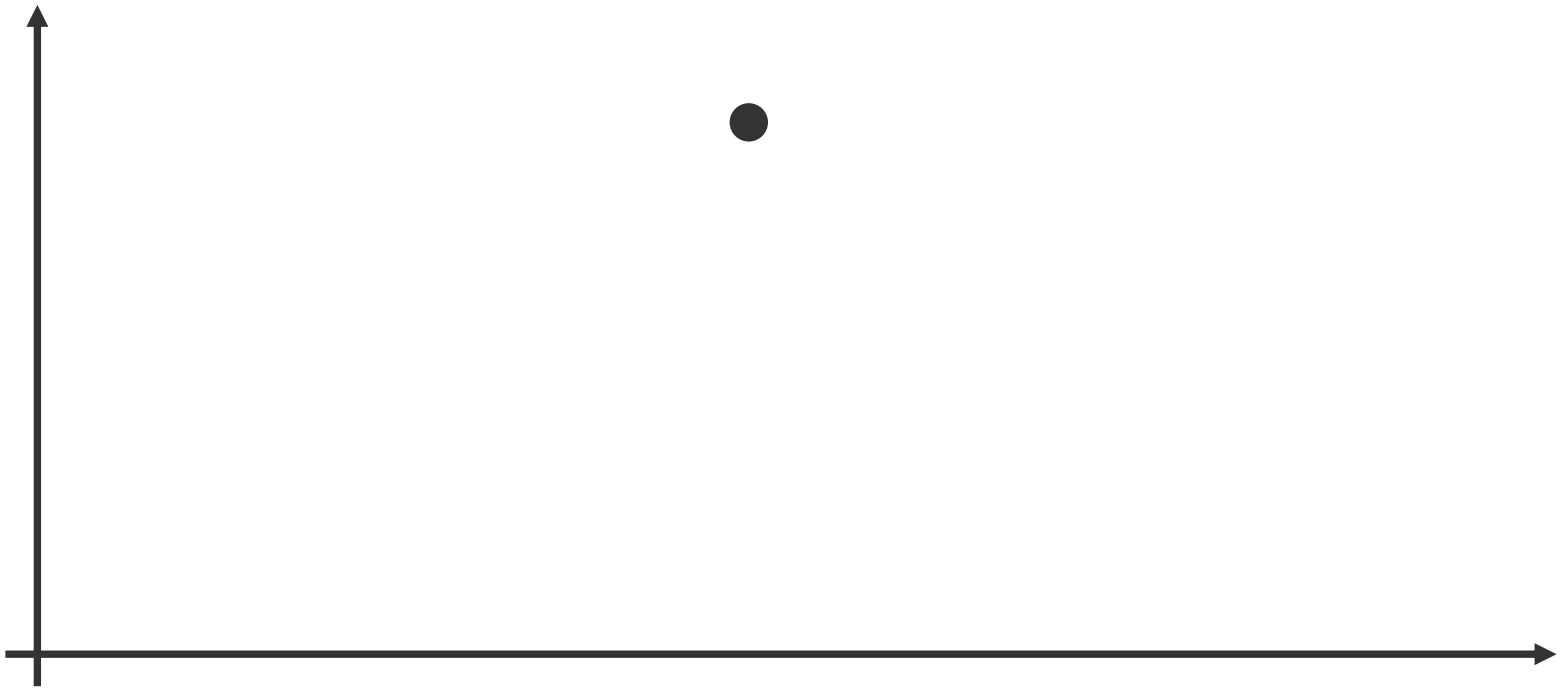
- If we have 2 data points, we “naturally” use linear model
- What should we use if we have three data points? Parabola or linear?
- What if we have one data point?

Physics vs. data science



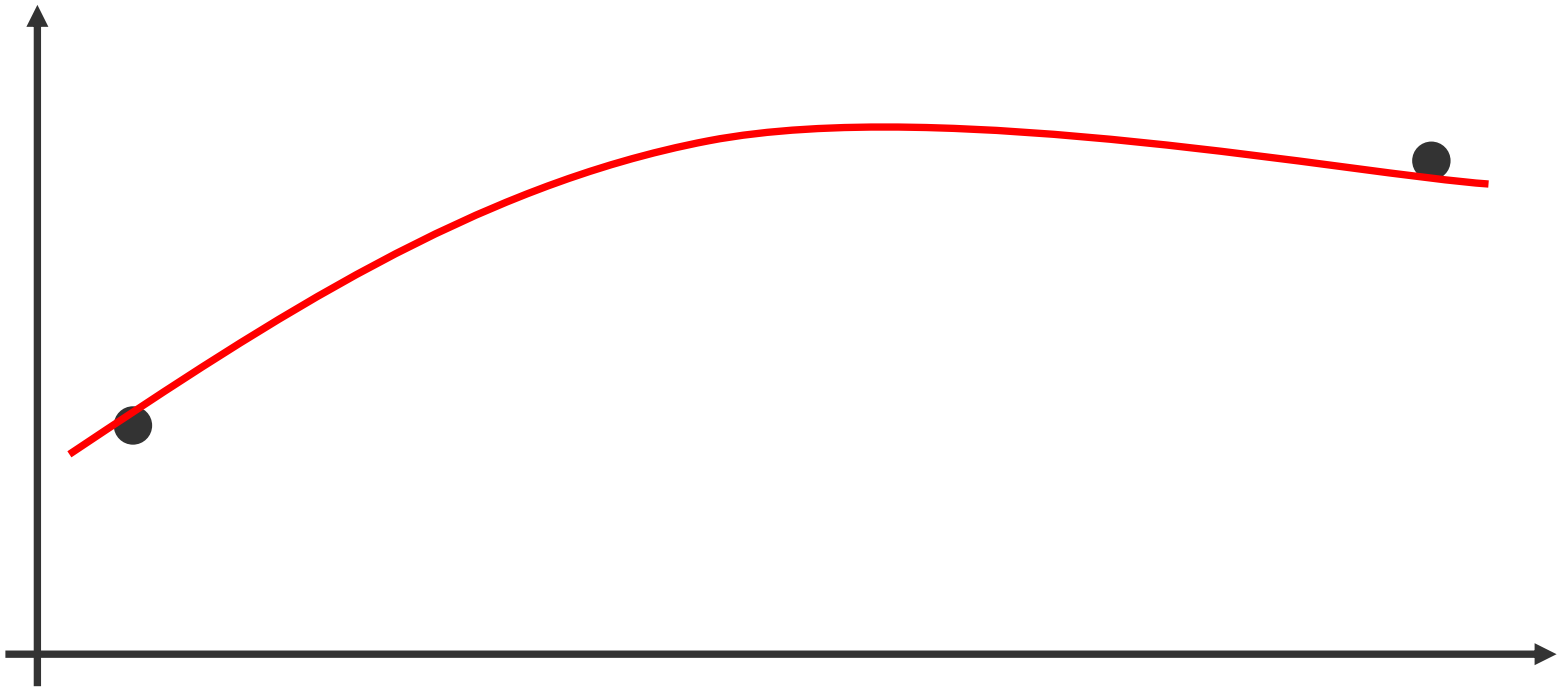
- If we have 2 data points, we “naturally” use linear model
- What should we use if we have three data points? Parabola or linear?
- What if we have one data point?

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?