Overview of Machine Learning

From Lecture 2

Machine learning involves interrelated tasks of

- *representing data,*
- *modeling data, and*
- *learning from data*

(with little to no human intervention)

We have now reviewed mathematics of

- *vectors, matrices, tensors*
- *functions and mappings*
- *optimization*

Concepts, Supervised Learning, Unsupervised Learning, Regression, Classification

The Essential Elements of (most) ML

To use most ML methods, we will need to conceptualize our problem into the following:

Features (inputs, descriptors)

$$\{\boldsymbol{x}_i\} \leftrightarrow \boldsymbol{X}$$

$\boldsymbol{x}_i$ *feature vector of sample i*

- a numerical description of (ideally) characteristics that distinguish one sample from another
- may (or may not) have direct implications on the modeling outputs

Labels (outputs)

$$\{y_i\} \leftrightarrow \boldsymbol{y}; \{\boldsymbol{y}_i\} \leftrightarrow \boldsymbol{Y}$$

y_i or $\boldsymbol{y}_i$ *scalar or vector label of sample i*

- also a numerical (integer or real) description of sample i
- usually reserved for some special quantity or property of interest

Labeled Data

$$\{(\boldsymbol{x}, y)_i\}$$

a set of tuples where features and labels are known

Unlabeled Data

$$\{\boldsymbol{x}_i\}$$

labels are not necessarily known or provided with features

Model

a function that operates on features

$$f(\boldsymbol{x}) \text{ or } \boldsymbol{f}(\boldsymbol{x})$$

- *often defines a mapping from feature space to label space*

Predictions

the function output or predicted labels

$$\hat{y} = f(\boldsymbol{x}) \text{ or } \hat{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{x})$$

Classes of Machine Learning

Machine learning is deployed in three main modes:

Supervised Learning

- In ***supervised learning***, we aim to create a model that can predict ***y*** as a function of ***x***.
- The optimization/learning of our model is ***supervised*** because the algorithm will exploit knowledge of labels over the dataset

Supervised learning can be used for either

- ***Regression*** –predict a *continuous* label. This is likely to be true for QSPR problems in physical science.
  e.g., conductivity, melting point, band gaps

- ***Classification*** – predict *categorical* labels or class membership. This can be useful for characterizing discrete outcomes
  e.g., (in)soluble, (un)sythesizable, (in)activity, hazardous

Unsupervised Learning

- In ***unsupervised learning***, we aim to create a model that identifies patterns in ***x***.
- The optimization/learning of our model is ***unsupervised*** because the algorithm will not exploit knowledge of labels over the dataset

Unsupervised learning is usually used for

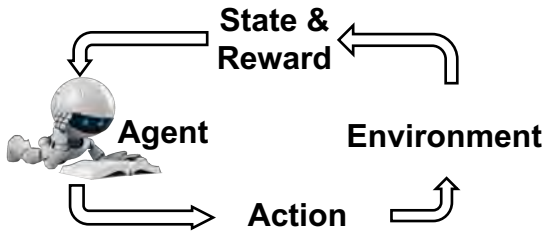
- ***Clustering*** –partition features into a set of different classes/groups, which is the *y*.
  e.g., chemical classes

- ***Signal processing*** – Uncover the underlying signal within a set of features. This is often a part of representation learning.
  e.g., protein folding pathways

- ***Generating –*** create a model distribution over **x** such that we can generate new samples

Classes of Machine Learning

Machine learning is deployed in three main modes:



Supervised Learning

- In ***supervised learning***, we aim to create a model that can predict ***y*** as a function of ***x***.
- The optimization/learning of our model is ***supervised*** because the algorithm will exploit knowledge of labels over the dataset

Supervised learning can be used for either

- ***Regression*** –predict a *continuous* label. This is likely to be true for QSPR problems in physical science.
  e.g., conductivity, melting point, band gaps
- ***Classification*** – predict *categorical* labels or class membership. This can be useful for characterizing discrete outcomes
  e.g., (in)soluble, (un)sythesizable, (in)activity, hazardous

Unsupervised Learning

- In ***unsupervised learning***, we aim to create a model that identifies patterns in ***x***.
- The optimization/learning of our model is ***unsupervised*** because the algorithm will not exploit knowledge of labels over the dataset

Unsupervised learning is usually used for

- ***Clustering*** –partition features into a set of different classes/groups, which is the *y*.
  e.g., chemical classes
- ***Signal processing*** – Uncover the underlying signal within a set of features. This is often a part of representation learning.
  e.g., protein folding pathways
- ***Generating*** – create a model distribution over **x** such that we can generate new samples

Reinforcement Learning

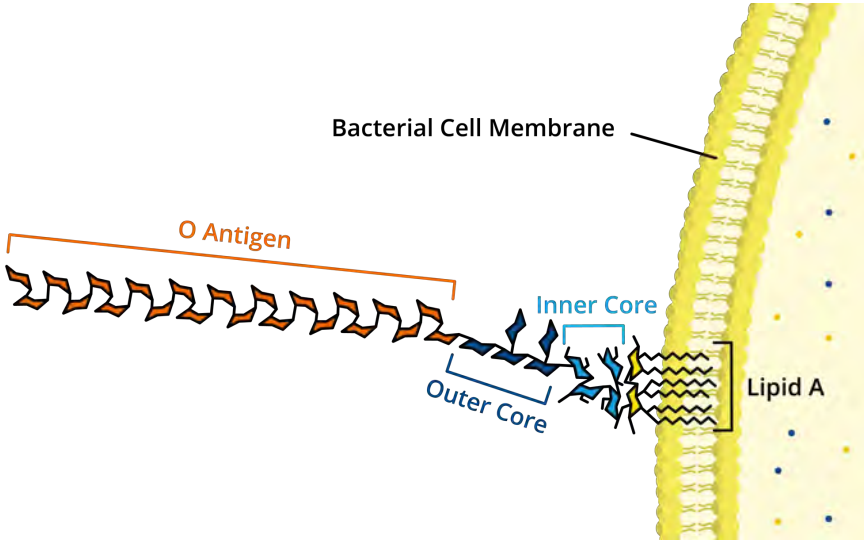
- In reinforcement learning, an "agent" learns how to interact with its environment based on feedback via cumulative rewards/penalties
- Many things that people think are reinforcement learning are probably not reinforcement learning
- Usually about planning and scheduling

e.g., automated process synthesis, process control

- In ***semi-supervised learning***, we want a model that can predict ***y*** as a function of ***x,*** just as in supervised learning
- ***Both labeled and unlabeled data*** are used in modes like *co-training, pseudo-labeling, and label propagation*

- In ***self-supervised learning***, we eventually want a model that can predict ***y*** as a function of ***x,***
- ***Only unlabeled data*** are used during training; one form is *contrastive learning*

Example on Bioanalytical Chemistry



thenativeantigencompany.com

- Pharmaceutical Manufacturing
- Medical Device Testing
- Water Testing
- Food and Beverage Testing

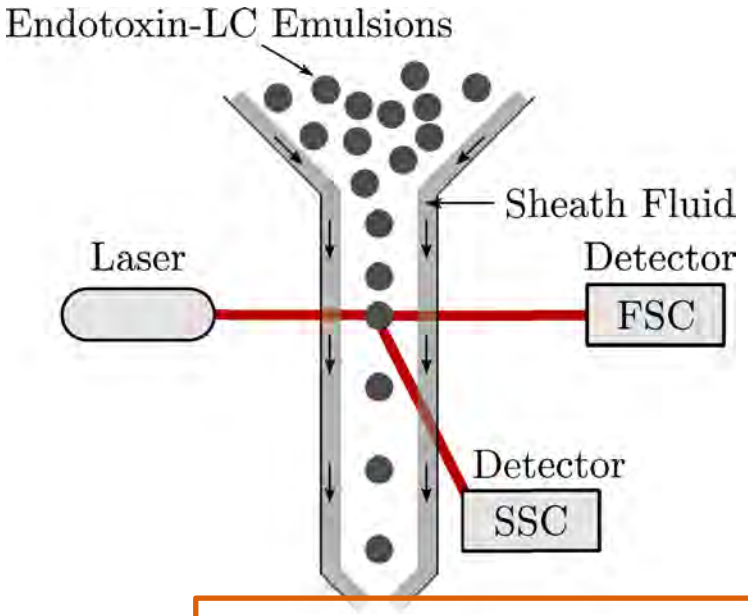
nhm.ac.uk

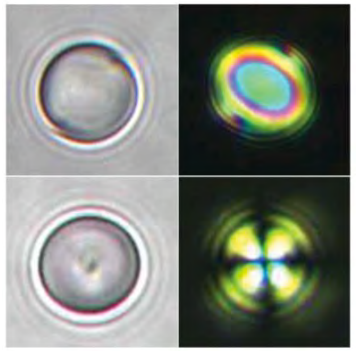
Endotoxins, found in the outer membrane of gram-negative bacteria like *E. coli*, release upon cell death. When entering the bloodstream, they cause *fever, septic shock, and immune responses* due to the immune system's recognition of the unique lipid A component.

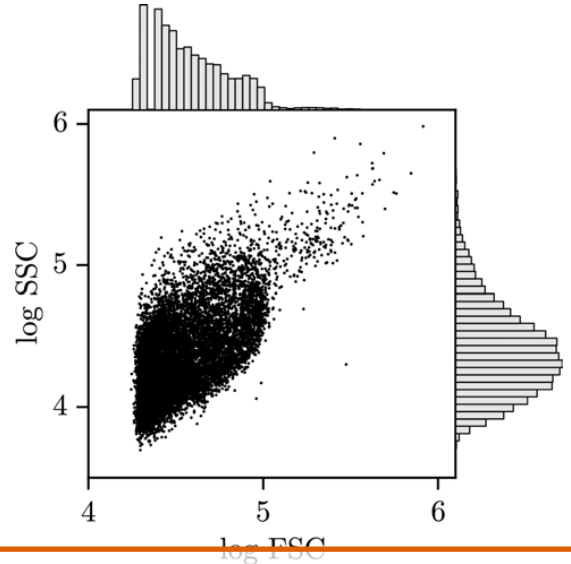
Motivation: The Limulus amoebocyte lysate (LAL) test, the most common method for endotoxin detection, uses ***horseshoe crab blood***, valued at *$60,000 per gallon*, to form a clot and can detect endotoxin levels as low as 3.125 pg/mL.

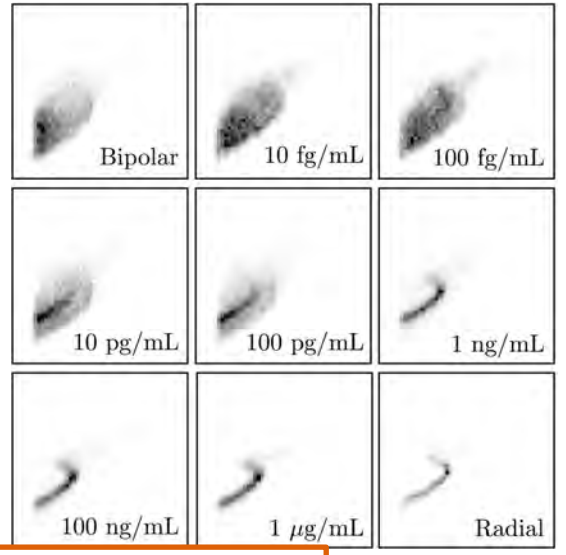
Current methods are limited to detecting the presence of endotoxins *without predicting concentrations or classifying species.*

Example on Bioanalytical Chemistry









How to represent flow cytometry scatter plots?

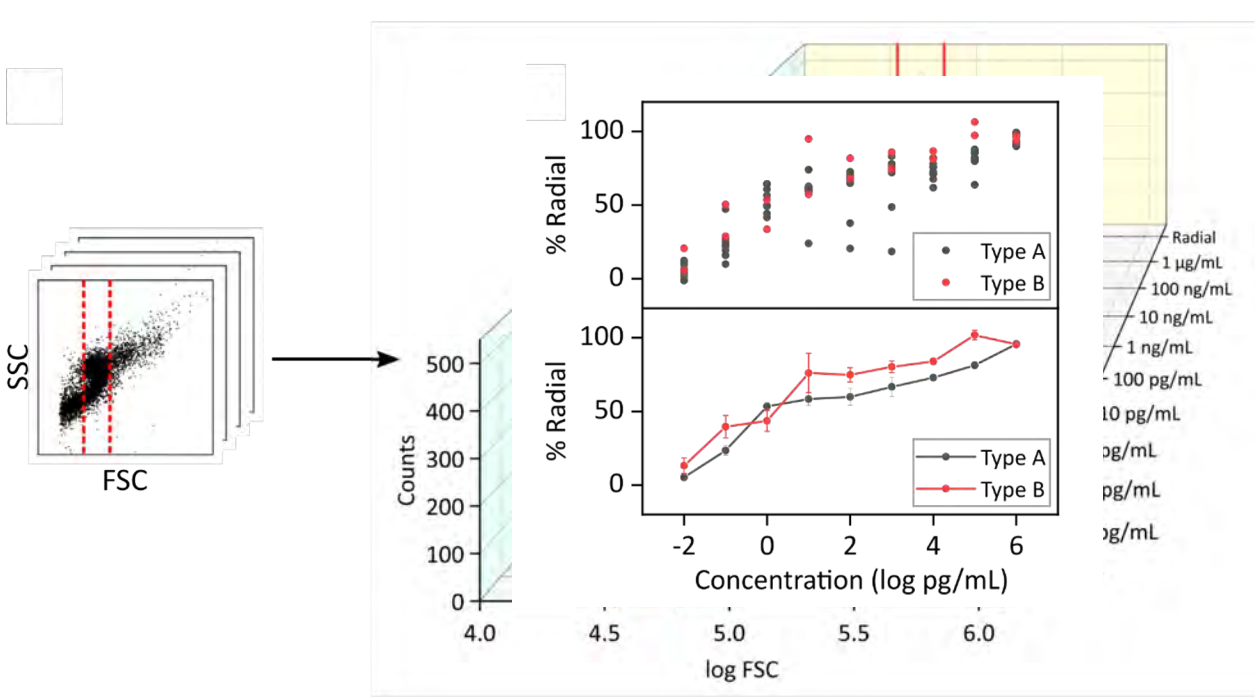
Jiang et al. *Analyst* 146 (**2021**)

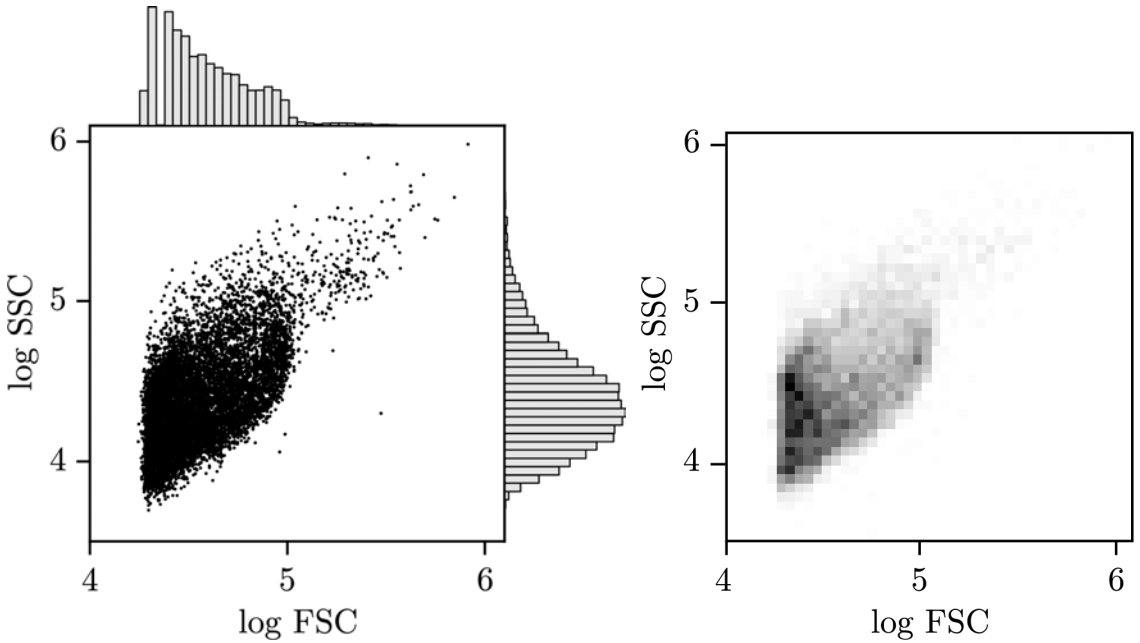
Endotoxins trigger changes in the *ordering of liquid crystal (LC) droplets*, which are injected into a fluid stream. After flow focusing into a laser beam, the intensity of light scattered by each LC droplet is measured at both small ("forward scattering," FSC) and large scattering angles ("side scattering," SSC).

The flow cytometry approach is both inexpensive and highly sensitive, detecting levels as low as *0.01 pg/mL*.

Scatter plots vary with endotoxin concentration, showing two extremes: *bipolar and radial*.

Example on Bioanalytical Chemistry



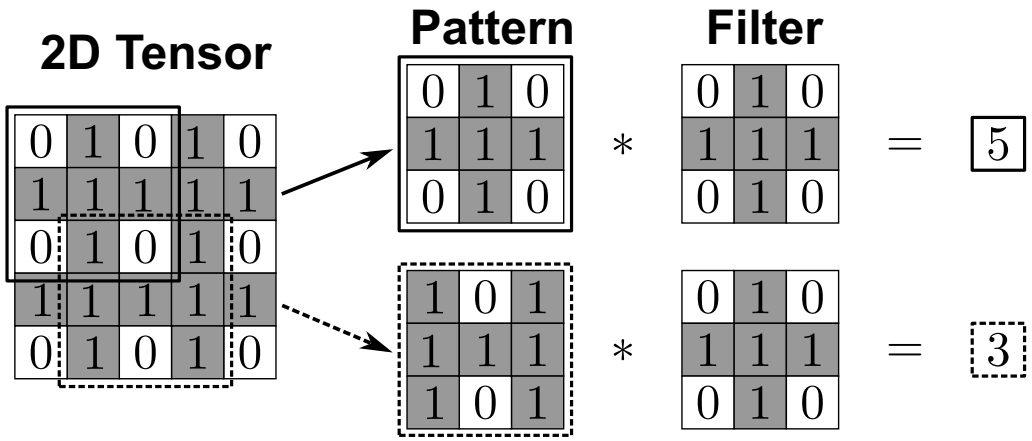


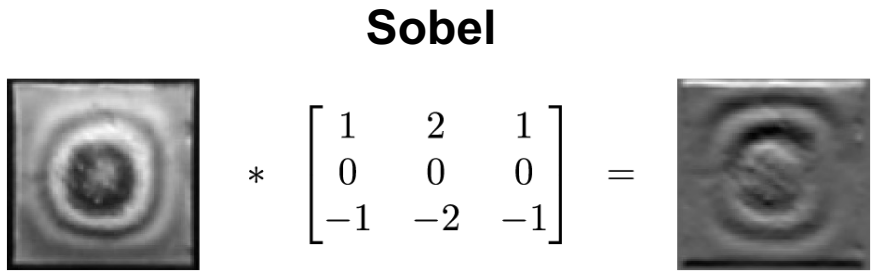
Input 1: Radial Configuration (RC) Method.
Counts *the <u>number of points between the red lines</u>* in the characteristic "S" region to create a single scalar ($x \in \mathbb{R}$).
The RC method has a high degree of uncertainty due to limited information.

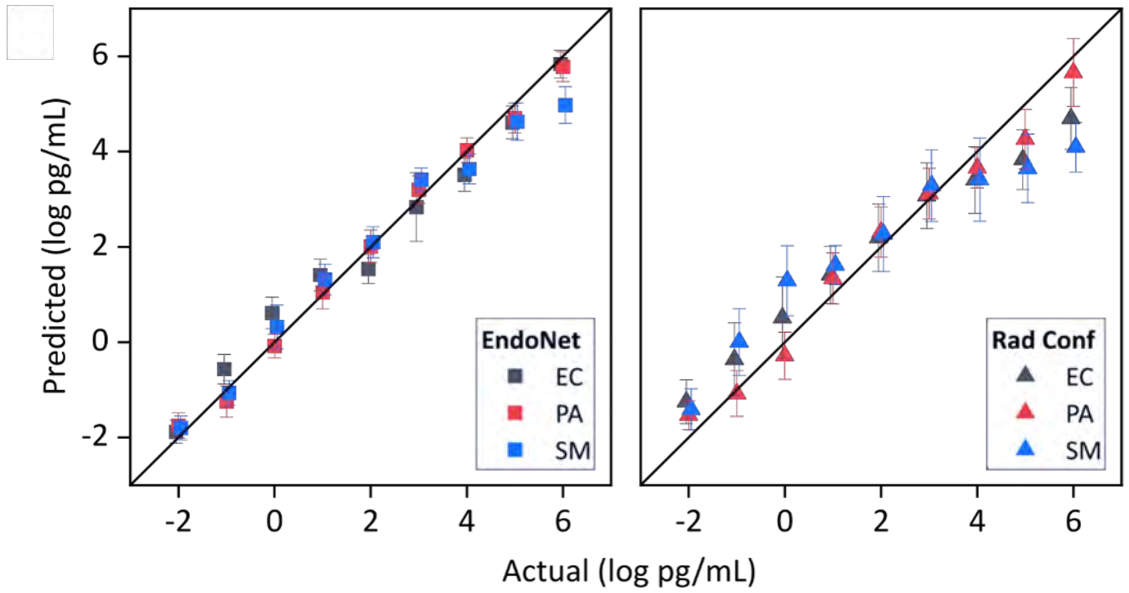
Input 2: Histogram Method.
Counts the *<u>number of points in each 2D bin</u>* to create a histogram matrix (grayscale image)($\boldsymbol{X} \in \mathbb{R}^{m \times n}$), capturing complex patterns.

Output: Endotoxin concentration ($y \in \mathbb{R}$).
Task: Supervised Regression.

Example on Bioanalytical Chemistry



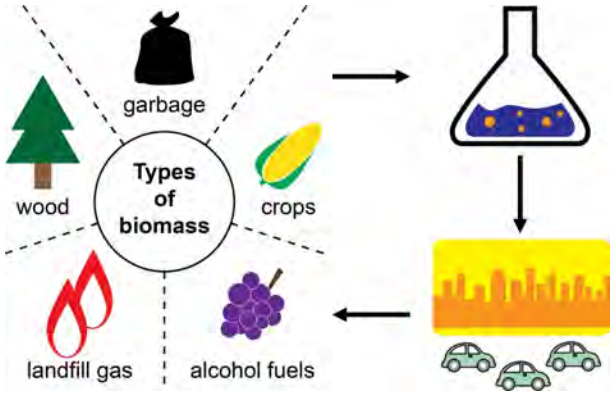


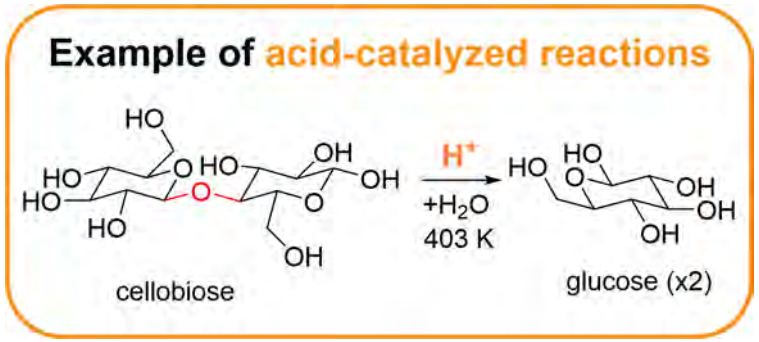


Model: Convolutional Neural Network (CNN). Convolutional filters *identify and quantify* patterns in tensors.

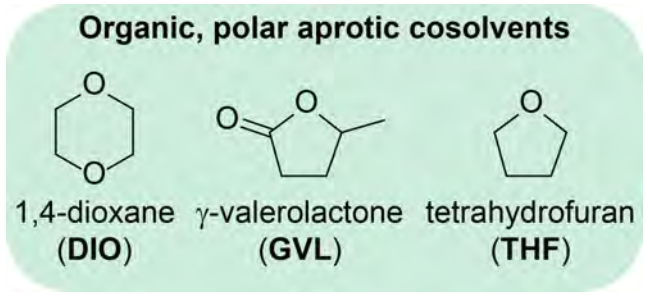
By learning complex patterns in flow cytometry scatter plots, the CNN achieves a high prediction accuracy with an R^2=0.91, compared to the RC method (R^2=0.68).

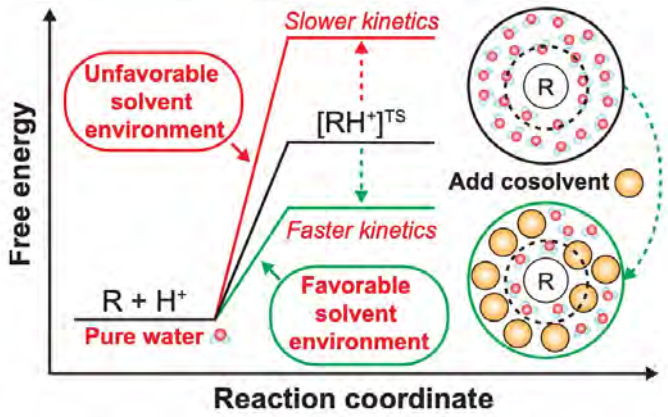
Example on Catalytical Reaction Chemistry

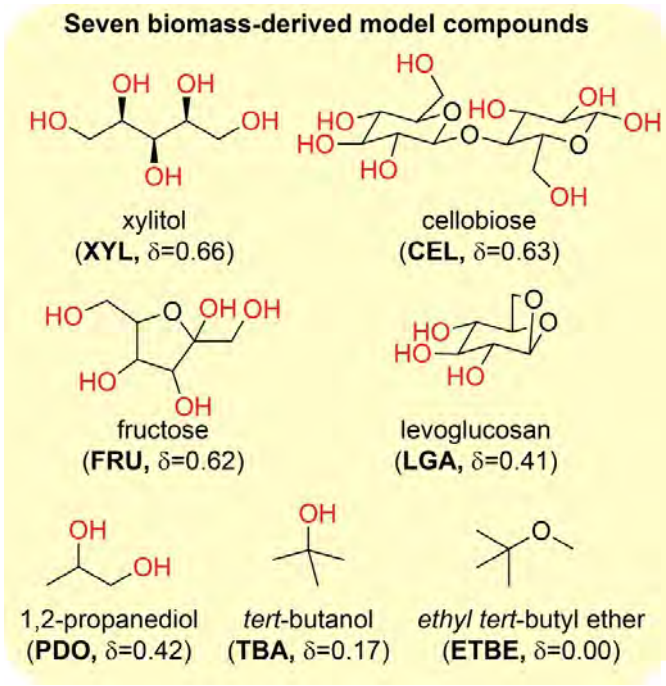




Chew et al. *Chem. Sci.* 11 (**2020**)

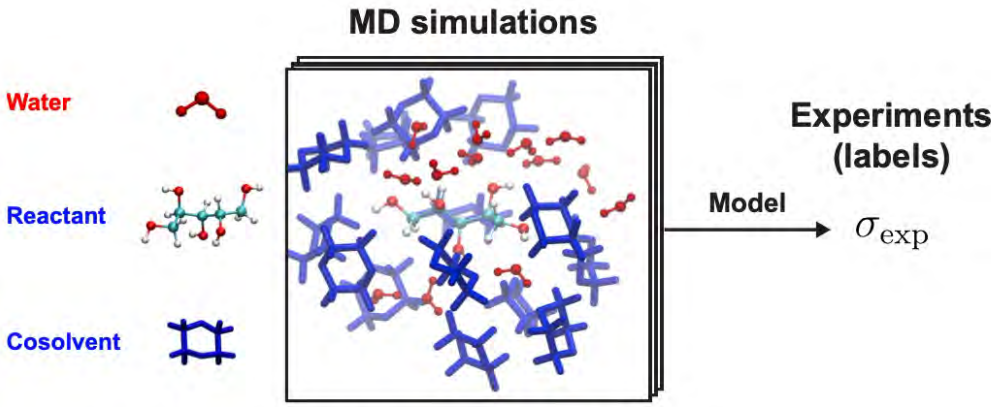


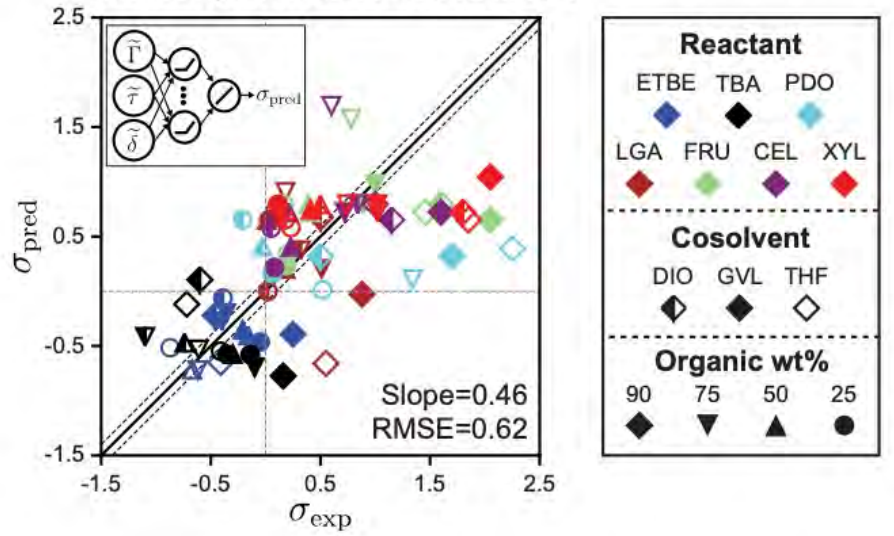




Motivation: Lignocellulosic biomass can be converted into transportation fuels and high-value chemicals through liquid-phase, acid-catalyzed reactions. However, *reactivity is low in aqueous solutions*.
Reaction rates can be increased up to 100-fold by mixing *organic, polar aprotic cosolvents* with water to create a mixed-solvent environment.

Example on Catalytical Reaction Chemistry





Molecular dynamics (MD) simulation trajectories are used to predict experimental kinetic solvent parameters (σ_{exp}).

Output: $\sigma_{\text{exp}} \in \mathbb{R}$

Task: Supervised Regression.

How to represent MD trajectories?

Input 1: Human-Selected Descriptors.

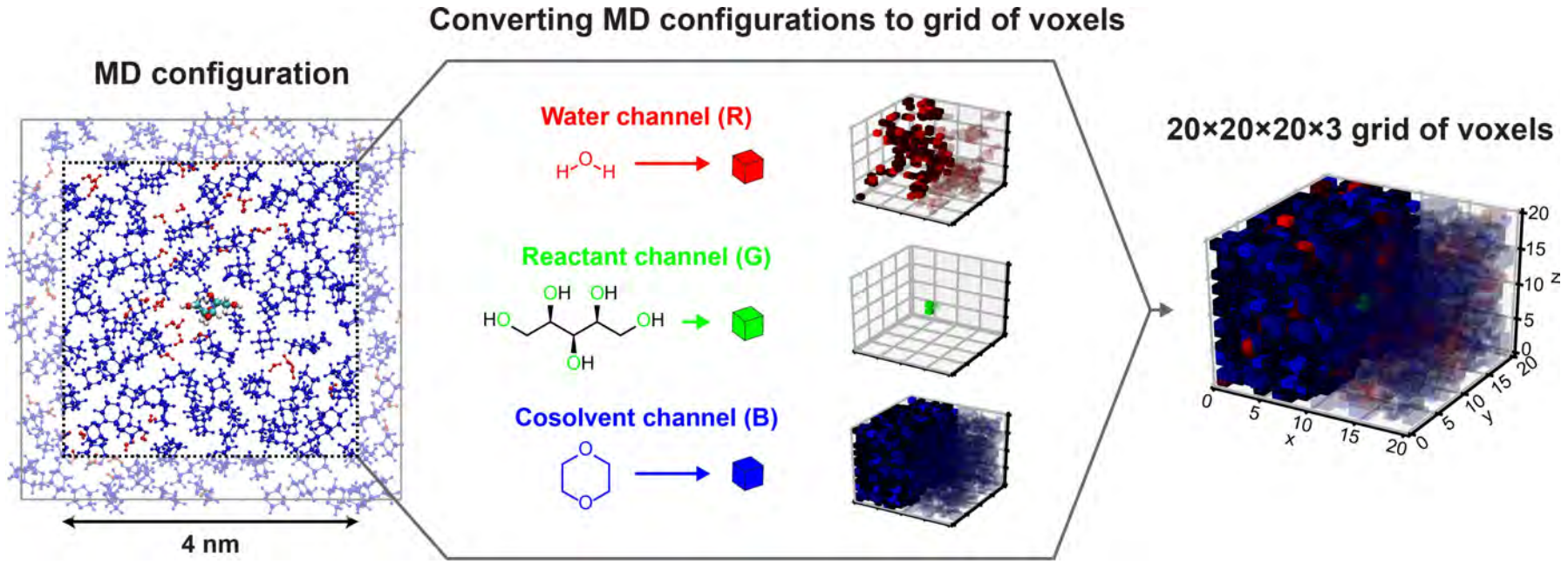
The dataset includes 76 reactant-solvent combinations, with each simulation trajectory lasting 205 ns.

For each trajectory, $\Gamma \in \mathbb{R}$ (reactant-solvent affinity), $\tau \in \mathbb{R}$ (reactant-water binding), and $\delta \in \mathbb{R}$ (reactant hydrophilicity) are calculated.

Model 1: Fully Connected Neural Network.

Significant errors for larger σ_{exp}.

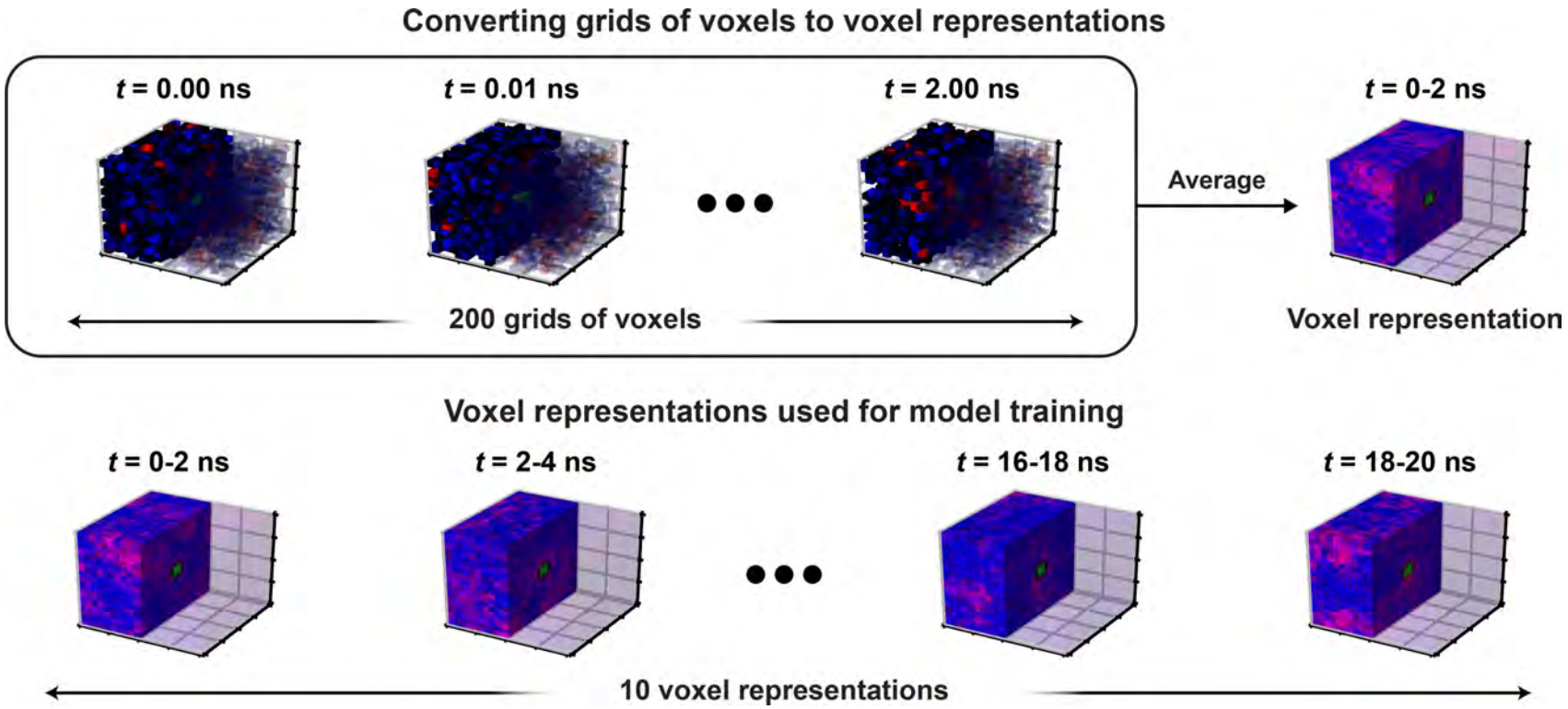
Example on Catalytical Reaction Chemistry



Input 2: Atomic Positions as 3D Tensors.
Atomic positions from an MD simulation are converted into a *3D tensor (voxel) representation*. A cube centered on the reactant is divided into a 20 × 20 × 20 grid to discretize space. The occurrences of water, reactant oxygens, and cosolvent molecules are normalized ($\boldsymbol{X} \in \mathbb{R}^{20\times20\times20\times3}$).

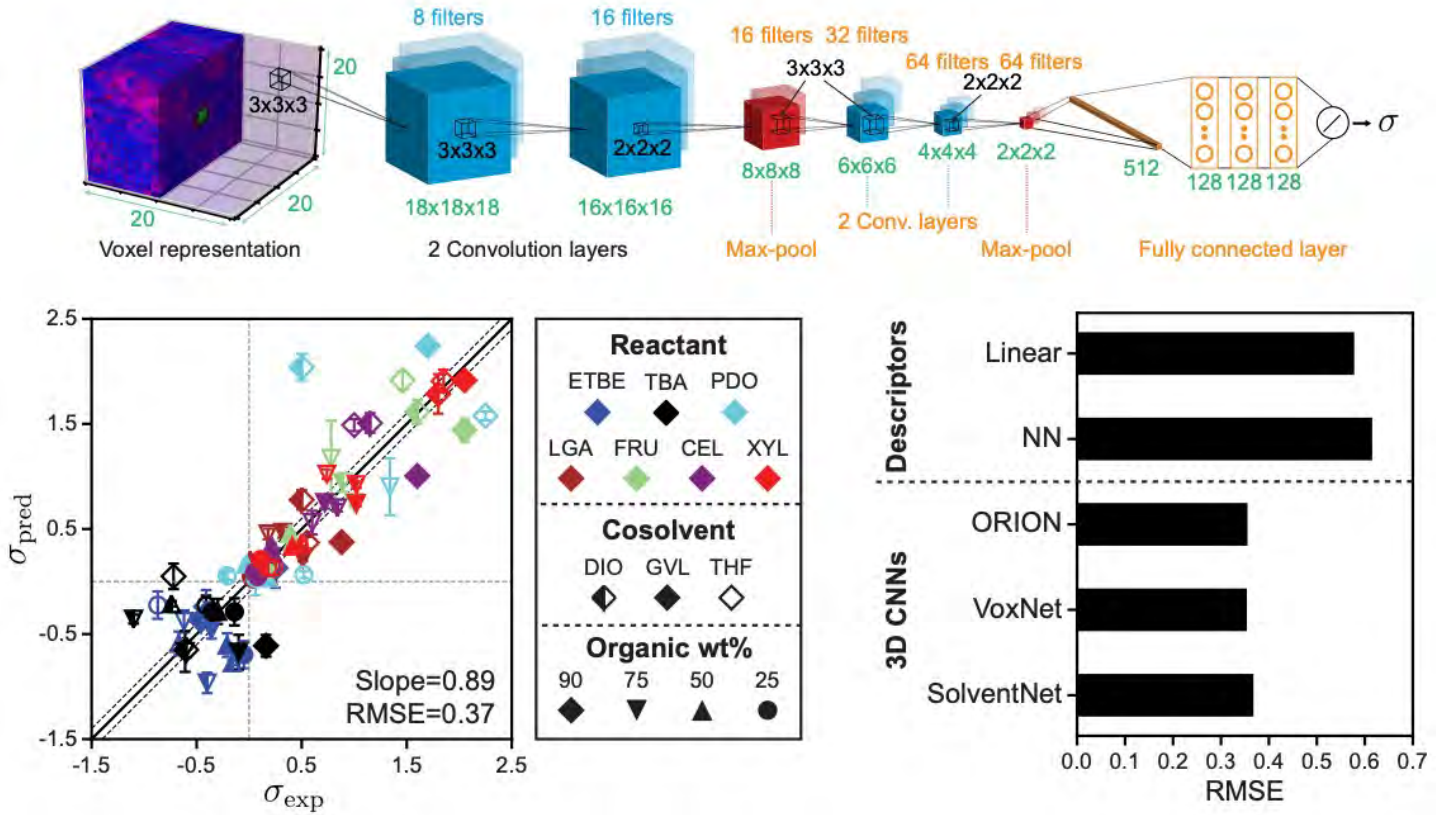
How to represent temporal information?

Example on Catalytical Reaction Chemistry



Input 2: Atomic Positions as 3D Tensors.
To capture the preferential locations of solvent molecules relative to the reactant as they diffuse within the cubic volume and to avoid unoccupied voxels, voxel grids were *averaged over 2 ns of MD data* (200 configurations). A total of 20 ns of simulation data generates 10 independent voxel representations.

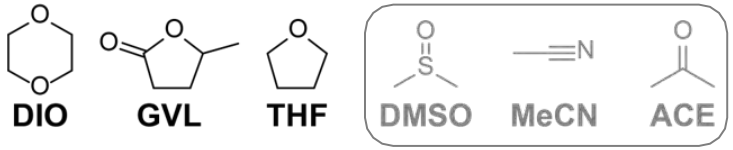
Example on Catalytical Reaction Chemistry

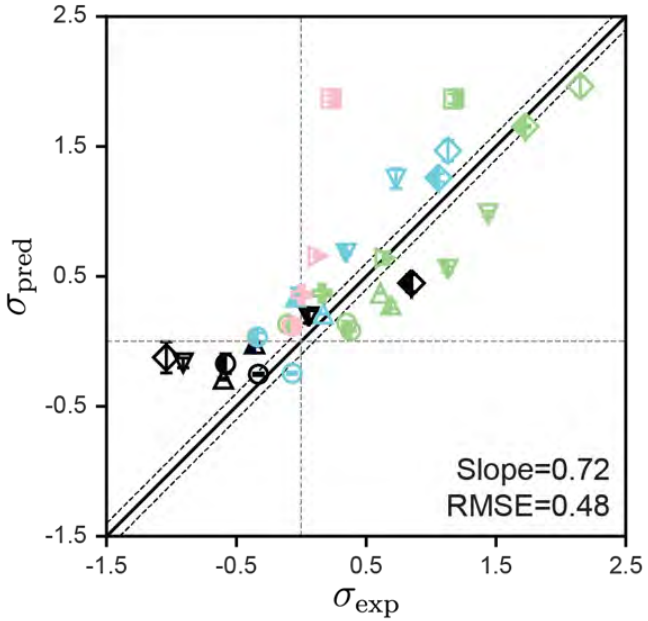


Model 2: 3D Convolutional Neural Network.

3D convolutional filters *identify and quantify* patterns in 3D tensors. All models employ 5-fold cross-validation with 24 rotational data augmentations. Across different architectures, the 3D CNNs consistently achieve higher accuracy.

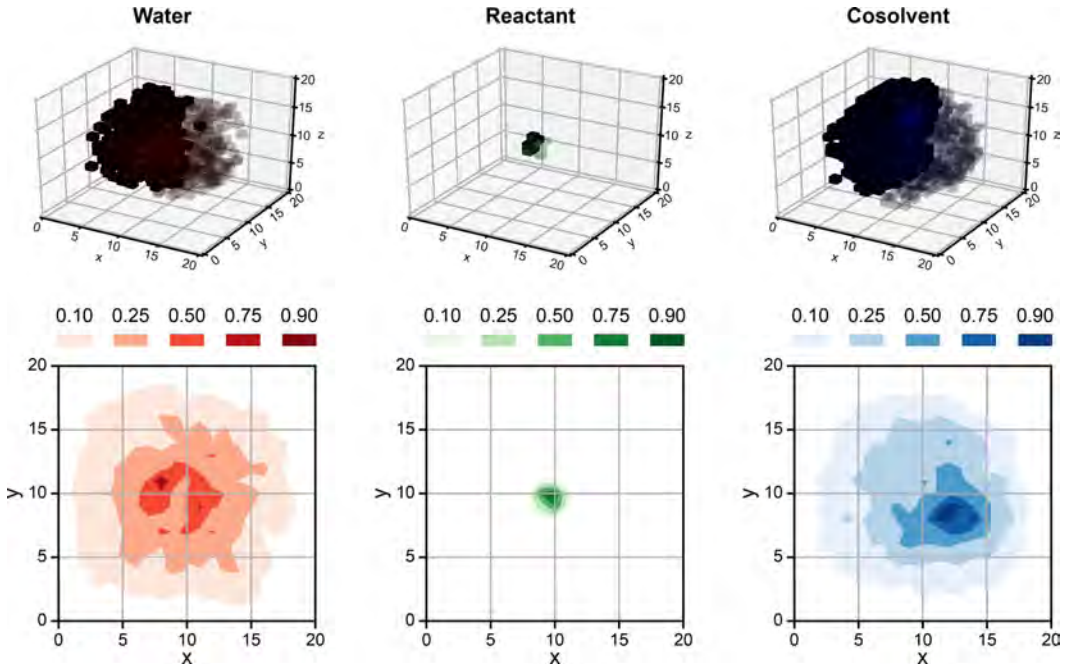
Example on Catalytical Reaction Chemistry





Model Transferability

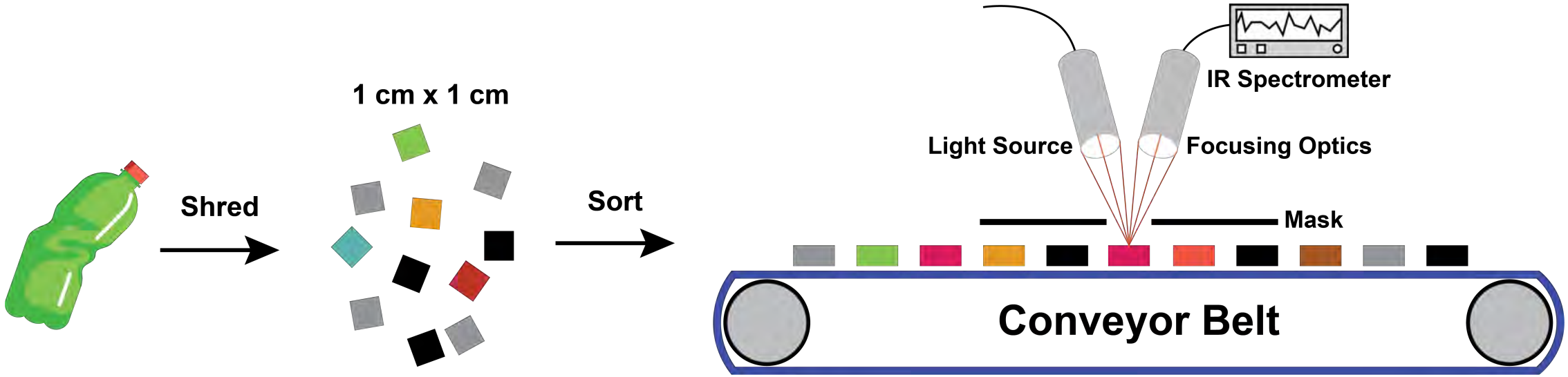
The model was tested on new solvents and reactants from literature sources, representing *out-of-distribution (OOD) data*. The high OOD test set accuracy demonstrates that the model is *transferable* across various reactant-solvent combinations.



Saliency Analysis

The key feature is the *local solvent environment near the reactant,* which agrees with the human-selected descriptors.

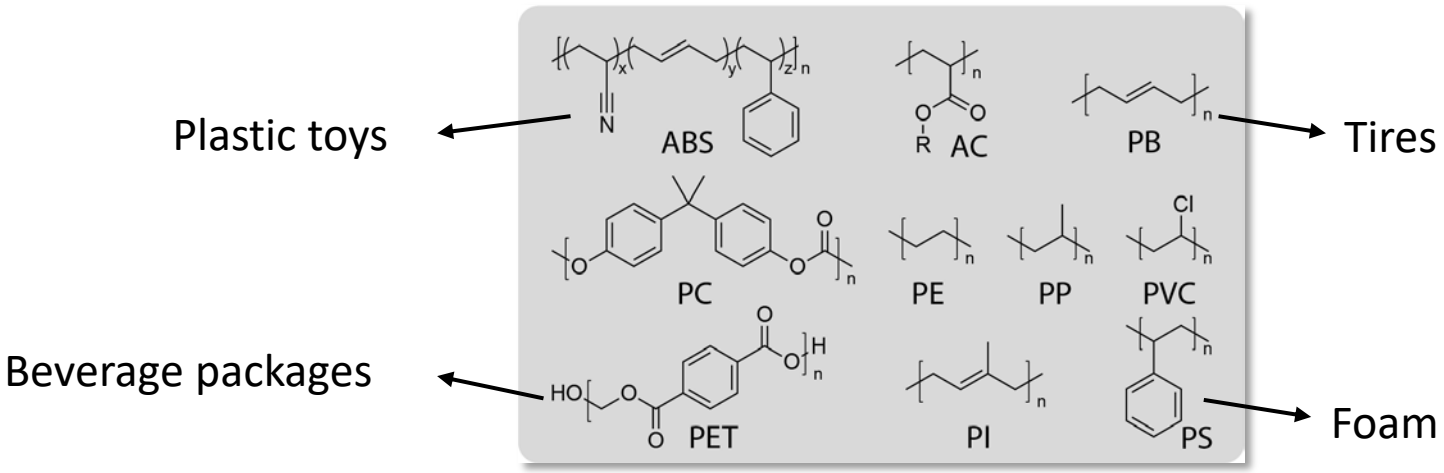
Example on Spectral Analysis



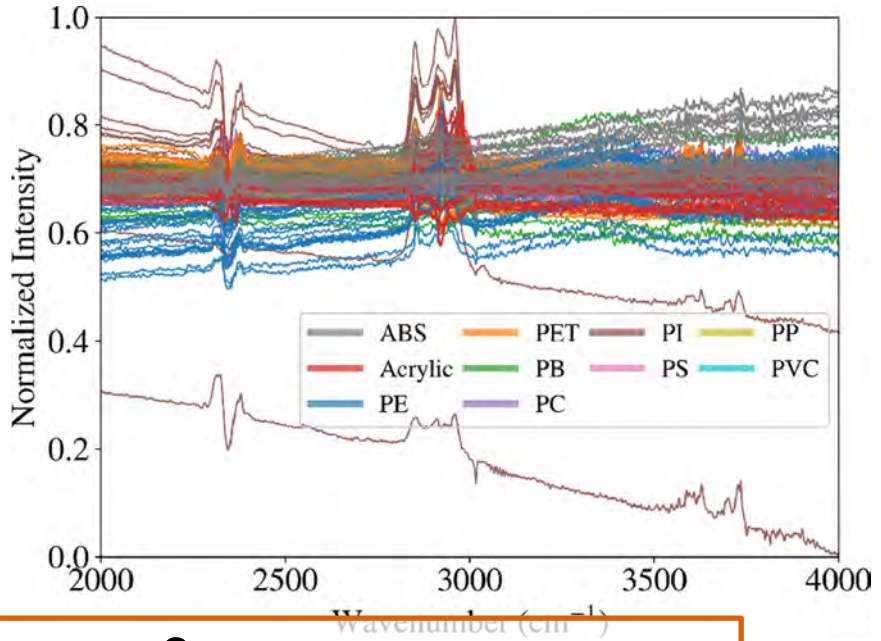
Long et al. *ACS Sustainable Chem. Eng.* 10 (**2022**)

Motivation: Preventing plastic waste pollution is an urgent challenge. Since 1950, only *6% of plastic has been recycled, and just 1.2% remains in use.*
When mixed, recycled plastic often results in low-value products like plastic lumber. To enhance the value of recycled plastic, effective sorting of different plastic types is essential. *ATR-FTIR* offers a high-throughput method for sorting plastic waste.

Example on Spectral Analysis



Jiang et al. *Comp. Chem. Eng.* 155 (**2021**)



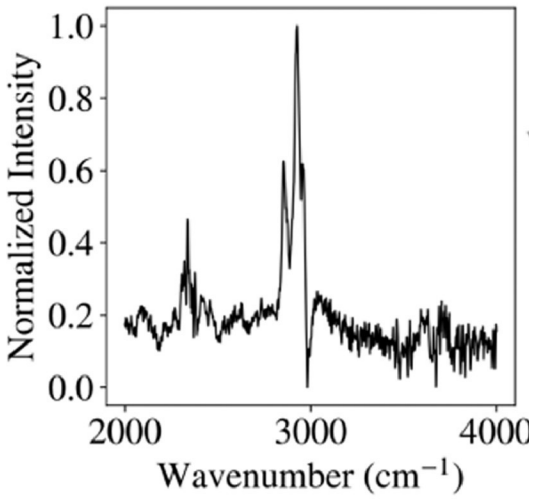
How to represent ATR-FTIR spectra?

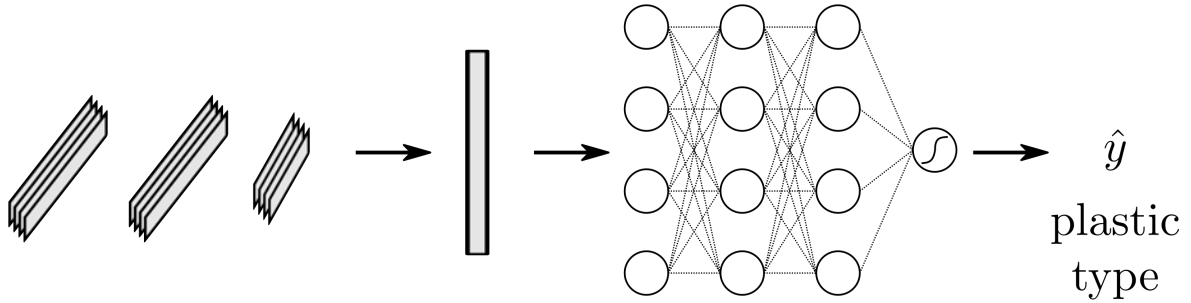
This study involves 10 plastic types, with 70 spectral measurements for each. The spectra exhibit significant overlap between different plastic types, along with noise and systematic errors.

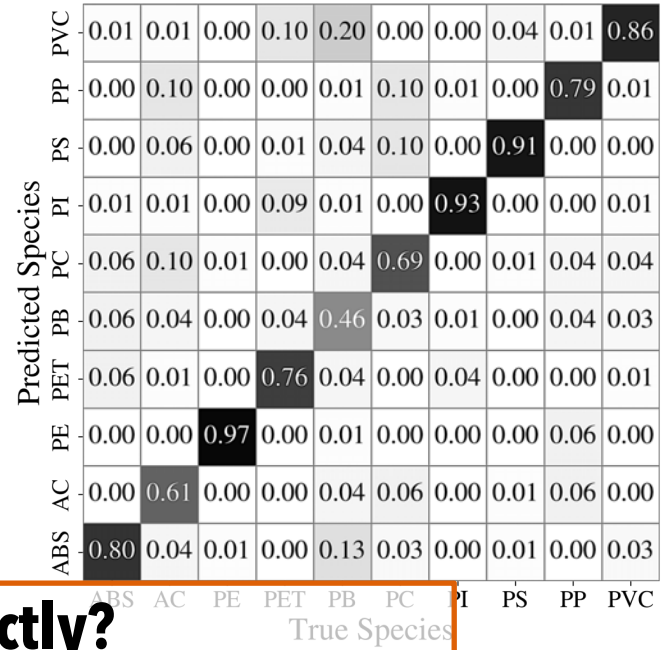
Task: Supervised Classification.

Output: Probability vector indicating the likelihood of each plastic type ($\boldsymbol{y} \in \mathbb{R}^{10}$).

Example on Spectral Analysis

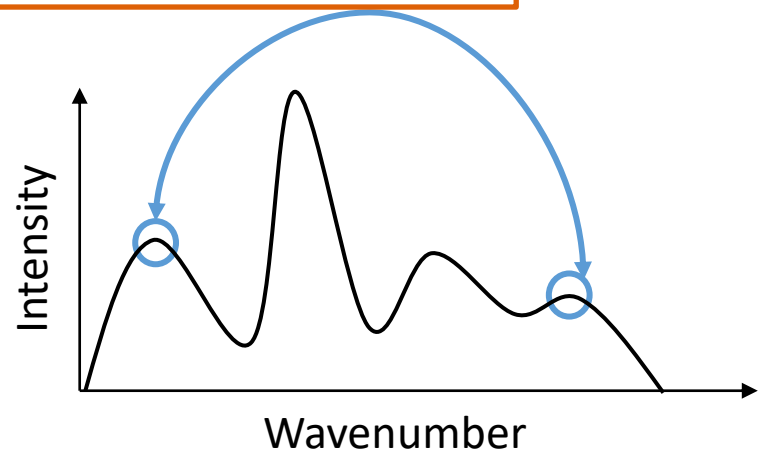




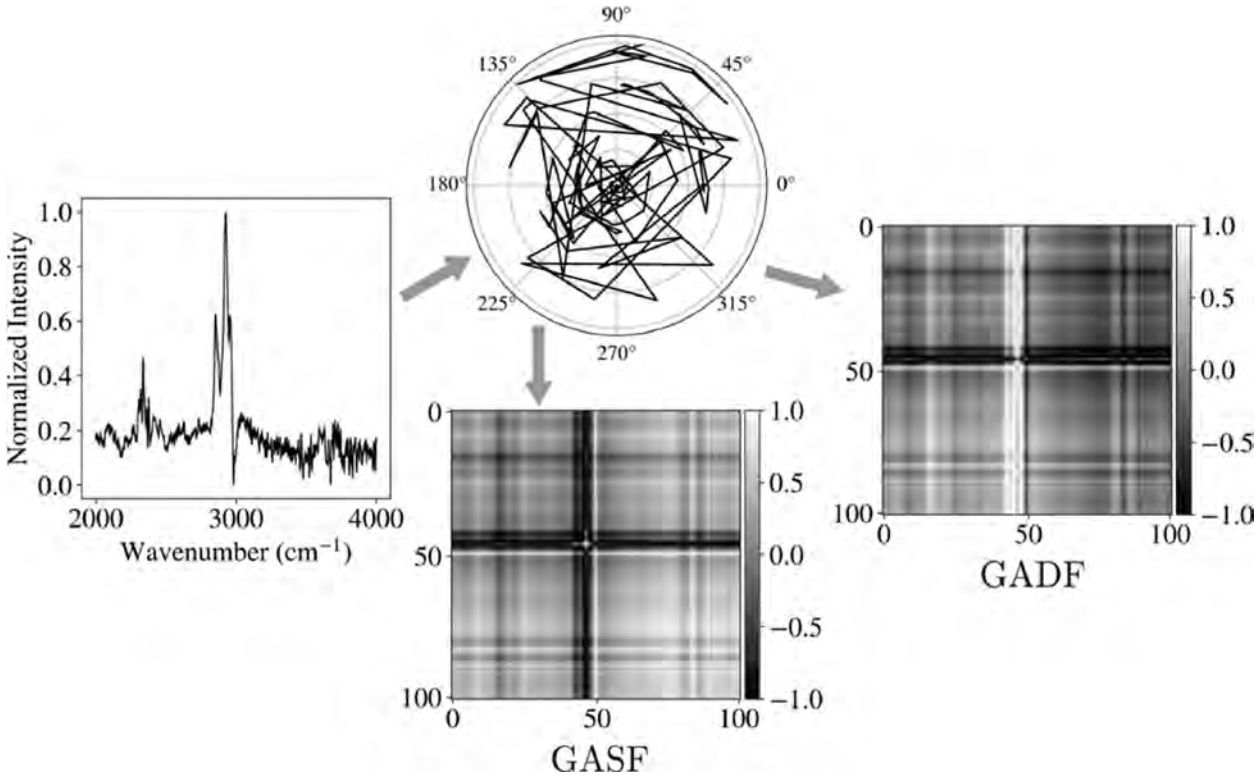


What if we encode long-range correlations directly?

Input 1: Vector (1D Tensor).
Model 1: 1D CNN.
Each entry represents the intensity value at a specific wavenumber ($\boldsymbol{x} \in \mathbb{R}^{4150}$).
The accuracy, reflected in the *confusion matrix*, is relatively low (0.78).
Machine learning model struggles to *capture long-range correlations across wavenumbers*.



Example on Spectral Analysis



Input 2: Matrix (2D Tensor).
Model 2: 2D CNN.
The matrix representation (Gramian Angular Field, $\boldsymbol{X} \in \mathbb{R}^{100\times100}$) captures correlations across frequencies, achieving a higher accuracy of 0.87.
A simple change in data representation greatly improves accuracy.

IR Spectrum x

↓

Polar Representation

$$\phi_i = \arccos(x_i)$$

$$r_i = \frac{i}{4150}, i = 1, \cdots, 4150$$

↓

$$GASF = \cos(\phi_i + \phi_j)$$

$$GADF = \sin(\phi_i - \phi_j)$$

↓

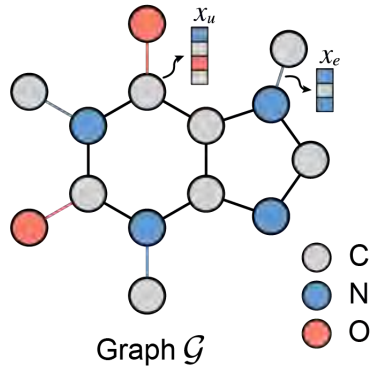
Dimension Reduction

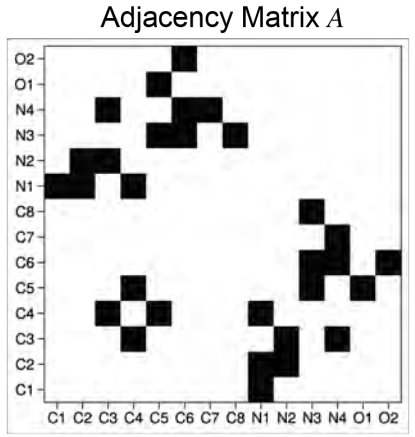
$$GASF \in \mathbb{R}^{100\times100}$$

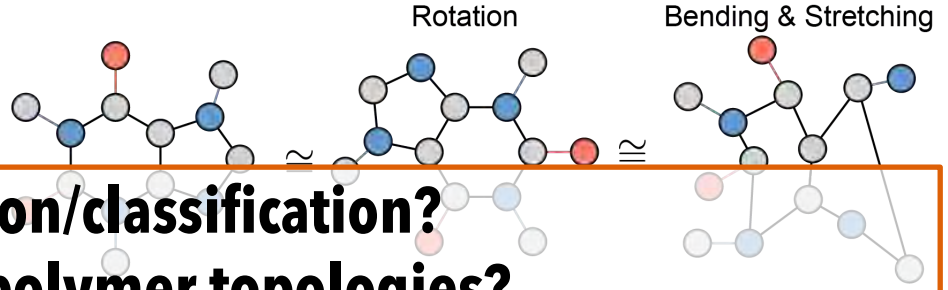
$$GADF \in \mathbb{R}^{100\times100}$$

Example on Polymer Science

Jiang et al. *npj. Comp. Mat.* 10 (**2024**)







Motivation:

Designing polymer topologies with desired characteristic size is challenging, as the design space cannot be fully enumerated. Instead, we use *generative modeling* to avoid enumeration.

Multi-Task Learning (MTL):

- **Topology reconstruction (connectivity)**
- **Characteristic size prediction (scalar)**
- **Topology prediction (probability vector)**

Which task is regression/classification?

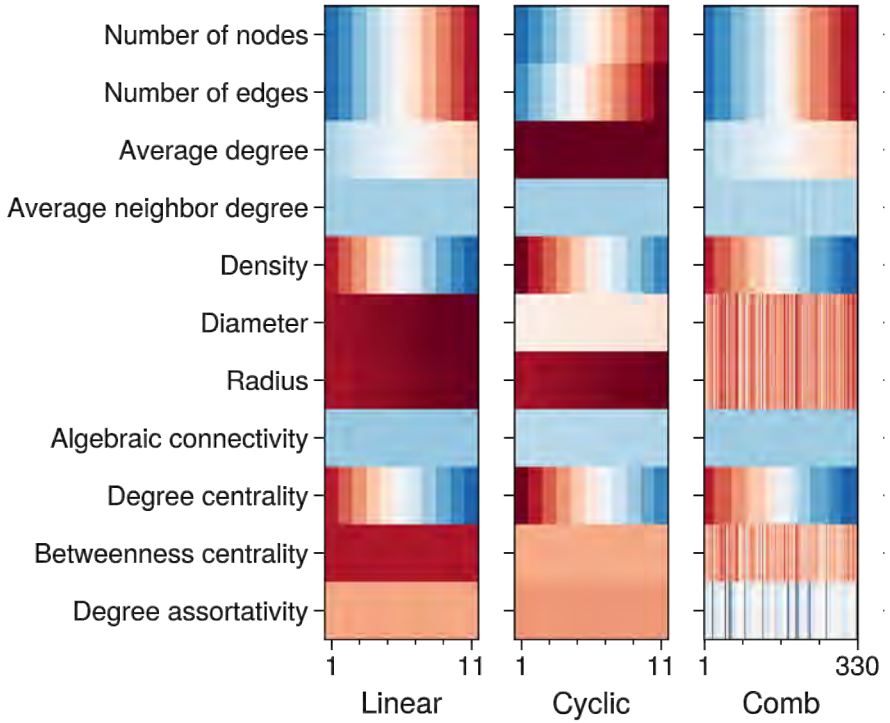
What about generating polymer topologies?

Input 1: Graph Representation.

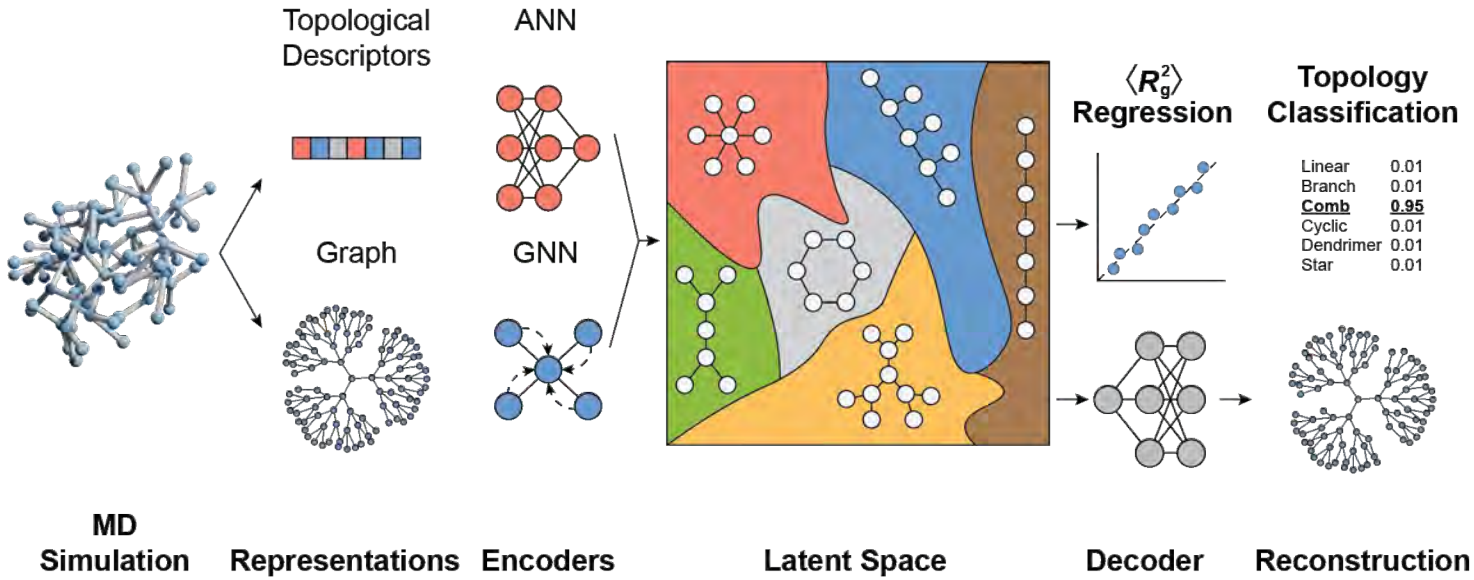
Polymers and small molecules can be represented as graphs, with *constituent units or atoms as nodes* and *bonds as edges*. The adjacency matrix describes the *connectivity* between nodes. $\mathcal{G} = (\boldsymbol{X} \in \mathbb{R}^{100\times d}, \boldsymbol{A} \in \mathbb{R}^{100\times 100})$.

Graphs are invariant to deformations.

Example on Polymer Science



Jiang et al. *npj. Comp. Mat.* 10 (**2024**)



Input 2: Topological Descriptor Vector ($\boldsymbol{x} \in \mathbb{R}^{11}$).

For example, *algebraic connectivity*, the second smallest eigenvalue of the graph Laplacian matrix, describes the *graph's connectivity*.

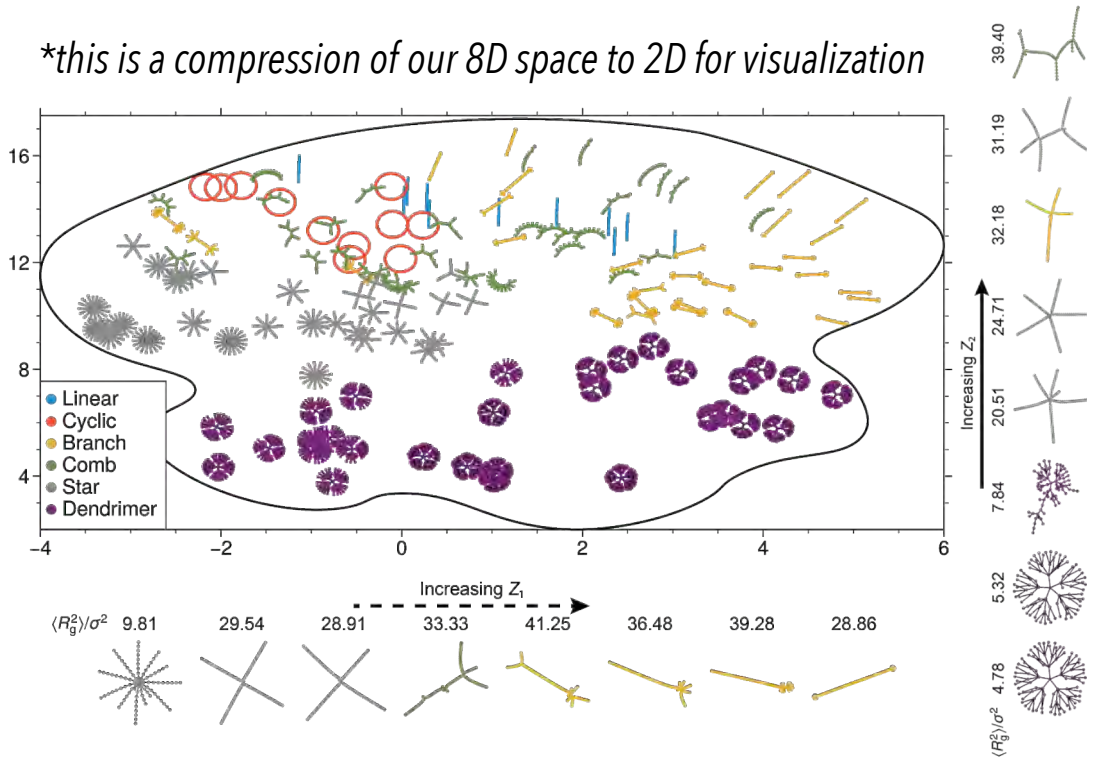
Model: Multi-input, Multi-output Variational Autoencoder.

The *encoder* condenses information into an 8-dimensional probabilistic feature space ("latent space"). The *decoder* reconstructs the polymer graph. The *regressor* predicts characteristic size, and the *classifier* classifies topology.

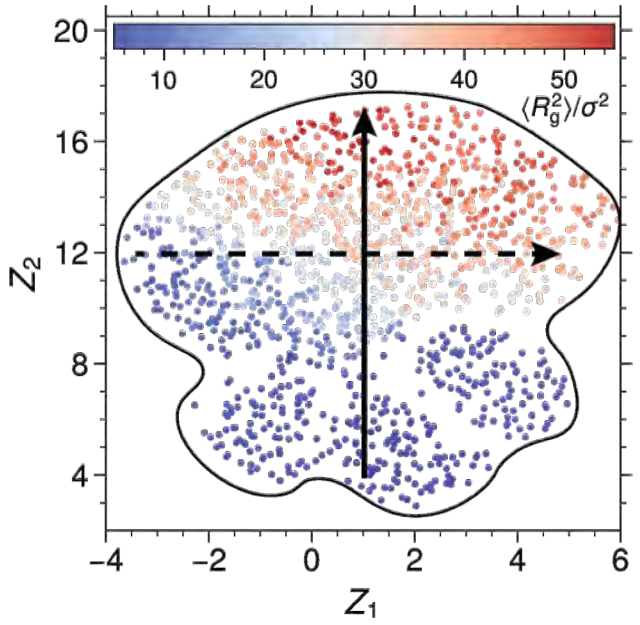
Output: $(\boldsymbol{A} \in \mathbb{R}^{100\times100}, y_r \in \mathbb{R}, \boldsymbol{y_c} \in \mathbb{R}^6)$.

Example on Polymer Science

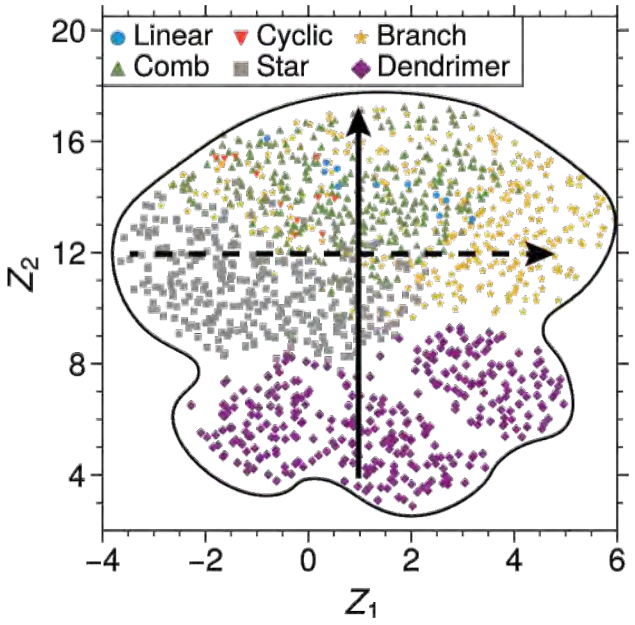
**this is a compression of our 8D space to 2D for visualization*



Radius of gyration

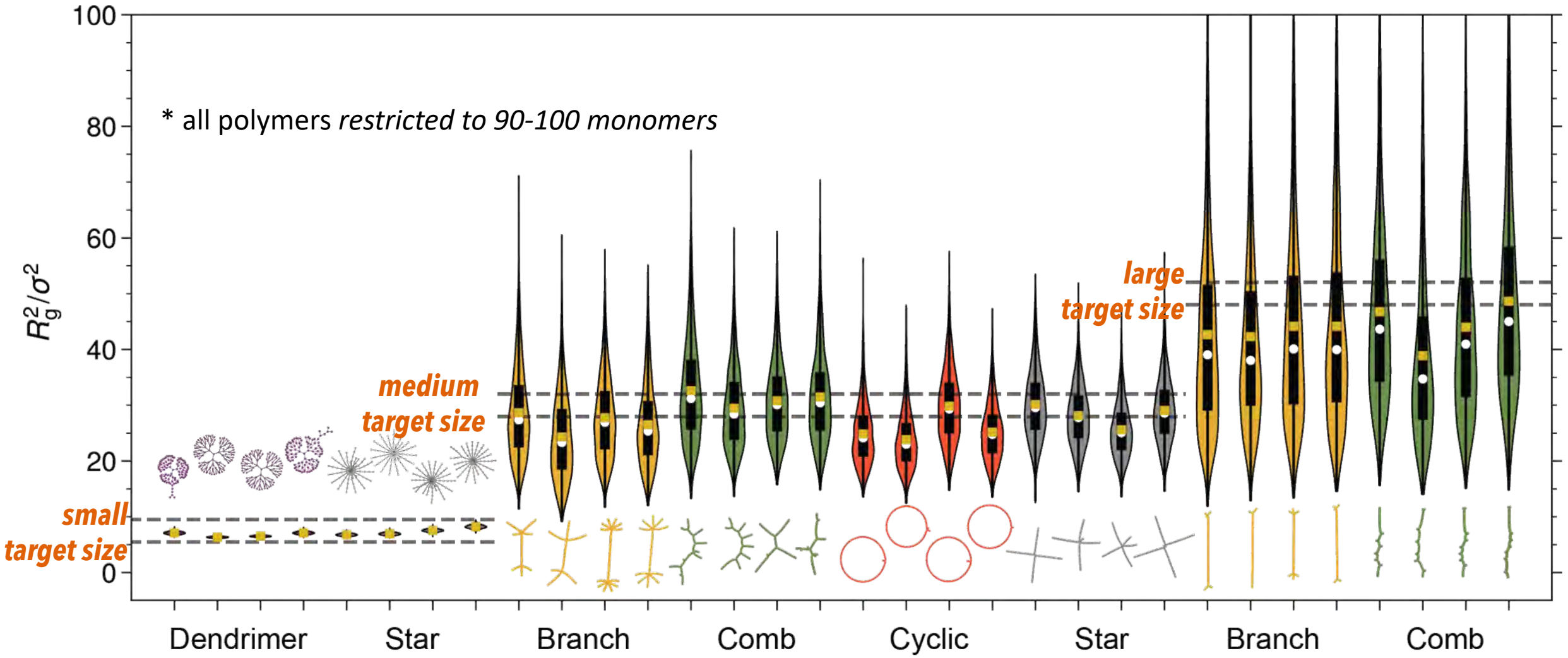


Topological classes



- **Task: Unsupervised Learning, UMAP**
- We find that including learning tasks based on *physical aspects of the polymers* facilitates architecture reconstruction and effective property prediction.
- We can *conditionally "sample" architectures (and hybrids)* from the 8-dimensional space according to desired constraints.

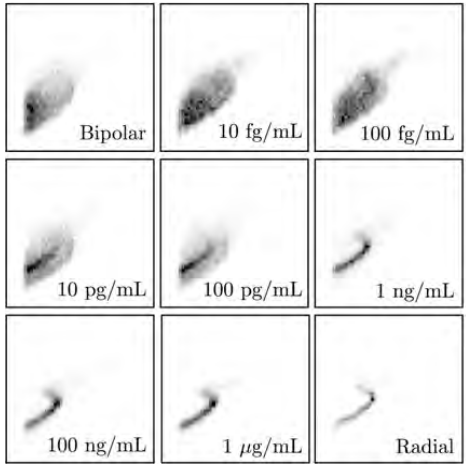
Example on Polymer Science



The VAE can design polymer topologies with a desired size in dilute solution, validated through simulation.

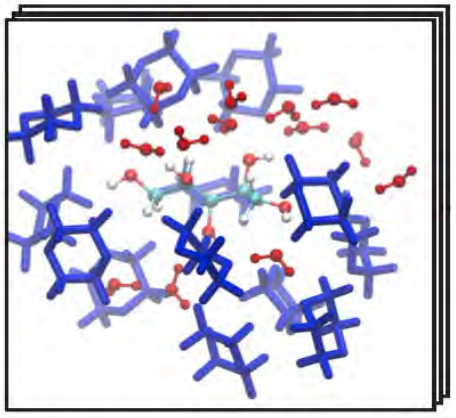
Summary

Scatter Points



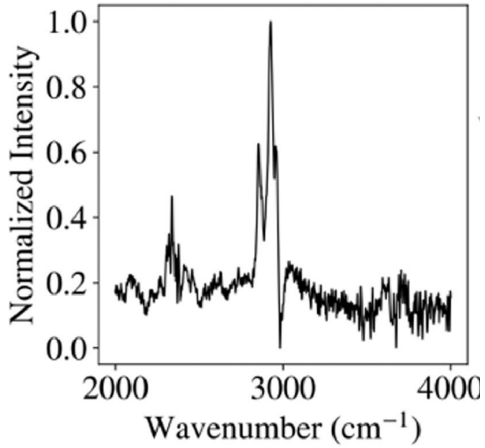
Histogram (2D Tensor)

Simulation Trajectories



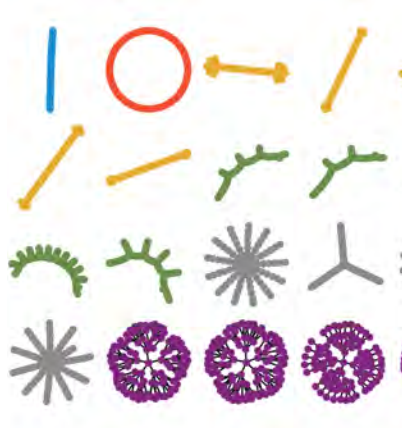
Voxel (3D Tensor)

IR Spectra



Gramian Angular Field (2D Tensor)

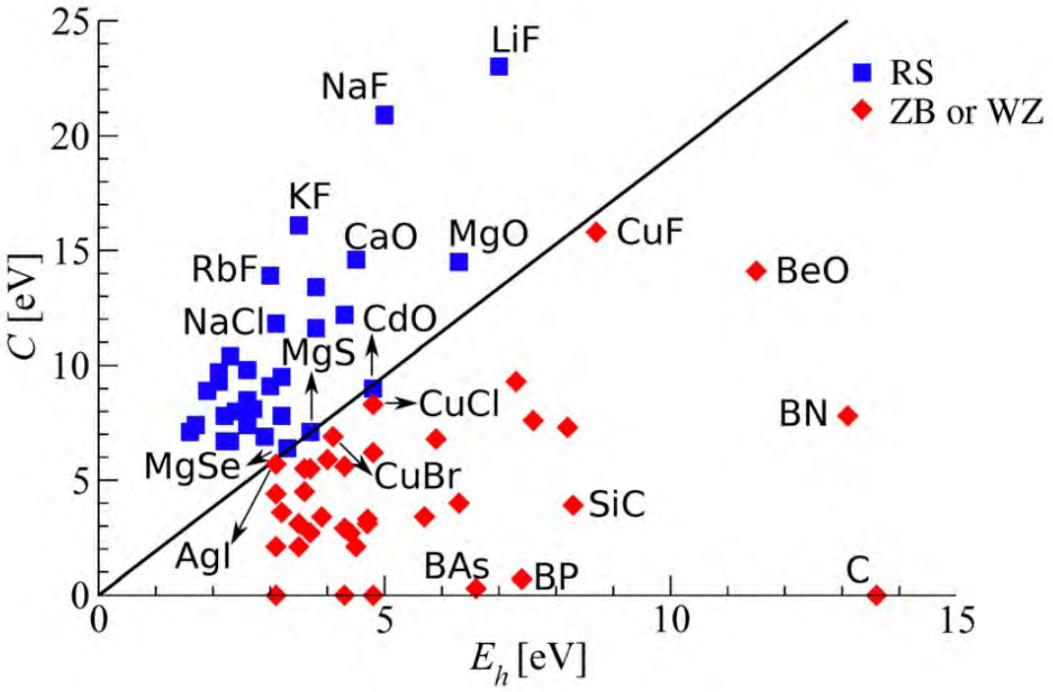
Polymers



Graph + 1D Tensor

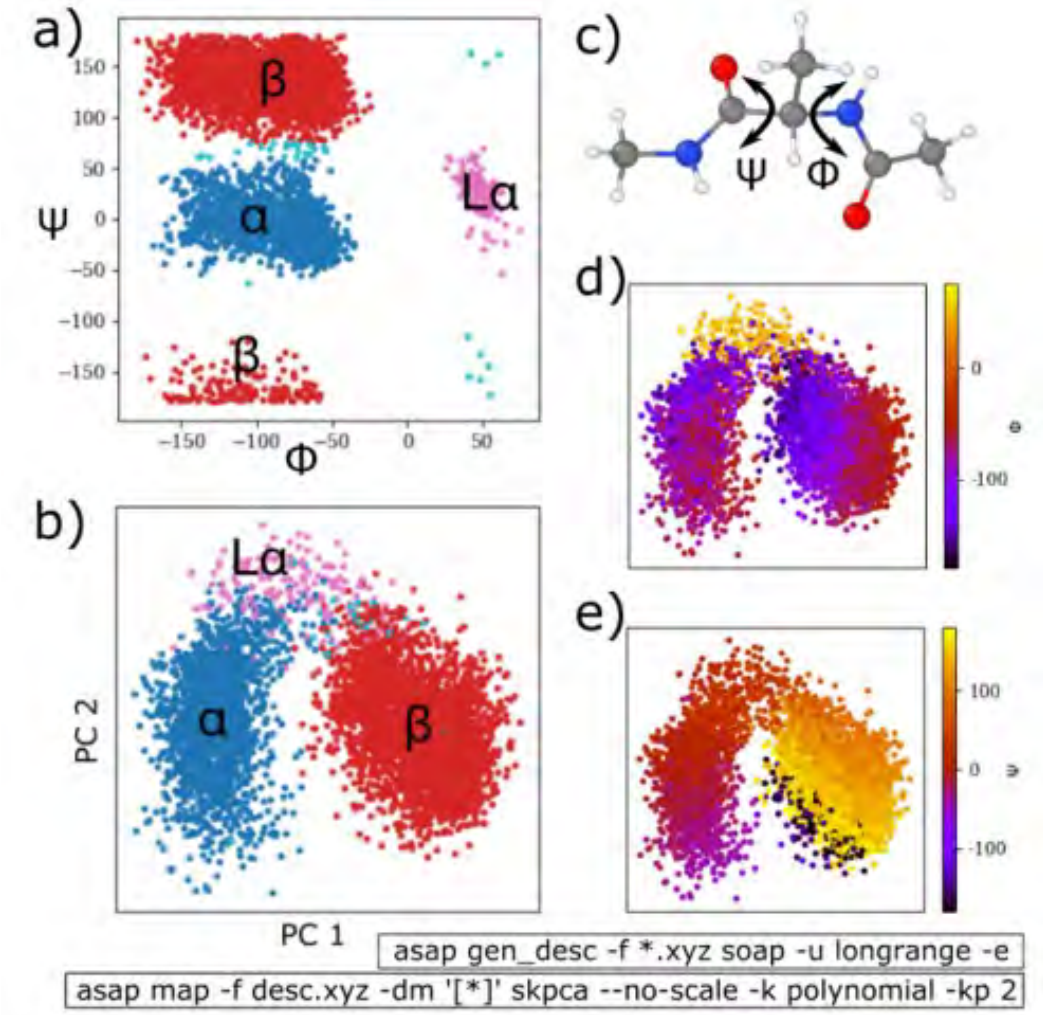
- *ML models can handle different types of data inputs and even multiple inputs at once.*
- *The output changes based on the task (classification, regression), and multi-output models are common.*
- *Including physical knowledge in the model input or using it as an additional learning task can sometimes enhance model performance.*

A couple more examples



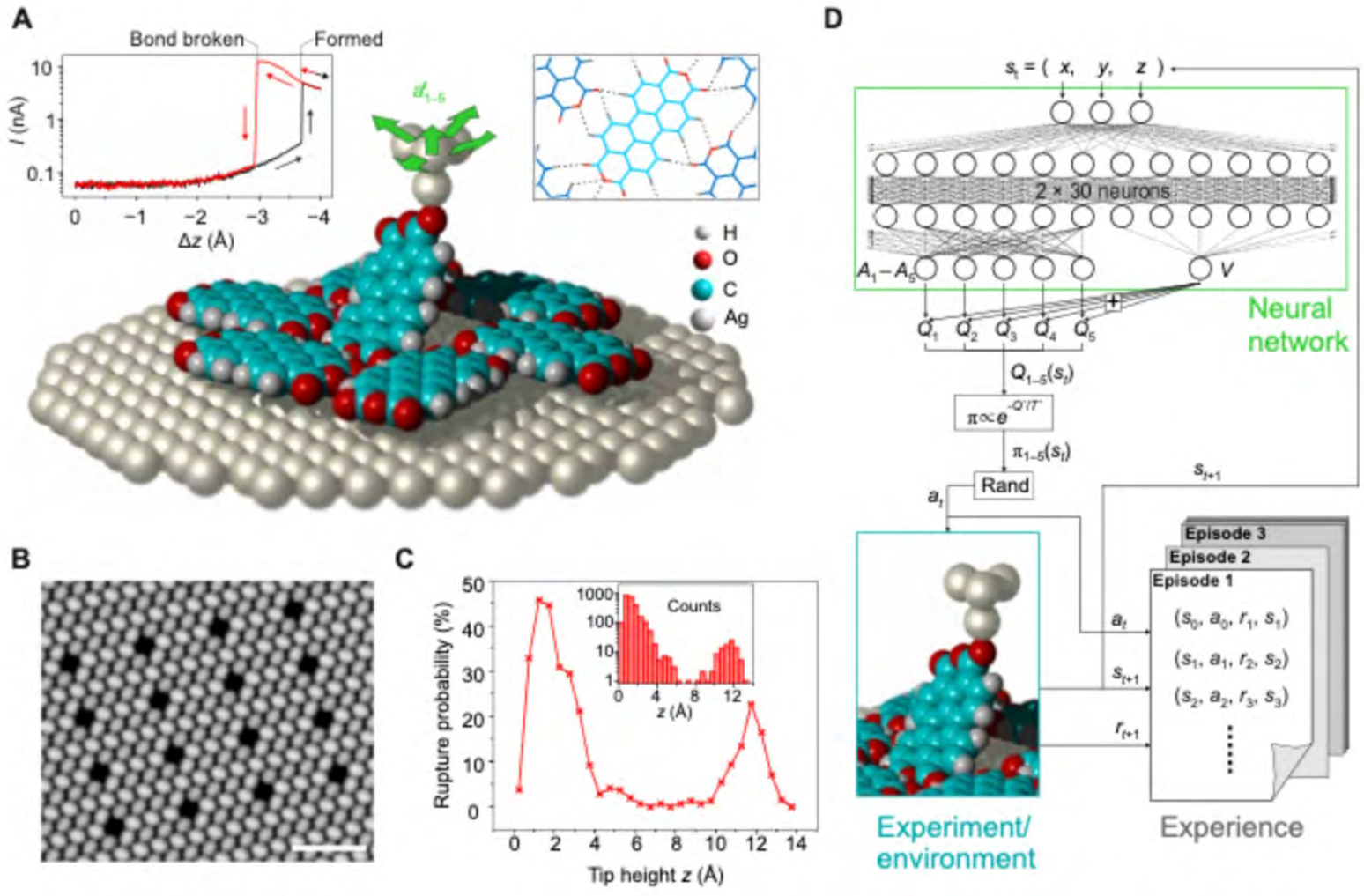
Ghiringhelli et al. *PRL* 114 (**2015**)

Two simple ”descriptors” (related to nearest neighbor distance and dielectric constant) define a function that serves as a decision boundary that distinguishes between rocksalt and zinc blend or wurtzite crystal structures.



This illustrates a typical Ramachandran plot of alanine dipeptide by comparison to a unsupervised learning over molecular configurations.

A couple more examples



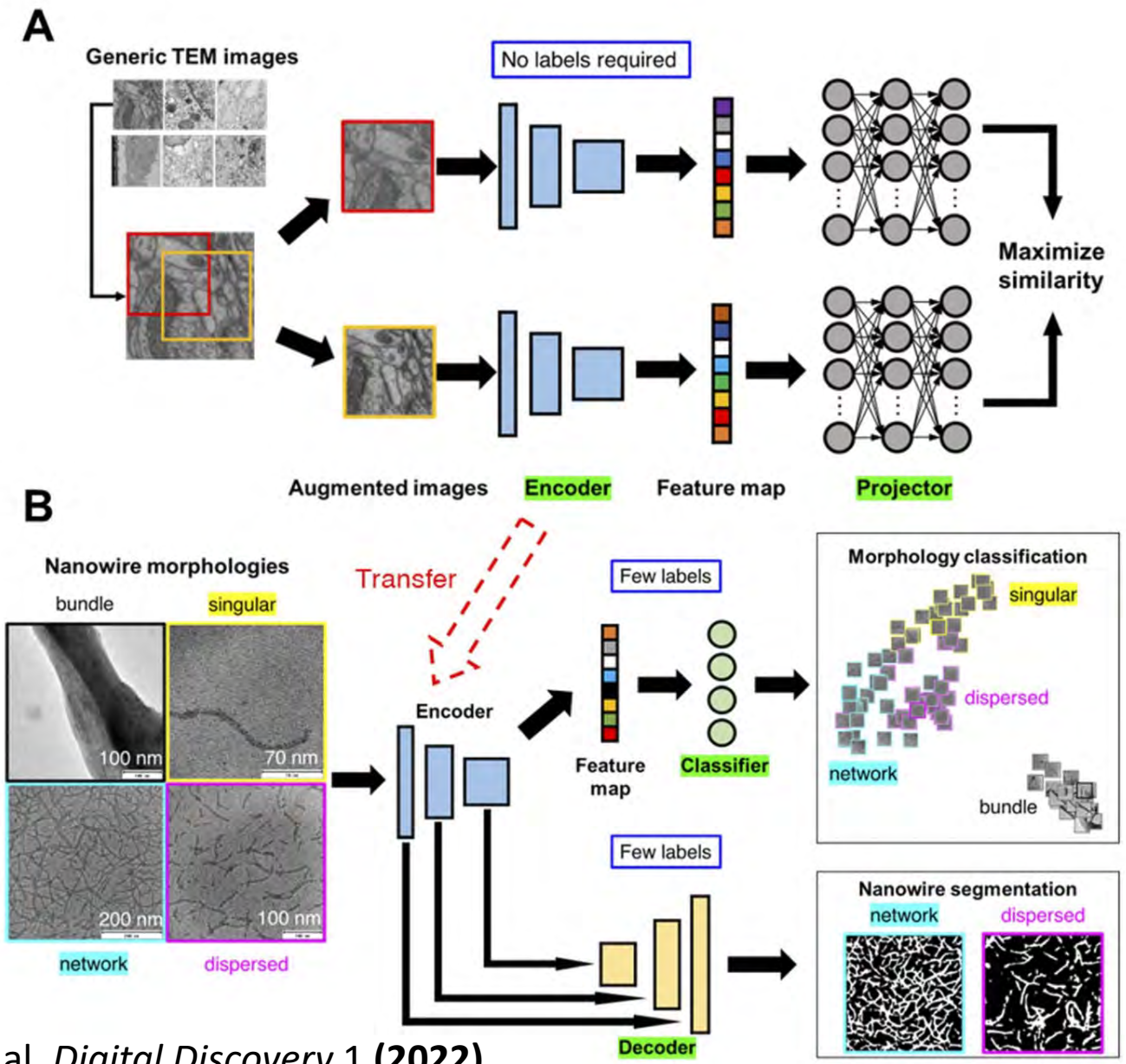
Guiding nanofabrication with single-molecule manipulation

- Scanning probe microscope can remove molecules from supramolecular assembly, but apparently this is non-trivial manual task
- Reinforcement learning is used to develop a protocol to move the tip in a manner that enables effective molecule lifting

We design the reward system as follows: If the environment transitions to a nonterminal state, we assign a default reward of $r_{t+1} = 0.01$ (see Materials and Methods for a discussion). If transitioning into a state in which the SPM tip loses contact with the molecule, the agent is penalized with $r_{t+1} = -1$, and the current episode stops. Last, if transitioning into a state where the molecule has been lifted successfully, we assign a reward of $r_{t+1} = +1$, and the episode also stops. After each failed episode, the molecule, by virtue of

einen et al. *Science Advances 6* **(2020)**

A couple more examples



Lu et al. *Digital Discovery* 1 **(2022)**

"Fancy" ML workflow for microscopy segmentation & classification

- Combines many "advanced" architecture concepts with semi-supervised approach in a "transfer learning" paradigm.
- Self-supervised learning component comes from matching an image to itself! (they must come from the same class… probably?)
- Overall goal is efficient labeling of TEM/data efficiency