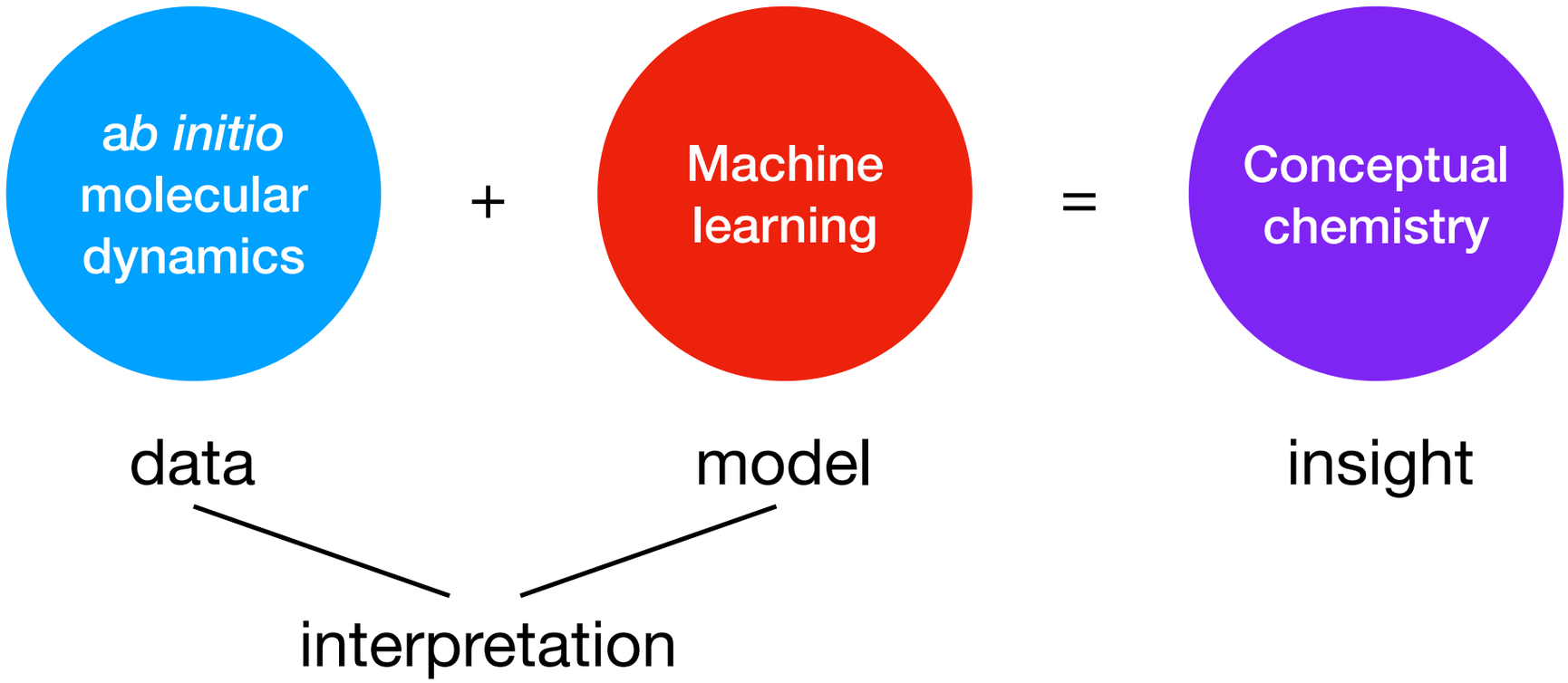
Interpretable Machine Learning for Conceptual Chemistry

Joshua Goings, PhD
Nov 25 2020

Roadmap

Goal: physical insight into reaction dynamics

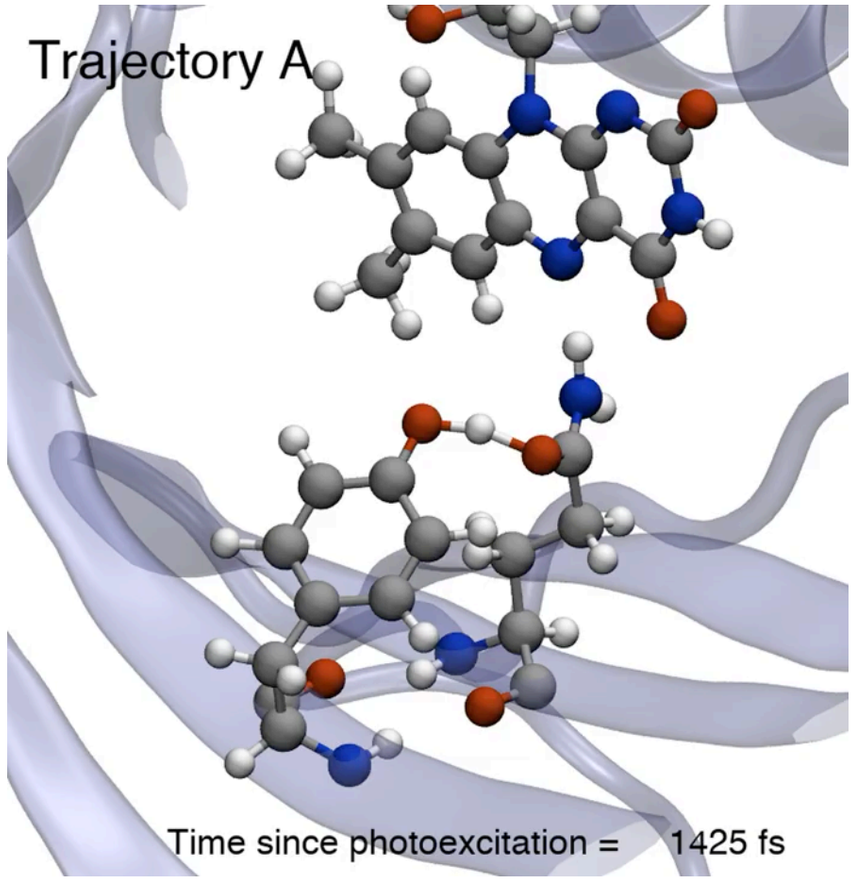


data → model → interpretation → insight!

Molecular dynamics

Simulate molecules or materials in time

1. Chemical reaction rates
2. Reaction yields
3. Mechanisms
4. (Non-)equilibrium properties



Active site of BLUF photoreceptor

Generates lots of data (position and velocity at each time point)

Much of current MD seeks better ways to connect raw trajectory data to experimental observables and physical insight.

for classical MD on a protein (10,000 atoms) for 1 nanosecond:

e.g. 10^4 atoms x 10^6 time steps x 6 coords/atom x 64 bit ~ 0.5 TB

JJG; Hammes-Schiffer, S. *JACS*. **2019.**

Flavors of molecular dynamics

Include different physics for different purposes

	Electrons	Nuclei	Accuracy	Cost

Flavors of molecular dynamics

Include different physics for different purposes

	Electrons	Nuclei	Accuracy	Cost
Classical MD	None	Classical force field	Low (empirical)	> 10,000 atoms

Flavors of molecular dynamics

Include different physics for different purposes

	Electrons	Nuclei	Accuracy	Cost
Classical MD	None	Classical force field	Low (empirical)	> 10,000 atoms
Born-Oppenheimer MD	Quantum (*ab initio*)	Classical, forces from electrons	Medium	10-100 atoms

Flavors of molecular dynamics

Include different physics for different purposes

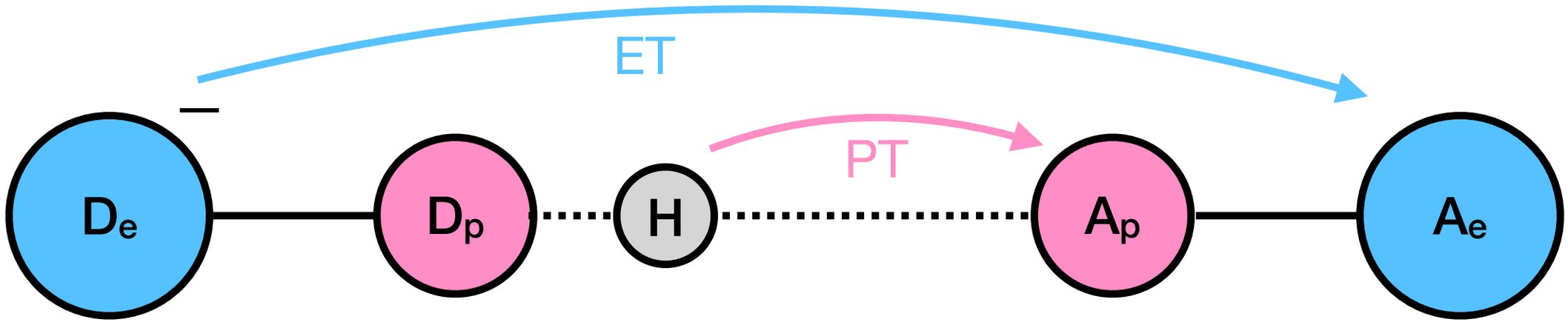
	Electrons	Nuclei	Accuracy	Cost
Classical MD	None	Classical force field	Low (empirical)	> 10,000 atoms
Born-Oppenheimer MD	Quantum (*ab initio*)	Classical, forces from electrons	Medium	10-100 atoms
RPMD, FSSH, MCTDH, etc.	Quantum	Quantum	High	< 10 atoms

Flavors of molecular dynamics

Include different physics for different purposes

	Electrons	Nuclei	Accuracy	Cost
Classical MD	None	Classical force field	Low (empirical)	> 10,000 atoms
Born-Oppenheimer MD	Quantum (*ab initio*)	Classical, forces from electrons	Medium	10-100 atoms
RPMD, FSSH, MCTDH, etc.	Quantum	Quantum	High	< 10 atoms

Proton-coupled electron transfer (PCET) theory



<u>Key idea: motion of protons and electrons are not independent (*coupled*)</u>

Many variations of PCET:

- Do the electrons & protons involve same or different donors/acceptors?
- Do the electrons & protons move in same or different directions?
- Do the electrons & protons move together or sequentially?
- Are multiple electrons and/or protons involved?

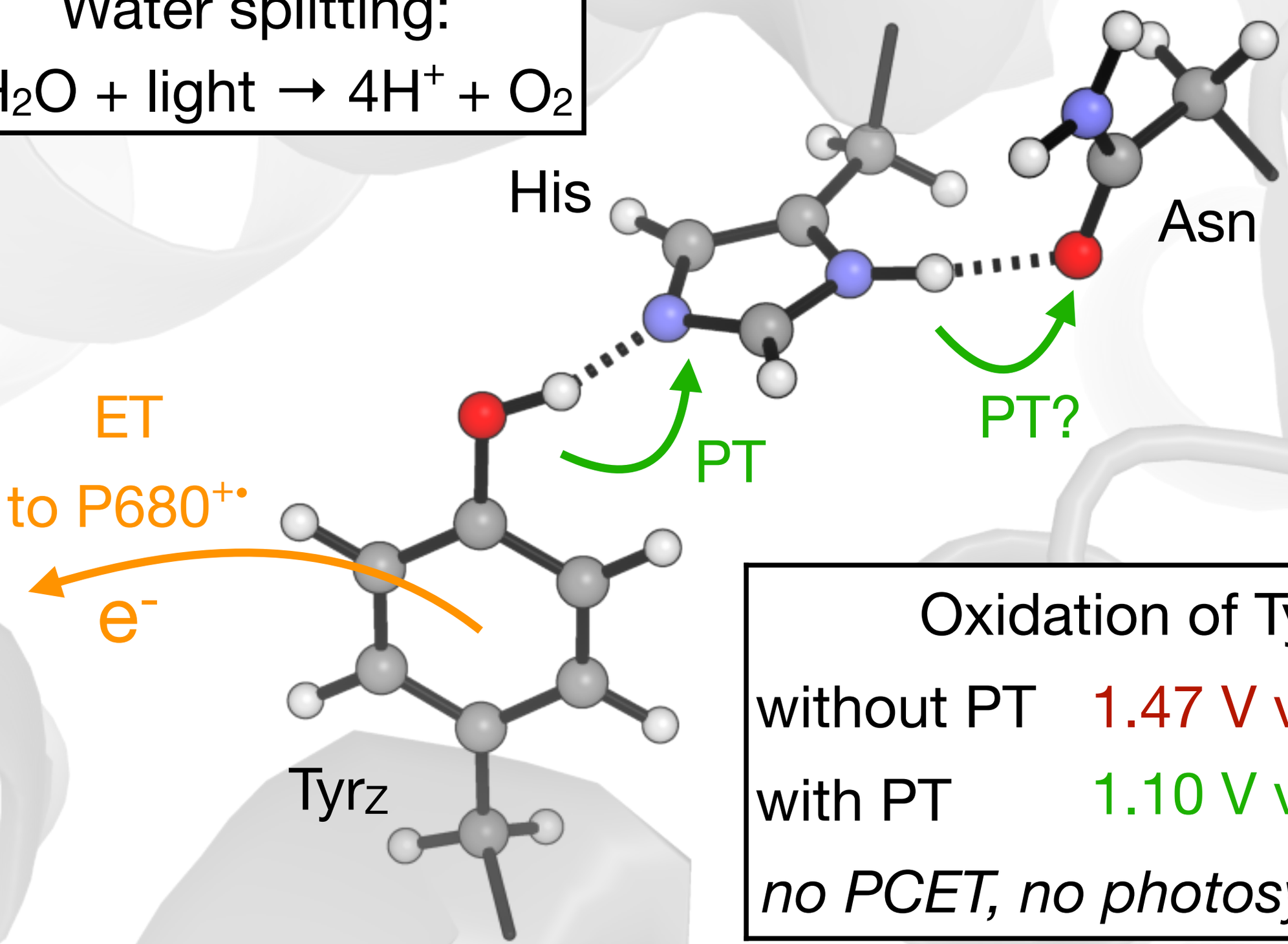
Quantum mechanical effects of electrons and proton(s)?

Different timescales and couplings among electrons, protons, solvent, environment, donor-acceptor modes, etc.

PCET in photosystem II

Water splitting:

$2H_2O + light \rightarrow 4H^+ + O_2$



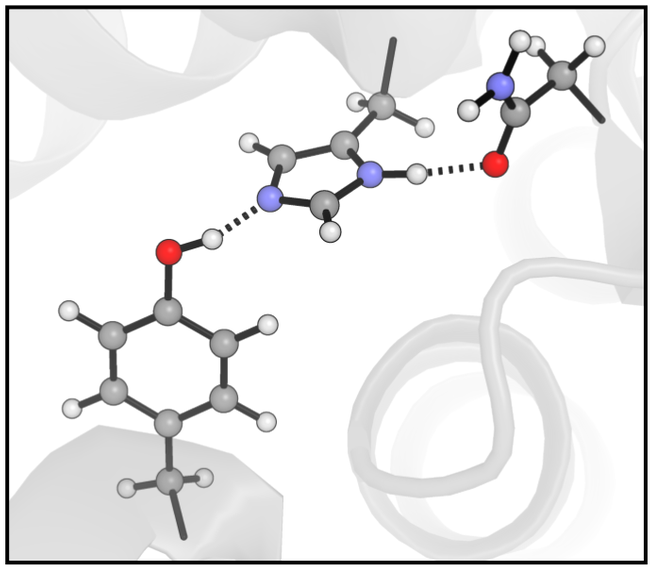
Oxidation of Tyr_Z

without PT	1.47 V vs NHE
with PT	1.10 V vs NHE

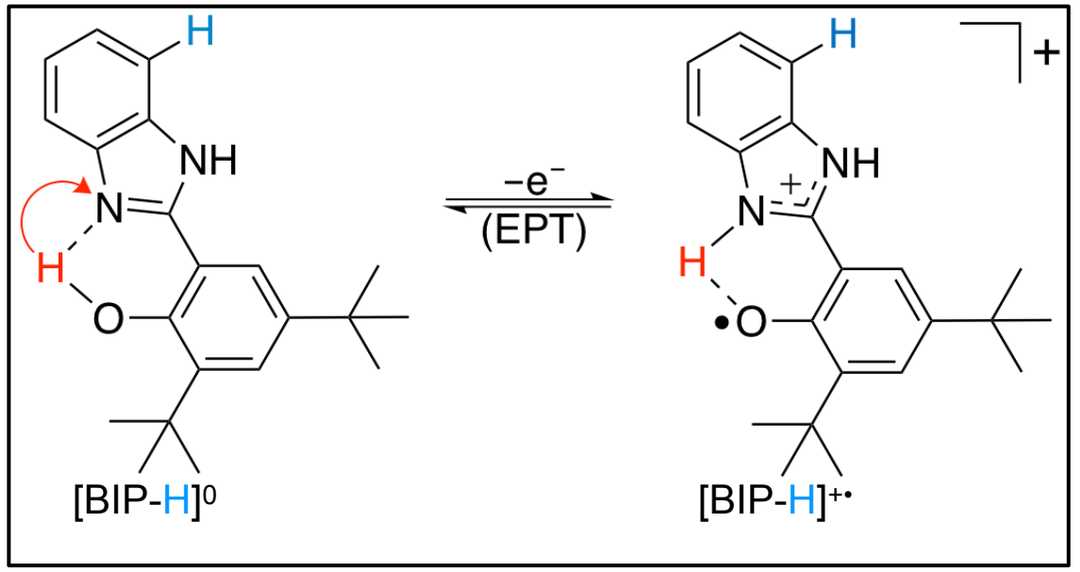
no PCET, no photosynthesis!

Imitating nature: bioinspired systems for PCET

Inspired by the Tyr/His redox relay in photosystem II, set out to create redox-active molecules to drive proton translocation.



photosystem II



benzimidazolephenol (BIP) constructs

Simplest case: reversible PCET upon oxidation of phenol in BIP.

How do the molecular motions influence this process?

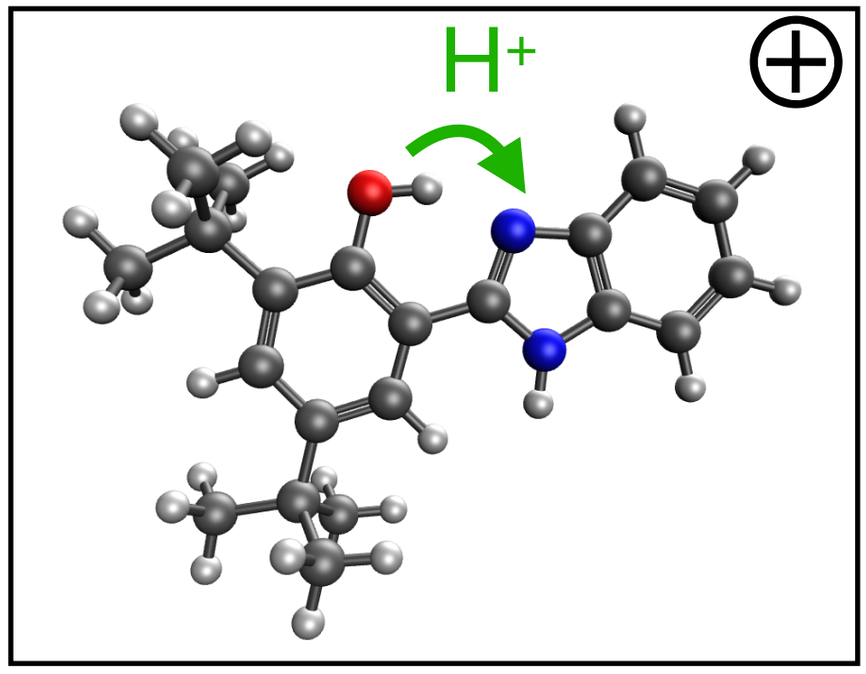
Moore et. al, JPCB, 2010, 114, 14450

ab initio molecular dynamics

After oxidation, how long until proton transfer?
What are the molecular motions that facilitate this process?

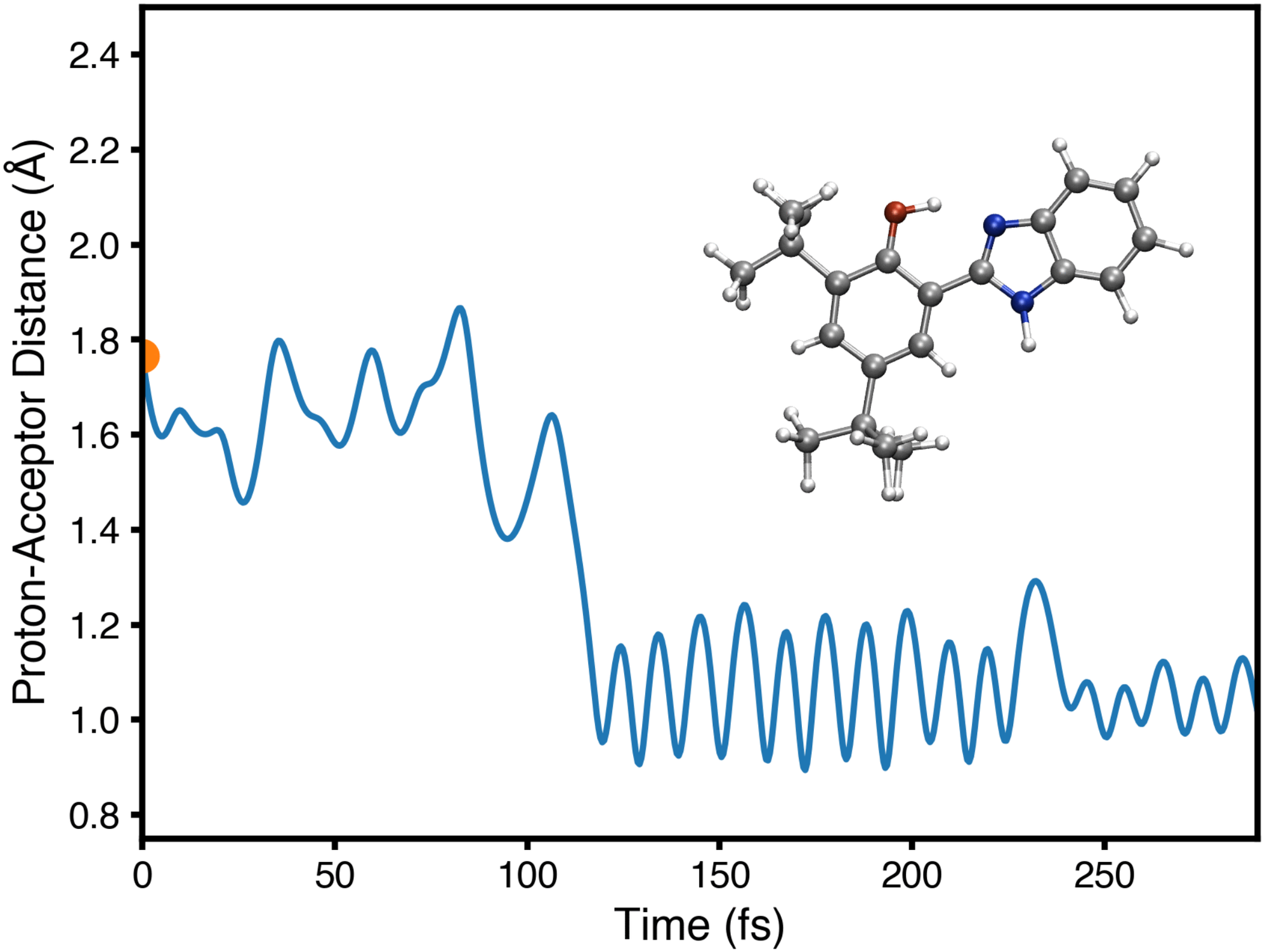
Ensemble of trajectories

- 240 independent, classical trajectories
- initial coordinates/momenta from neutral, but propagate in the oxidized state
- B3LYP-D3(BJ)/6-31G**
- gas phase
- run for at least 300 fs with 0.5 fs timestep
- trajectory gives ~ 0.5 GB data



J. Chem. Phys. 88, 4535 (1988); https://doi.org/10.1063/1.453761
Int. J. Quantum Chem. 2016, 116 (10), 762–771. https://doi.org/10.1002/qua.25049

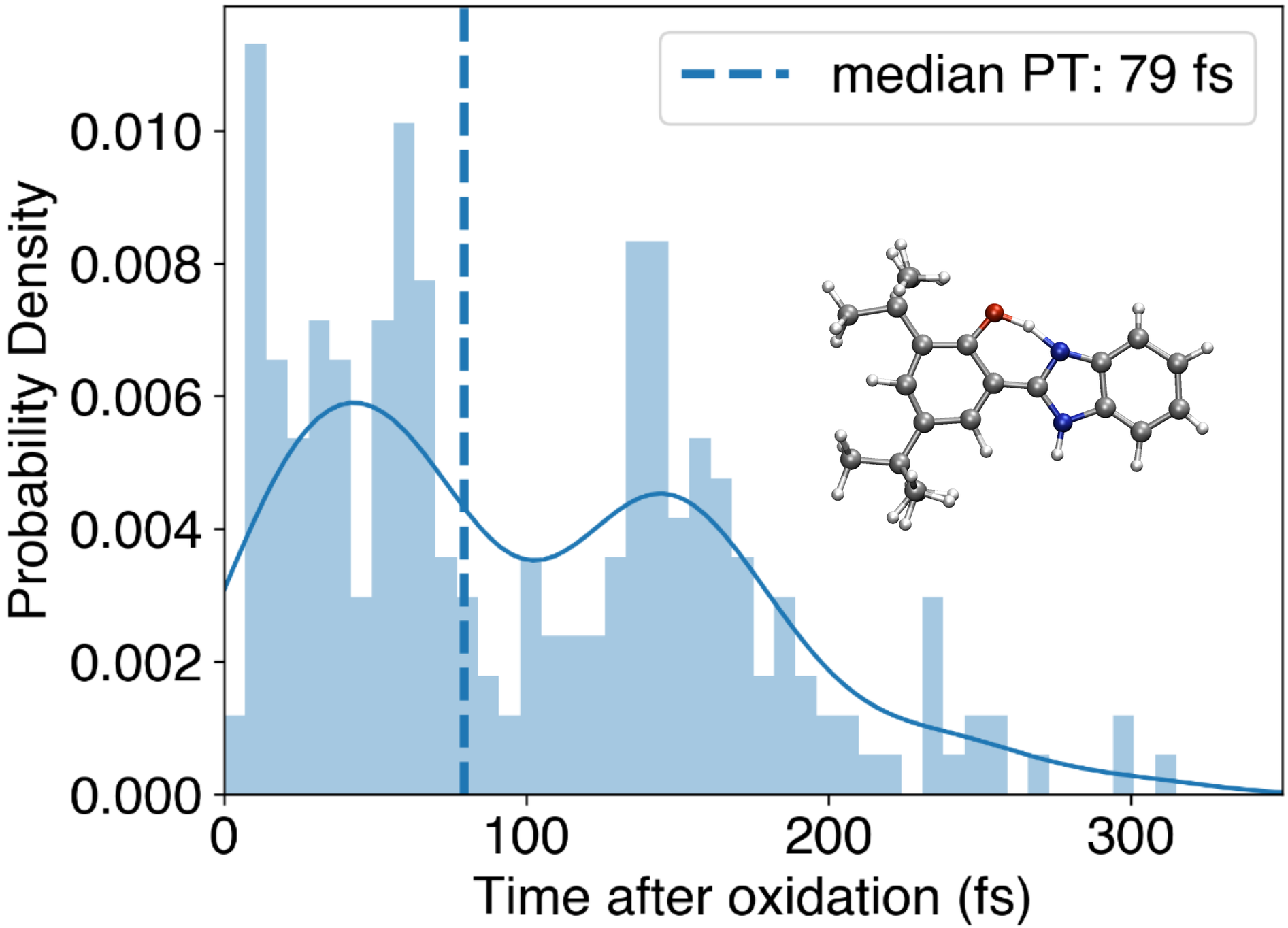
Movie time


Proton-Acceptor Distance (Å)
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0
50
100
150
200
250
Time (fs)

Proton transfer (PT) after oxidation

All 240 trajectories show proton transfer within ~300 fs.

Proton transfer appears to follow a bimodal distribution (more on this later).



Limitations for human understanding

Making sense of vast amounts of data

Statistics certainly help summarize the distribution of data (PT times)

Statistics also help connect simulation to experiment

Is there a way to obtain more information from the data we generated?

Limitations for human understanding

Making sense of vast amounts of data

Statistics certainly help summarize the distribution of data (PT times)

Statistics also help connect simulation to experiment

Is there a way to obtain more information from the data we generated?

We would like to know:

- What physical motions correlate with proton transfer time?
- How does the molecule respond dynamically to oxidation?
- Why is the distribution of proton transfer times bimodal? Is this noise, or is there something more?

Limitations for human understanding

Making sense of vast amounts of data

Statistics certainly help summarize the distribution of data (PT times)

Statistics also help connect simulation to experiment

Is there a way to obtain more information from the data we generated?

We would like to know:

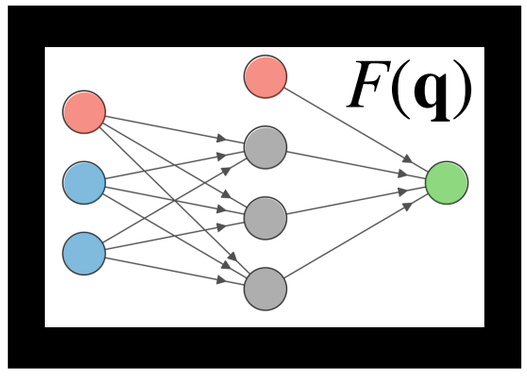
- What physical motions correlate with proton transfer time?
- How does the molecule respond dynamically to oxidation?
- Why is the distribution of proton transfer times bimodal? Is this noise, or is there something more?

Where do we start looking?

Can we use machine learning to point out places to look?

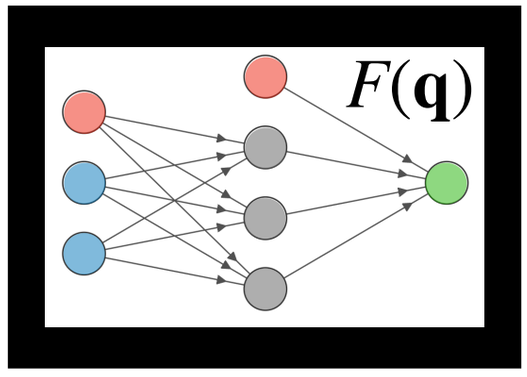
Universal approximation theorem

A neural network with a single finite-width hidden layer can approximate any continuous real function to arbitrary precision.



Universal approximation theorem

Neural nets are function-generators!



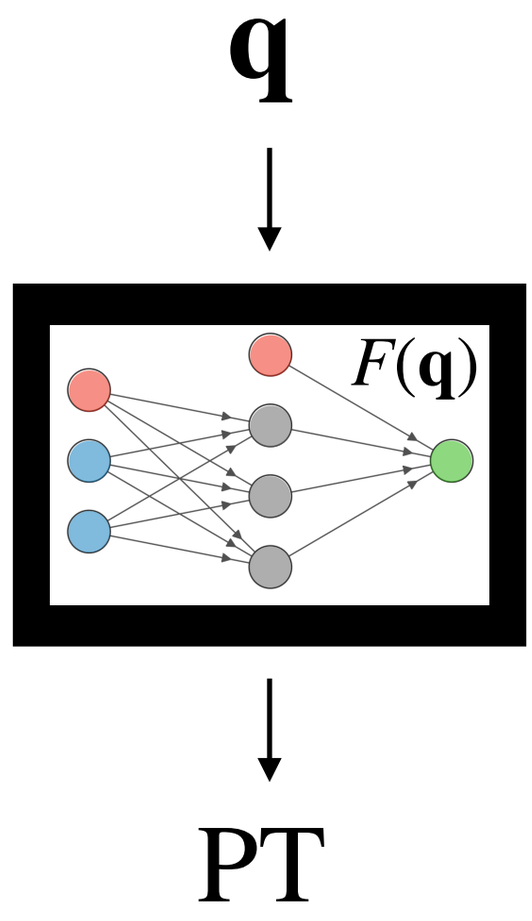
Universal approximation theorem

Neural nets are function-generators!

Two goals:

1. Find a function $F(\mathbf{q})$ to compute proton transfer time PT as a function of atomic coordinates $\mathbf{q}$.

2. Given the function, explain predictions to determine important structural changes during reaction.



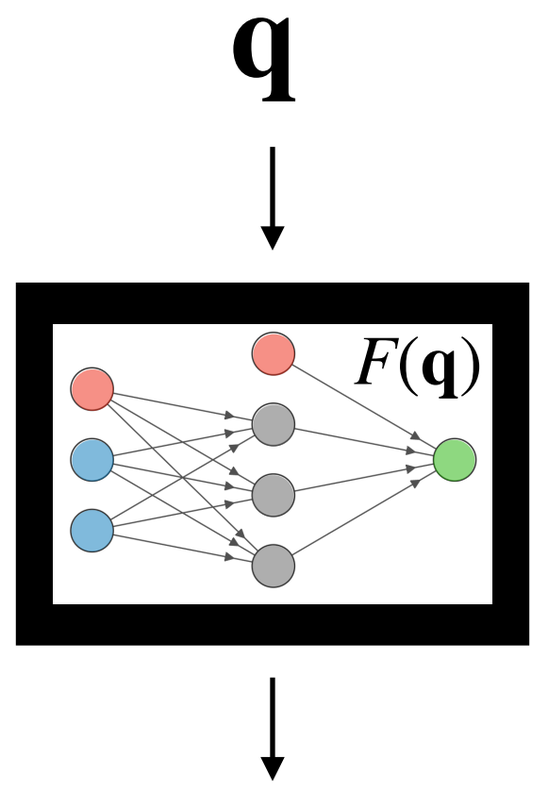
Universal approximation theorem

Neural nets are function-generators!

Two goals:

1. Find a function $F(\mathbf{q})$ to compute proton transfer time PT as a function of atomic coordinates $\mathbf{q}$.

2. Given the function, explain predictions to determine important structural changes during reaction.



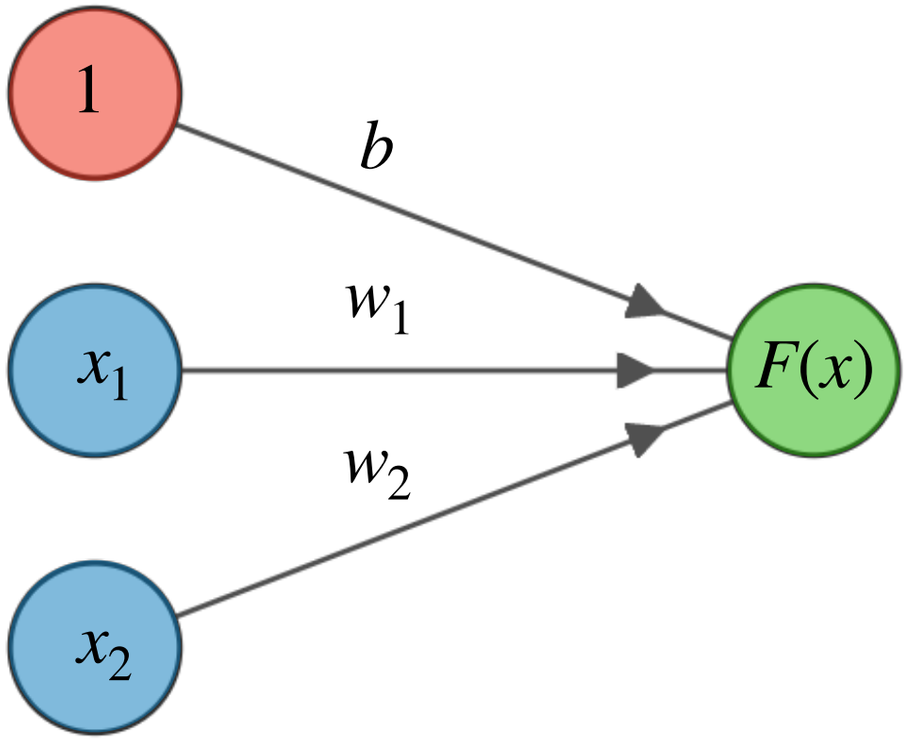
Our goal is NOT to replace *ab initio* molecular dynamics.

Our goal is NOT to quantitatively predict all PT times for all conditions.

We want to use machine learning to assist in the interpretation of complex molecular dynamics.

Neural networks generalize regression

Linear regression

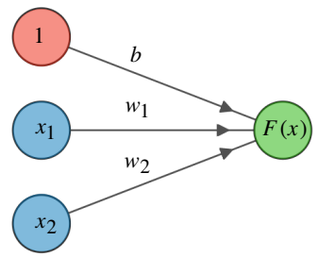


$$F(x) = w^T x + b$$

Input Layer $\in \mathbb{R}^3$ Output Layer $\in \mathbb{R}^1$

$$F(x) = w_1 \cdot x_1 + w_2 \cdot x_2 + 1 \cdot b$$

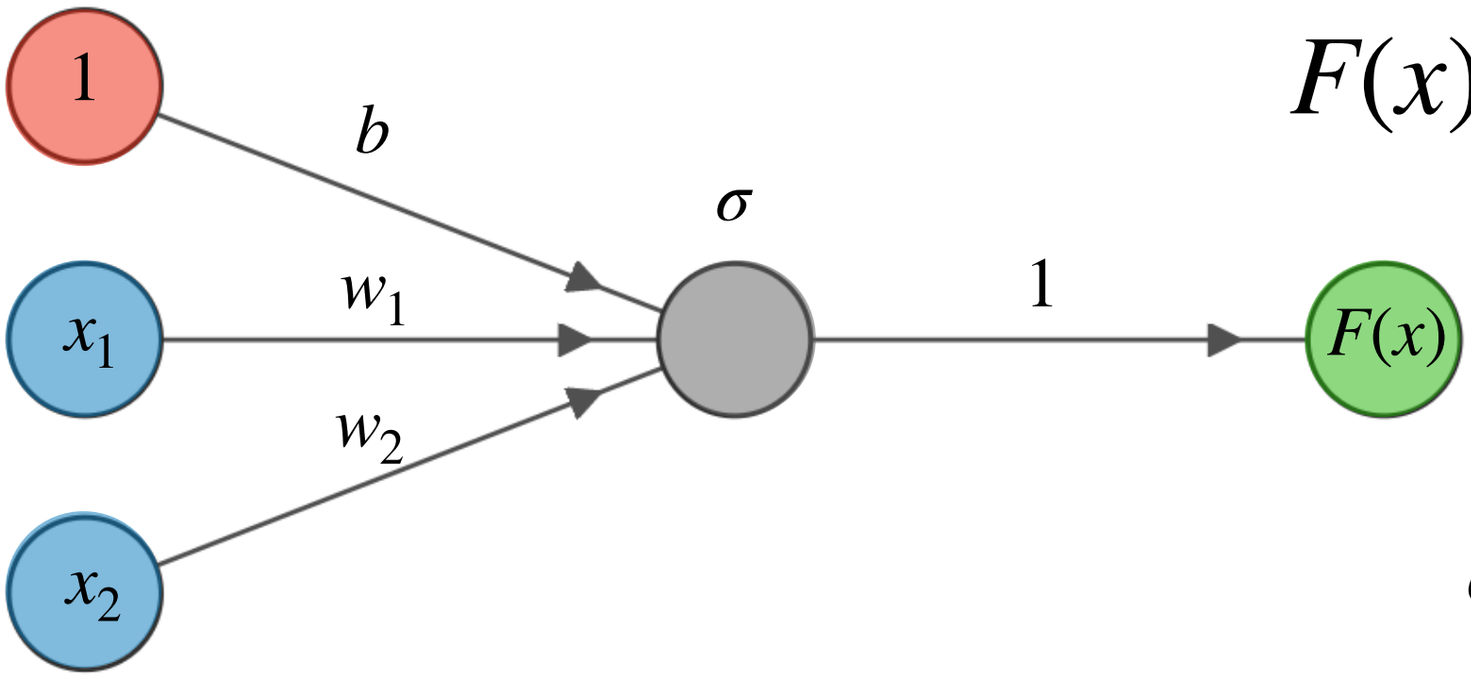
Linear regression



$$F(x) = w^T x + b$$

Input Layer $\in \mathbb{R}^3$ Output Layer $\in \mathbb{R}^1$

Logistic regression (generalized linear model)

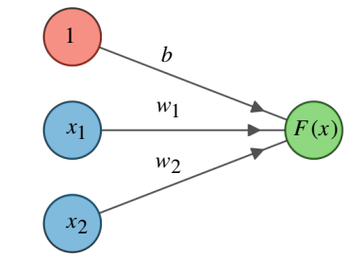


$$F(x) = \sigma(w^T x + b)$$

$$\sigma(h) = \frac{1}{1 + e^{-h}}$$

Input Layer $\in \mathbb{R}^3$ Hidden Layer $\in \mathbb{R}^1$ Output Layer $\in \mathbb{R}^1$

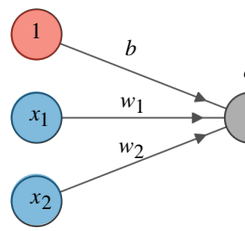
Linear regression



$$F(x) = w^T x + b$$

Input Layer $\in \mathbb{R}^3$ Output Layer $\in \mathbb{R}^1$

Logistic regression

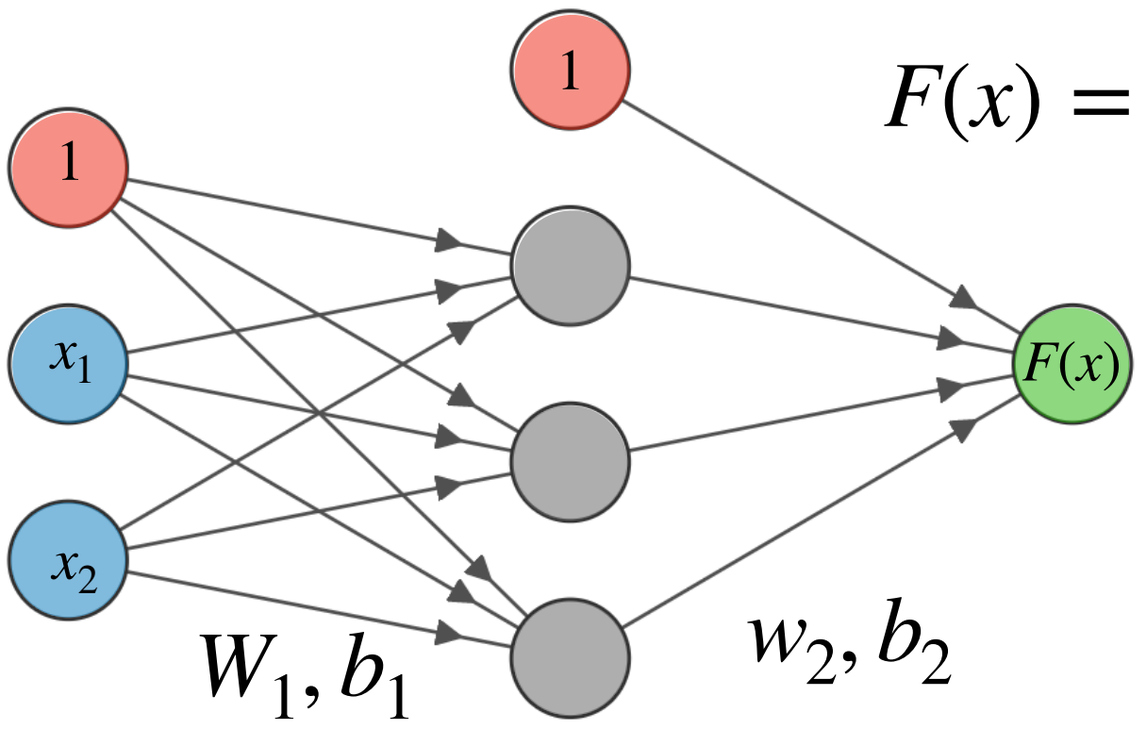


$$F(x) = \sigma(w^T x + b)$$

$$\sigma(h) = \frac{1}{1 + e^{-h}}$$

Input Layer $\in \mathbb{R}^3$ Hidden Layer $\in \mathbb{R}^1$ Output Layer $\in \mathbb{R}^1$

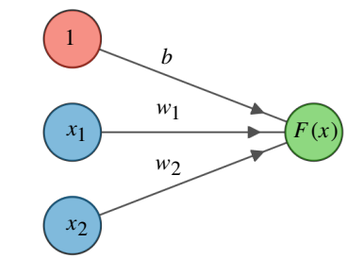
Shallow neural network (nonlinear regression)



$$F(x) = w_2^T \sigma(W_1^T x + b_1) + b_2$$

Input Layer $\in \mathbb{R}^3$ Hidden Layer $\in \mathbb{R}^4$ Output Layer $\in \mathbb{R}^1$

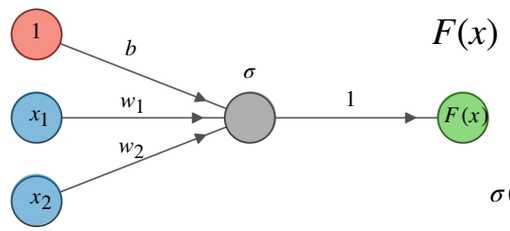
Linear regression



$$F(x) = w^T x + b$$

Input Layer $\in \mathbb{R}^3$ Output Layer $\in \mathbb{R}^1$

Logistic regression

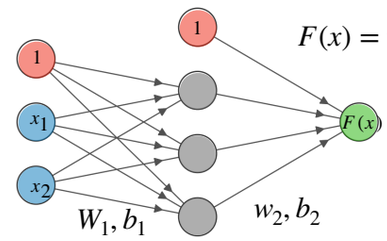


$$F(x) = \sigma(w^T x + b)$$

$$\sigma(h) = \frac{1}{1 + e^{-h}}$$

Input Layer $\in \mathbb{R}^3$ Hidden Layer $\in \mathbb{R}^1$ Output Layer $\in \mathbb{R}^1$

Shallow neural network

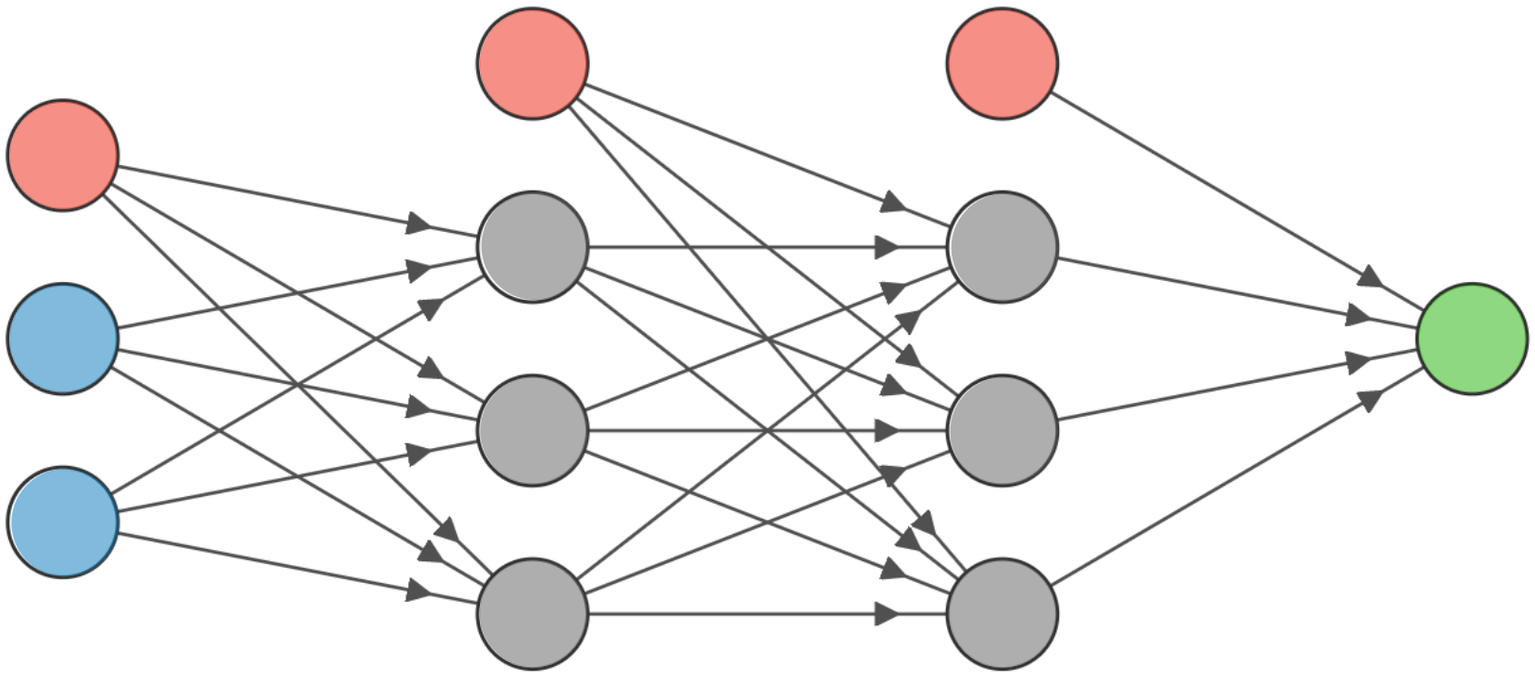


$$F(x) = w_2^T \sigma(W_1^T x + b_1) + b_2$$

Input Layer $\in \mathbb{R}^3$ Hidden Layer $\in \mathbb{R}^4$ Output Layer $\in \mathbb{R}^1$

Deep neural network (nonlinear regression)

$$F(x) = w_3^T \sigma(W_2^T \sigma(W_1^T x + b_1) + b_2) + b_3$$



Input Layer $\in \mathbb{R}^3$ Hidden Layer $\in \mathbb{R}^4$ Hidden Layer $\in \mathbb{R}^4$ Output Layer $\in \mathbb{R}^1$

Feedforward neural network

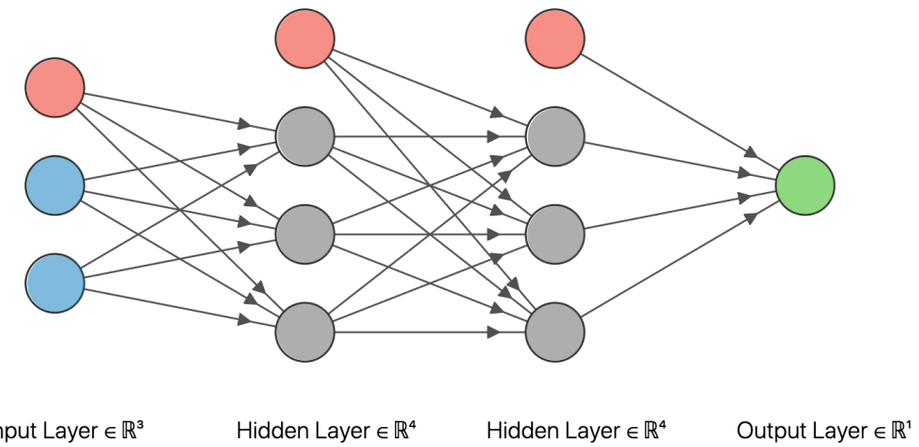
Training the network (optimize weights to minimize error)

Given a network:

$$y \approx F(x) = w_3^T \sigma(W_2^T \sigma(W_1^T x + b_1) + b_2) + b_3$$

And an error function:

$$E(x, y) = \frac{1}{2}\left(F(x) - y\right)^2$$



Optimize weights *via* steepest descent.

$$W_1^{(n+1)} = W_1^{(n)} - \alpha \frac{\partial E}{\partial W_1^{(n)}}$$

Gradients are computed with *backpropagation*…
which is just computer-science jargon for the chain rule.

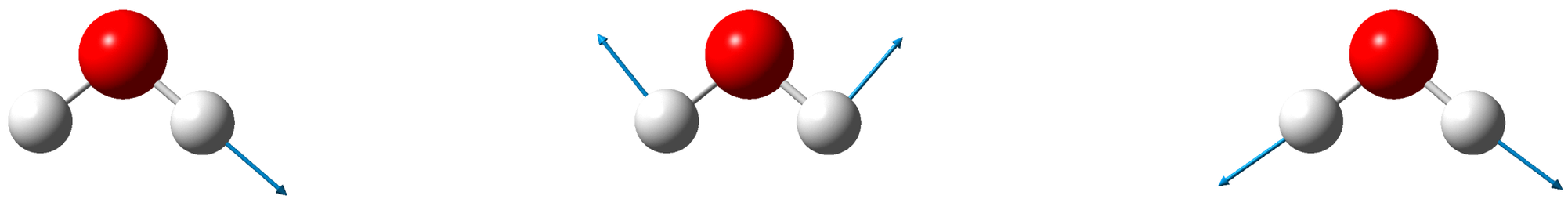
Feature selection and representation

For use in statistical models, we want features to be independent

Normal mode coordinates

Atomic coordinates in Cartesian coordinates are strongly coupled, and sensitive to rotations, translations, etc.

Molecular **normal mode basis** maps motions to an orthonormal (i.e. independent) basis **q**.

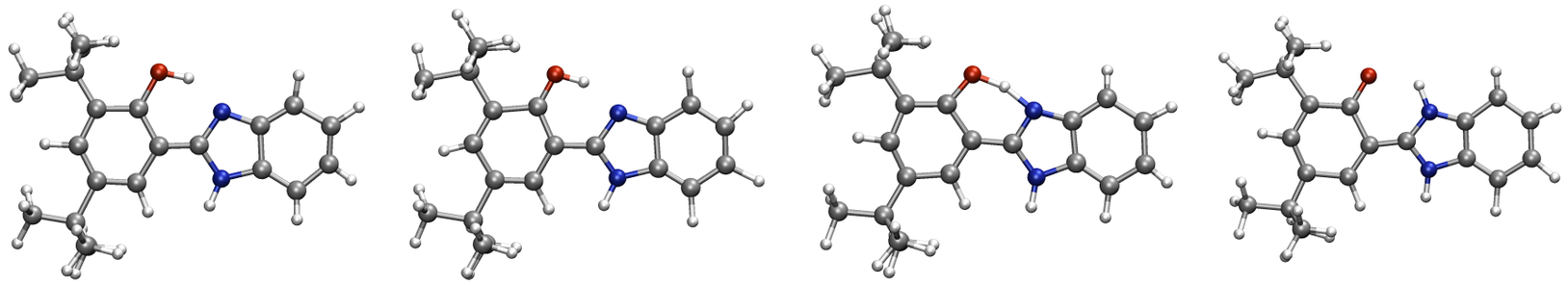


Now the motions of the molecule can be decomposed into independent features!

Generating data

Given coordinates and velocity, how long until PT?

For each trajectory:



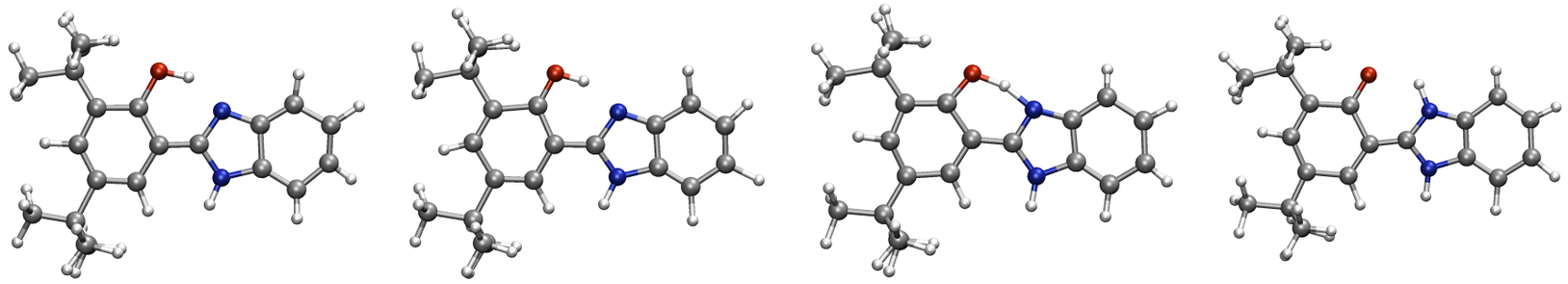
Real time	42 fs	44 fs	46 fs	48 fs
Time to PT	6 fs	4 fs	2 fs	0 fs

Time to PT	**q1**	**q2**	**…**	**p1**	**p2**	**…**
10	-0.1	0.1	…	0.2	-0.4	…
8	0.2	-0.3	…	0.1	-0.2	…
…	…	…	…	…	…	…
0	0.5	-0.1	…	-0.2	0.3	…

Generating data

Given coordinates and velocity, how long until PT?

For each trajectory:



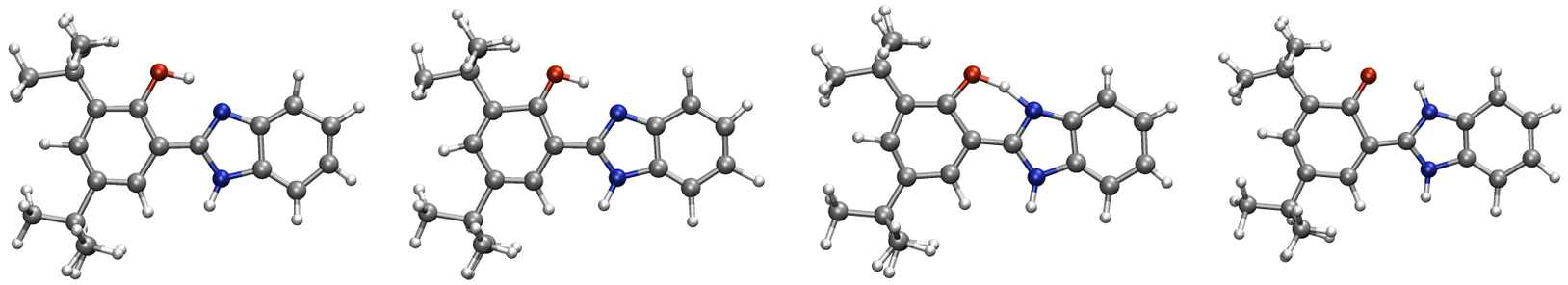
Real time	42 fs	44 fs	46 fs	48 fs
Time to PT	6 fs	4 fs	2 fs	0 fs

Time to PT	q1	q2	…	p1	p2	…
10	-0.1	0.1	…	0.2	0.4	…
8	0.2	-0.3	…	0.1	…	…
…	…	…	…	…	…	…
0	0.5	-0.1	…	-0.2	0.3	…

Generating data

Given coordinates and velocity, how long until PT?

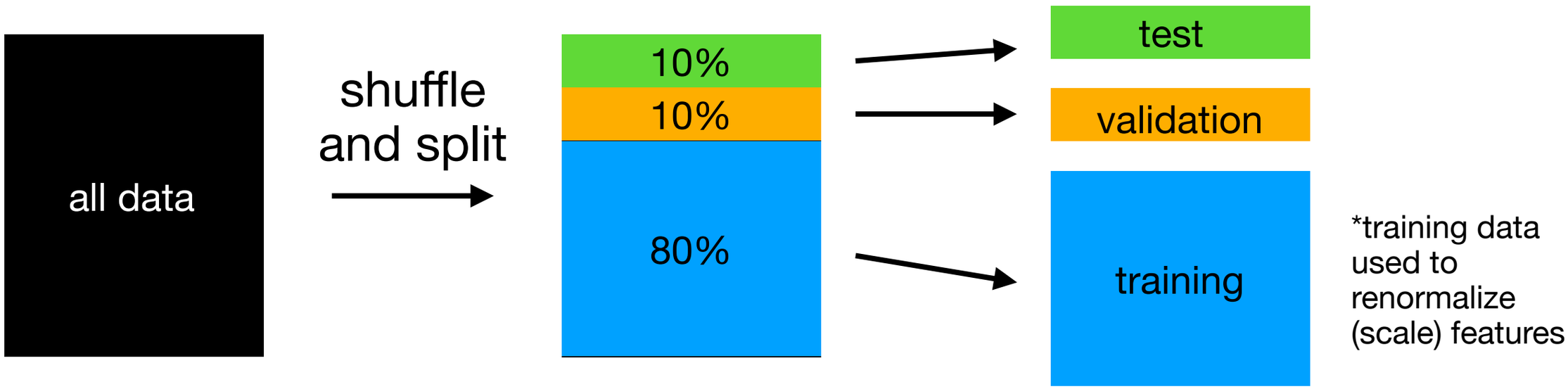
For each trajectory:



Real time	42 fs	44 fs	46 fs	48 fs
Time to PT	6 fs	4 fs	2 fs	0 fs

Time to PT	**q1**	**q2**	**…**	**q144**
10	-0.1	0.1	…	0.2
8	0.2	-0.3	…	0.1
…	…	…	…	…
0	0.5	-0.1	…	-0.2

Building the model



Grid search over hyperparameters (number layers, number neurons, etc.)

train model with:

evaluate performance with:

once model selected, report results with:

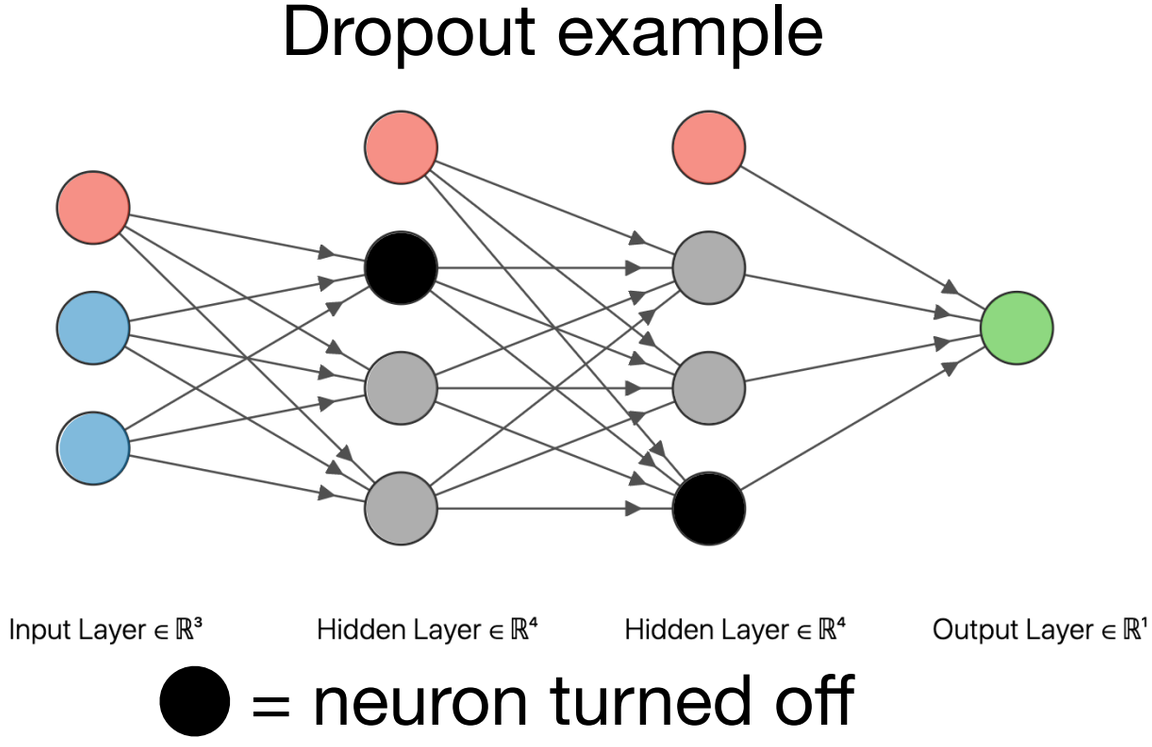
Final model has 3 layers, 768 neurons per layer, with 25% dropout.

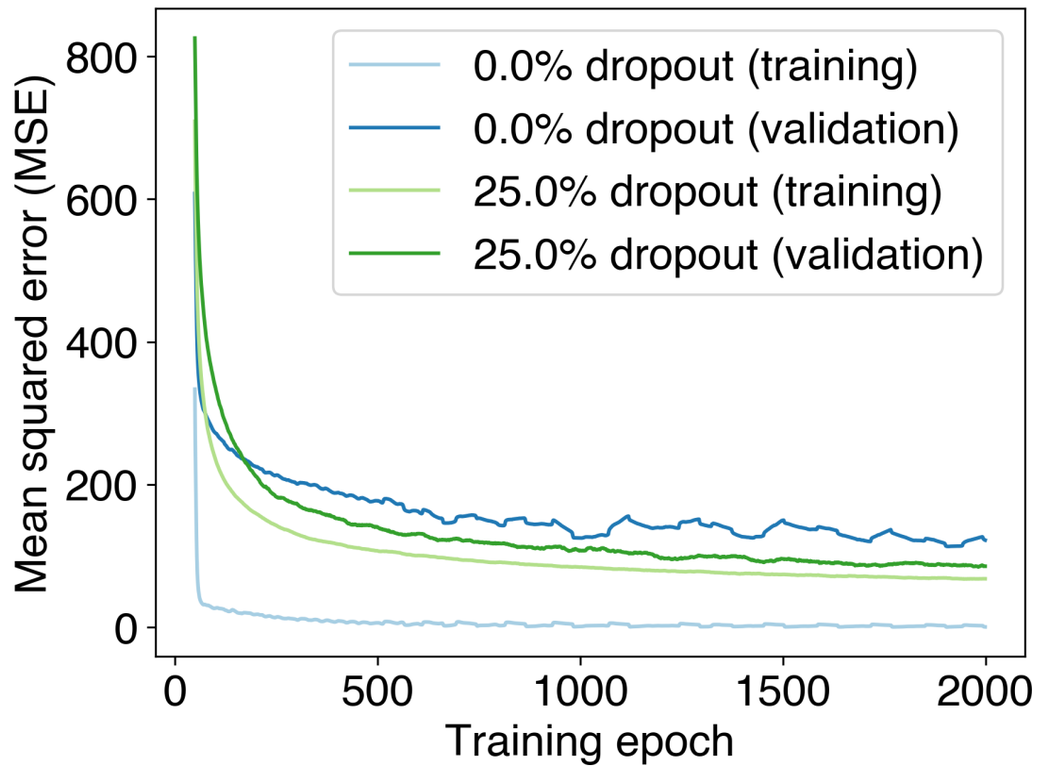
Preventing overfitting

Importance of dropout

We don’t want our model to just memorize the data. That’s not useful.

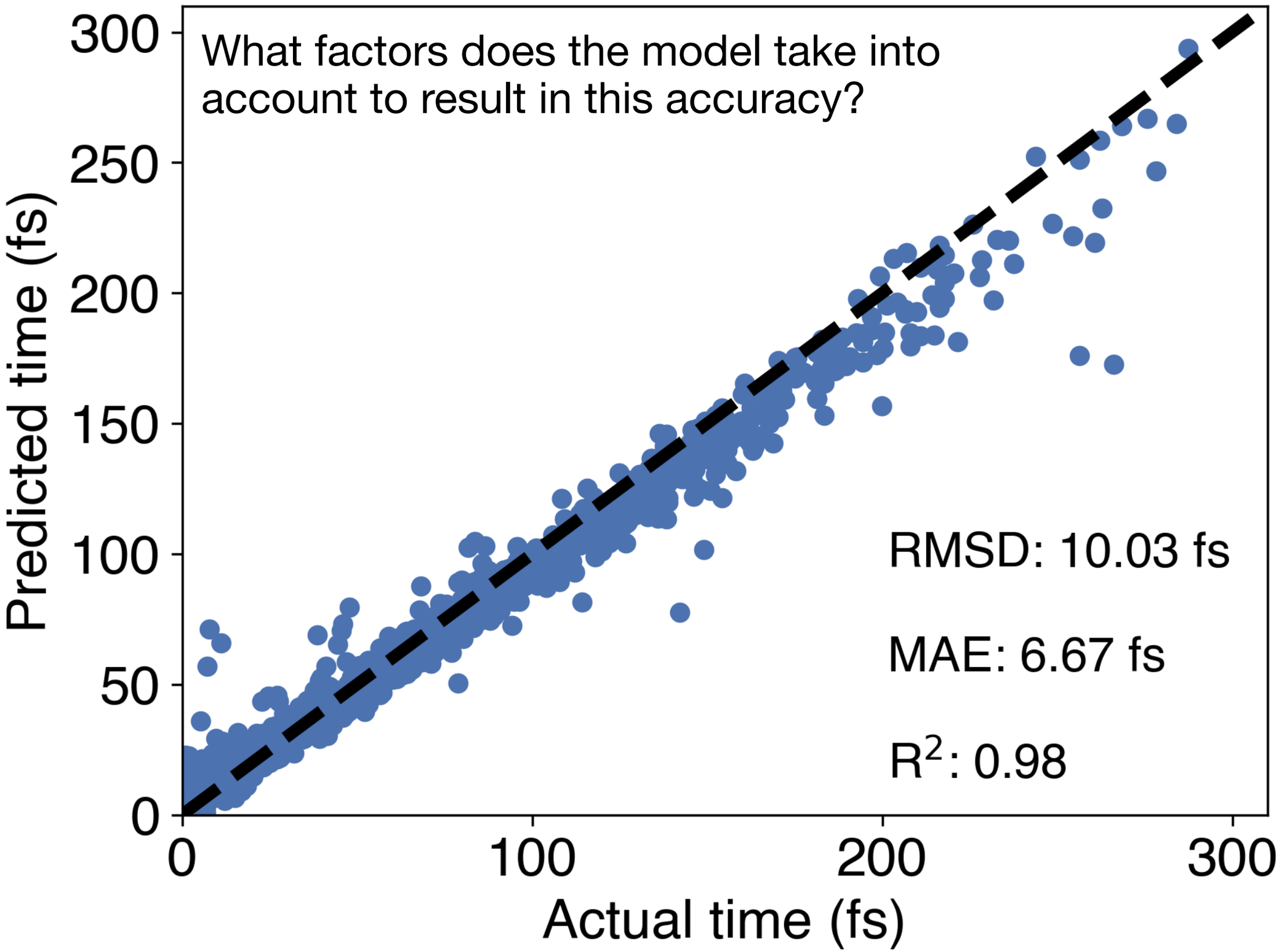
Dropout randomly turns off neurons during training, forcing the model to learn alternate representations.





Sweet spot is generally where validation error is *slightly* higher than training error

Out-of-sample predictions



Accuracy versus explainability

Simple models are easy to explain, like linear models

$$E = \frac{p^2}{2m} + \frac{1}{2}kq^2$$

Why this harmonic oscillator has an energy E is a simple question.

Other complicated mathematical models are not so obvious to explain.

$$\hat{H}_N e^{\hat{T}} |0^e 0^p\rangle = E^{\text{corr}}_{\text{NEO-CCSD}} e^{\hat{T}} |0^e 0^p\rangle$$

$$\hat{H}_N = \hat{H} - \langle 0^e 0^p | \hat{H} | 0^e 0^p \rangle$$

$$\begin{aligned}\hat{H} = &\left(h_q^p + \sum_i \overline{g}_{qi}^{pi} - \sum_I g_{qI}^{pI} \right) \tilde{a}_p^q + \frac{1}{4} \overline{g}_{rs}^{pq} \tilde{a}_{pq}^{rs} \\ &+ \left(h_Q^P + \sum_I \overline{g}_{QI}^{PI} - \sum_i g_{Qi}^{Pi} \right) \tilde{a}_P^Q + \frac{1}{4} \overline{g}_{RS}^{PQ} \tilde{a}_{PQ}^{RS} - g_{qQ}^{pP} \tilde{a}_{pP}^{qQ} \\ &+ \sum_i h_i^i + \frac{1}{2} \sum_{ij} \overline{g}_{ij}^{ij} + \sum_I h_I^I + \frac{1}{2} \sum_{IJ} \overline{g}_{IJ}^{IJ} - \sum_{iI} g_{iI}^{iI}\end{aligned}$$

Why does this model predict a correlation energy of $E^{\text{corr}}_{\text{NEO-CCSD}}$?

Accuracy versus explainability

Generally, the more complex the equation, the less human-interpretable it becomes (black box).

This is usually fine if you want an accurate result, but often you want (or need) to know why a particular result was obtained.

Accuracy versus explainability

Generally, the more complex the equation, the less human-interpretable it becomes (black box).

This is usually fine if you want an accurate result, but often you want (or need) to know why a particular result was obtained.

Machine learning models are no exception.

For example:

- A bank uses a ML model to evaluate you for a loan. You were denied. Why? Legally, you have a right to know.
- A ML model predicts you are at risk for heart disease. Why? What factors can you adjust to change your lot?

Accuracy versus explainability

Generally, the more complex the equation, the less human-interpretable it becomes (black box).

This is usually fine if you want an accurate result, but often you want (or need) to know why a particular result was obtained.

Machine learning models are no exception.

For example:

- A bank uses a ML model to evaluate you for a loan. You were denied. Why? Legally, you have a right to know.
- A ML model predicts you are at risk for heart disease. Why? What factors can you adjust to change your lot?

Because of this importance, several techniques have been devised to extract meaning from any machine-learned model.

Permutation importance

What features are most important?

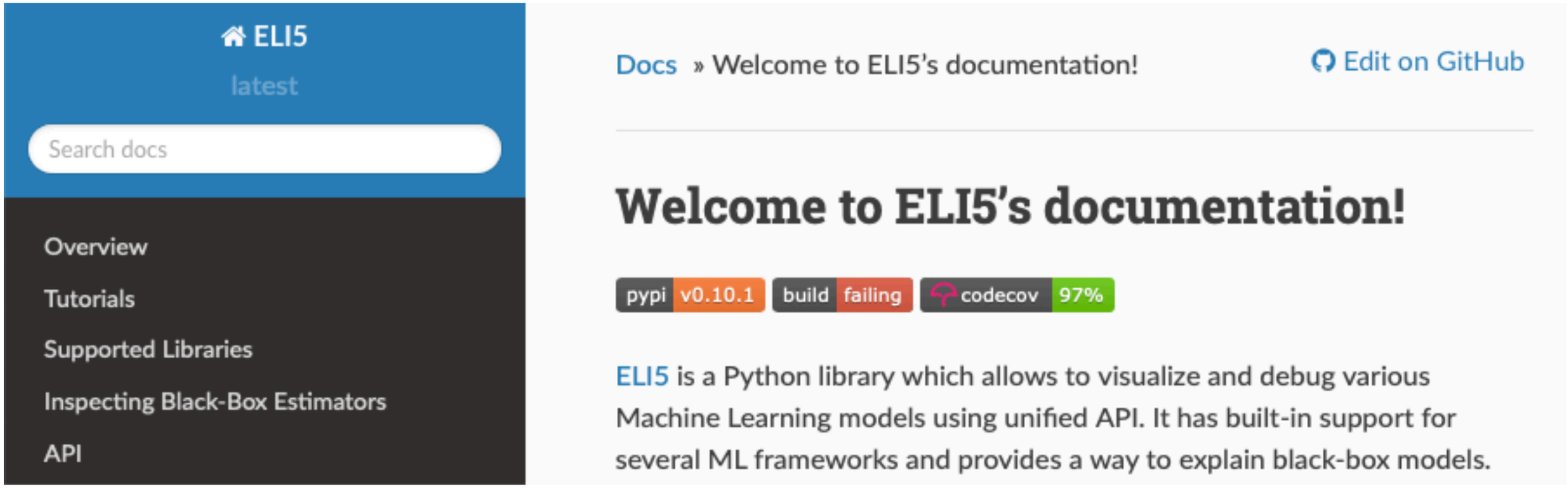
1. Get a trained model.
2. Shuffle the values in a single column and compute predictions.
3. Use these predictions and the true target values to calculate how much error was introduced by shuffling. This error measures the importance of the shuffled feature.
4. Return the data to the original order (undo step 2). Rinse and repeat for each column.

Time to PT	q1	q2	…	q144
10	-0.1	0.1	…	0.2
8	0.2	-0.3	…	0.1
…	…	…	…	…
0	0.5	-0.1	…	-0.2

Permutation importance in 4 lines

Once you have a trained `model`:

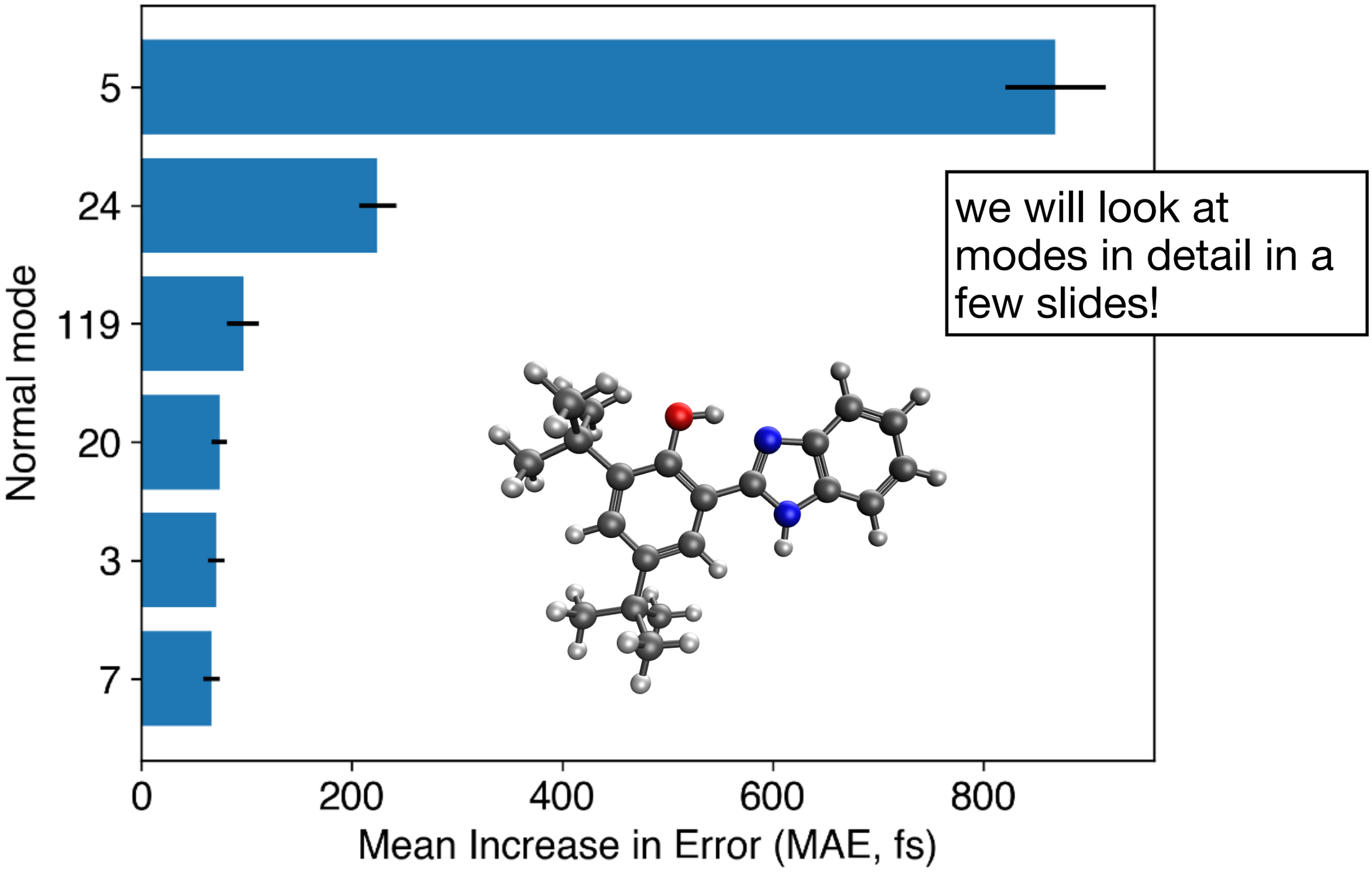
```
import eli5
from eli5.sklearn import PermutationImportance

perm = PermutationImportance(model,random_state=200,n_iter=200).fit(x_test,y_test)
explanation = eli5.explain_weights(perm, feature_names = x_test.columns.tolist())
```



https://eli5.readthedocs.io/en/latest/index.html

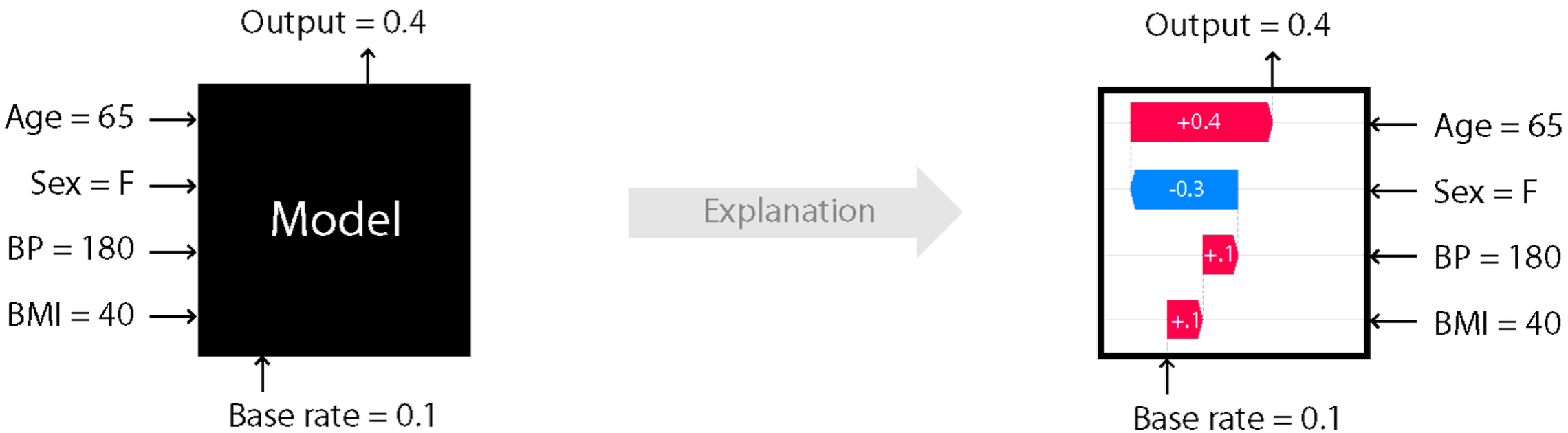
Permutation importance

What features does the model most rely on?



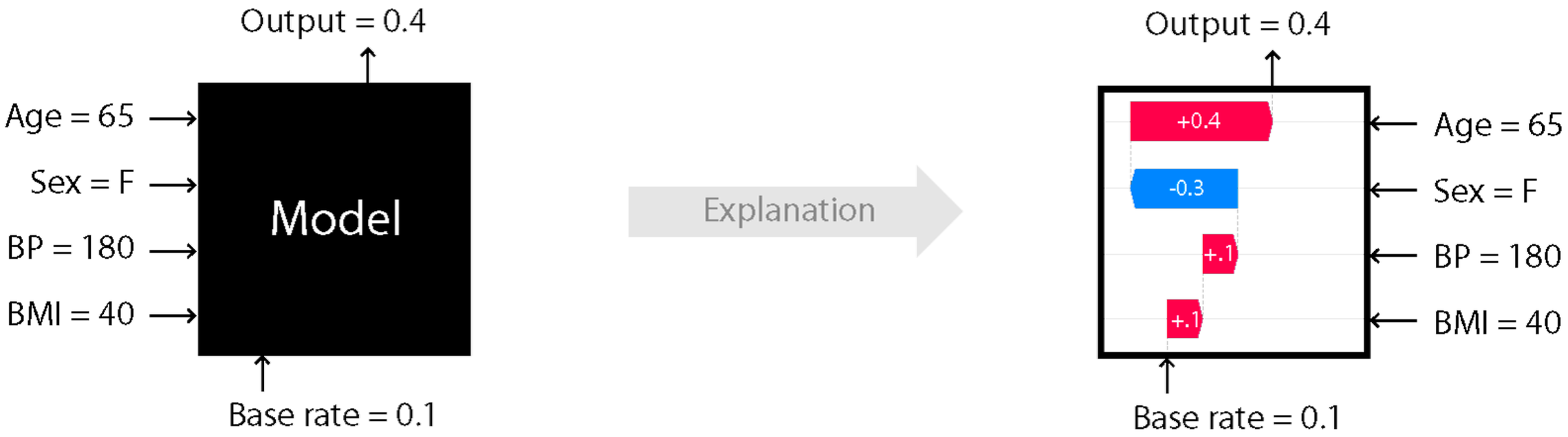
SHAP values (SHapley Additive exPlanations)

- SHAP (SHapley Additive exPlanations) is a game theoretic approach to explain the output of any machine learning model.
- Based on Shapley values (1951), which are a solution concept from game theory. SHAP values tell you which input feature contributed the most to a prediction. (**Nobel prize in economics 2012!**)
- SHAP values assign each input feature a value, that, when added together, results in the predicted output of that model.



Lundberg, S. M.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. ANIPS *30* (2017).

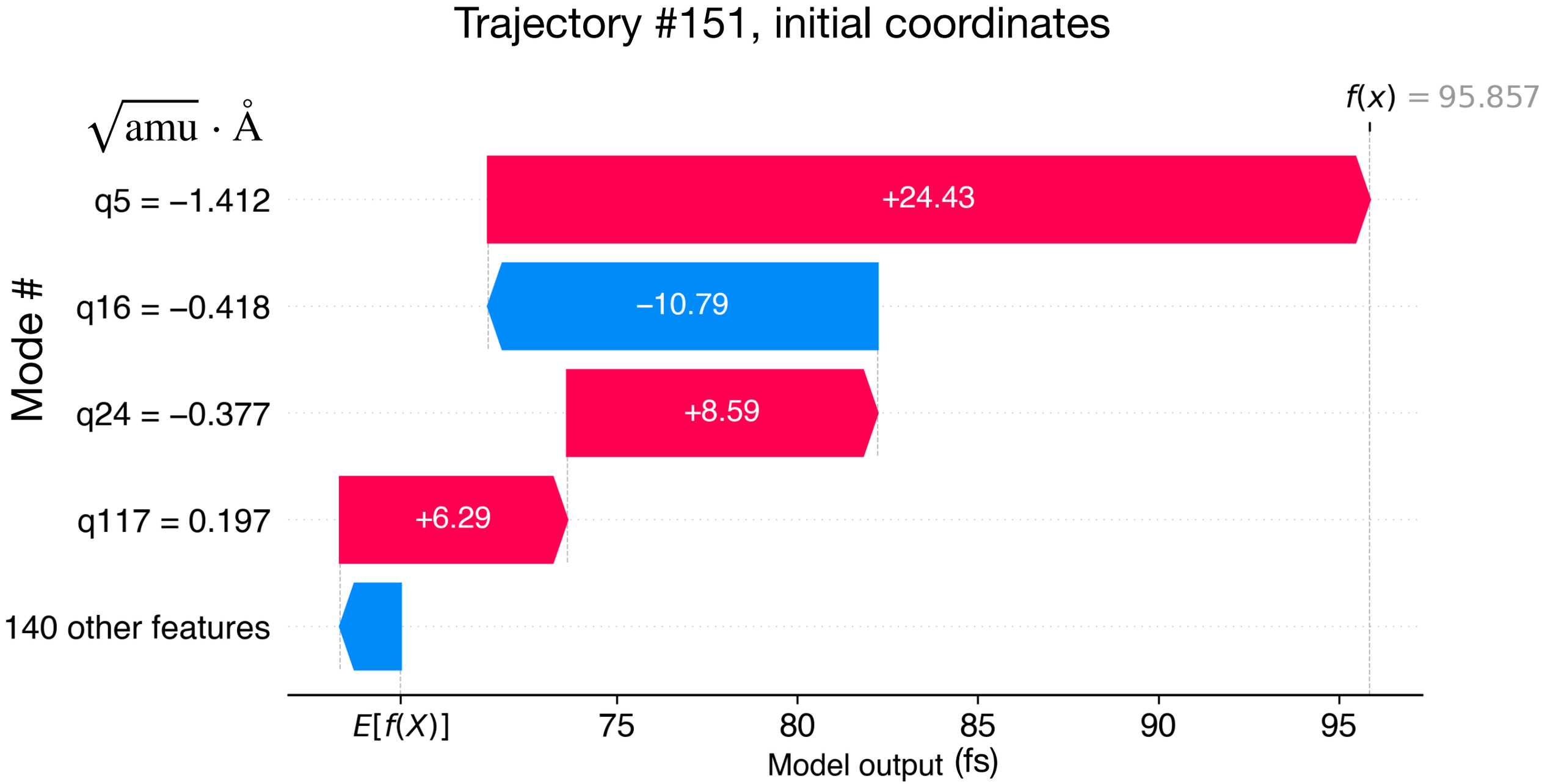
SHAP values (SHapley Additive exPlanations)



- SHAP values are *local additive*:
  - **Local**: SHAP values are assigned to a single prediction/instance
  - **Additive**: sum of features' SHAP values sums to total prediction
- In principle, takes exponential amount of time to assign SHAP values
  - Recent algorithmic improvements (here at **UW**!) overcome these issues

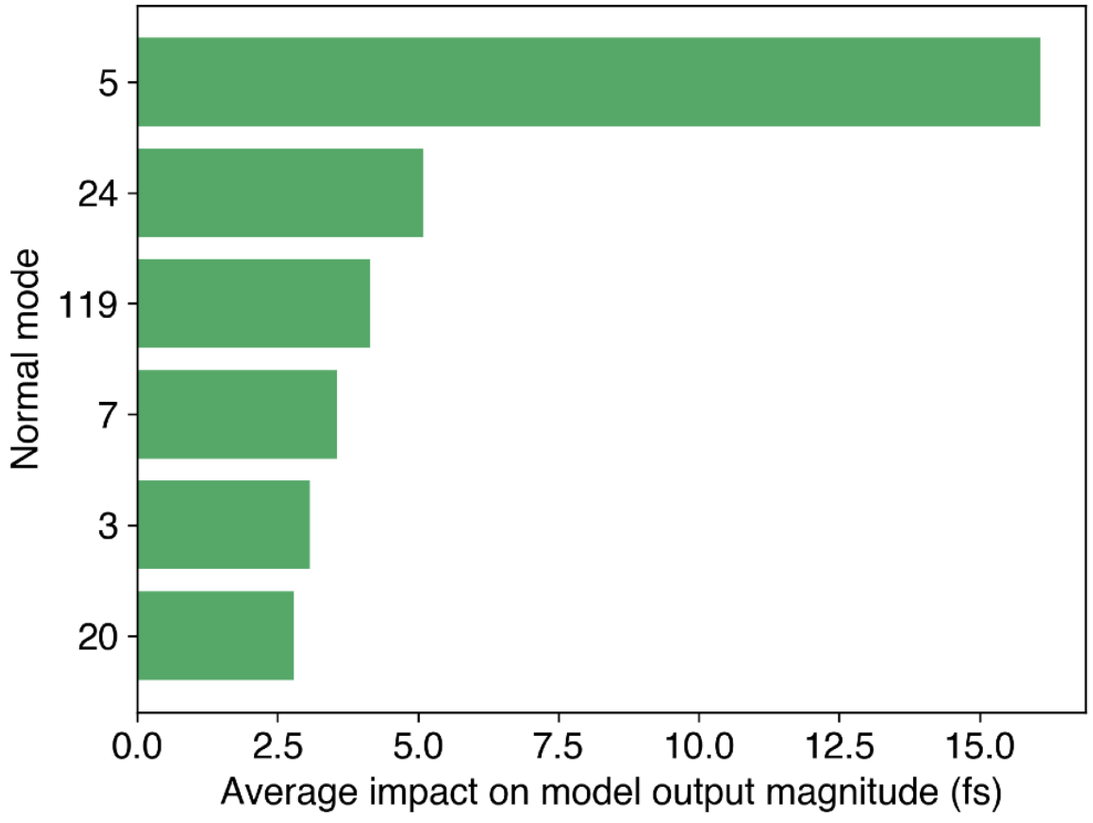
Lundberg, S. M.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. ANIPS *30* (2017).

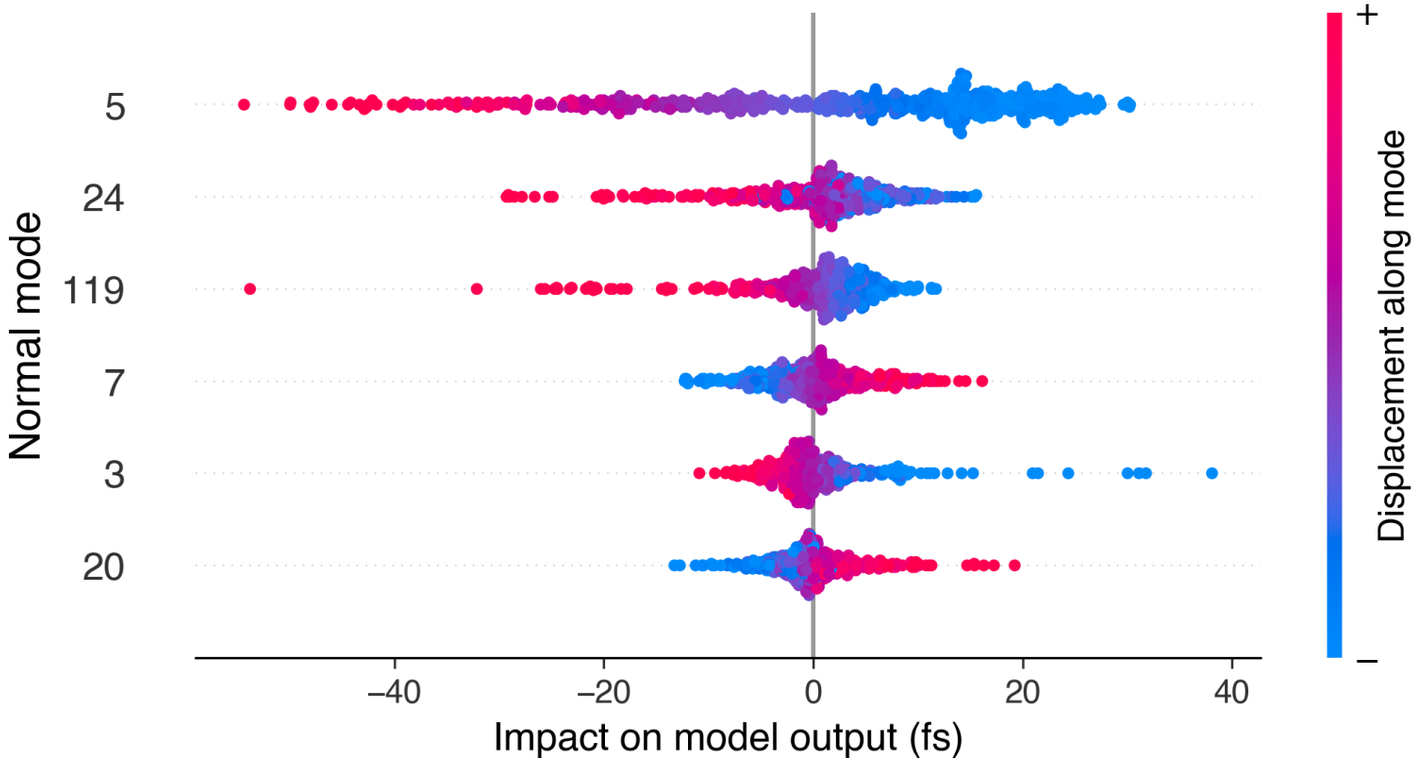
Individual predictions



we will look at modes in detail in a few slides!

Summary of SHAP values





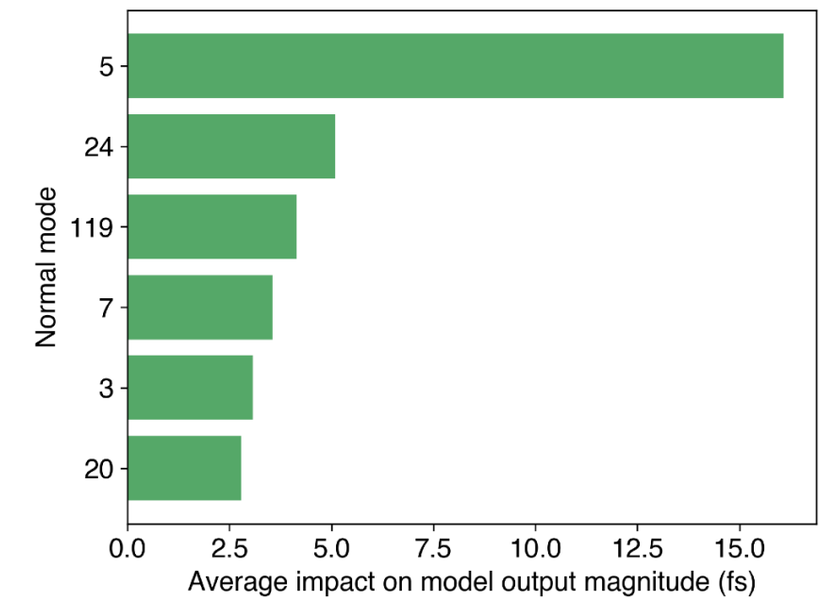
Taken for random sample of 500 data points, we see modes the model deems “important.”

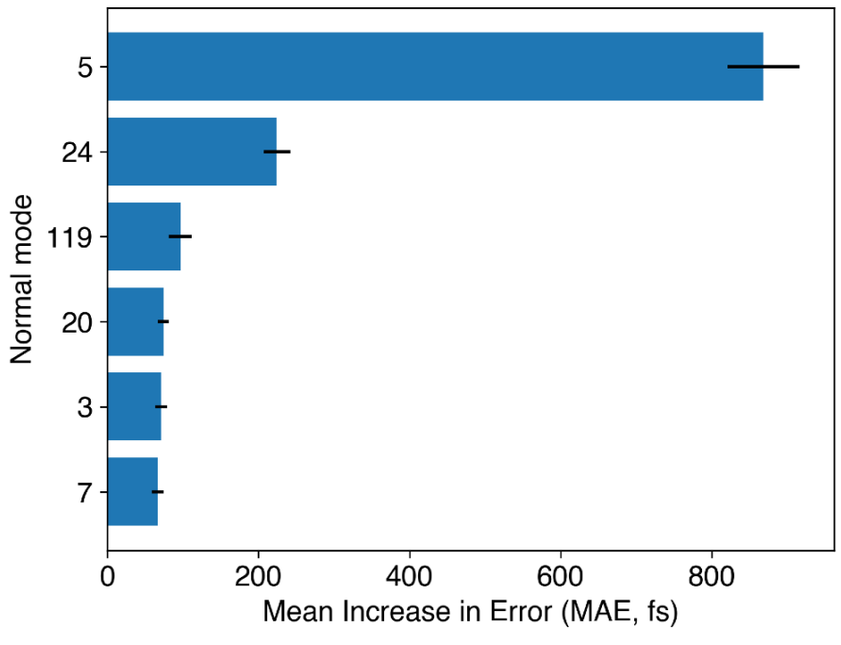
Consistent with permutation testing, mode #5 has a high degree of significance, followed by modes #119 and #24.

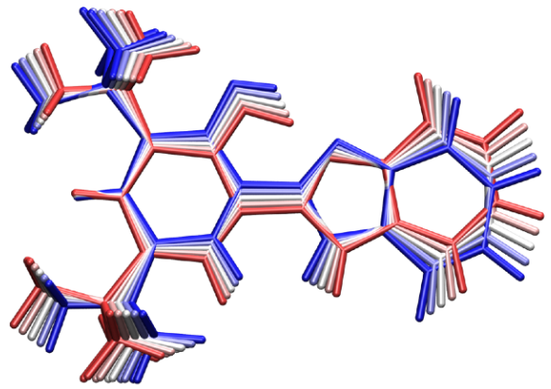
We will look at these three modes more closely, and see what insights can be found.

Modes of oxidized BIP

Mode #5 most significant, followed by modes #24 and #119



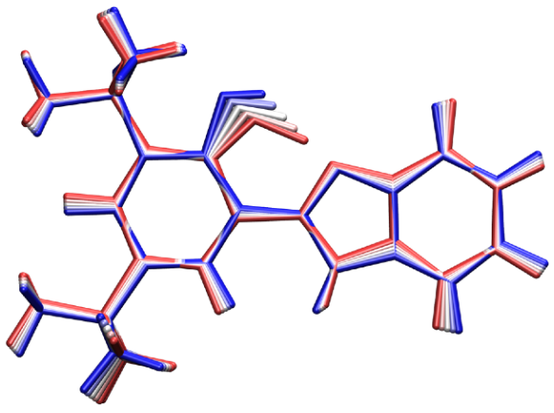




Mode 5

72 cm^{-1}

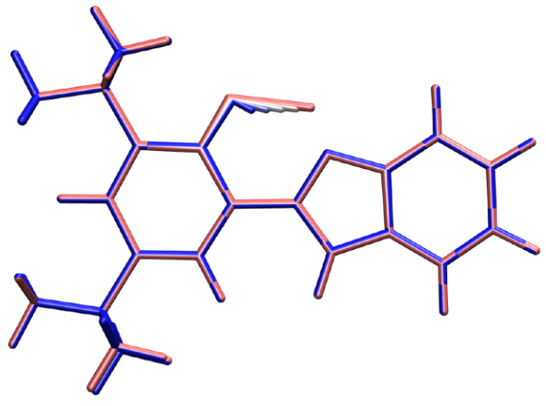
phenol-benzimidazole bend



Mode 24

356 cm^{-1}

donor-acceptor mode



Mode 119

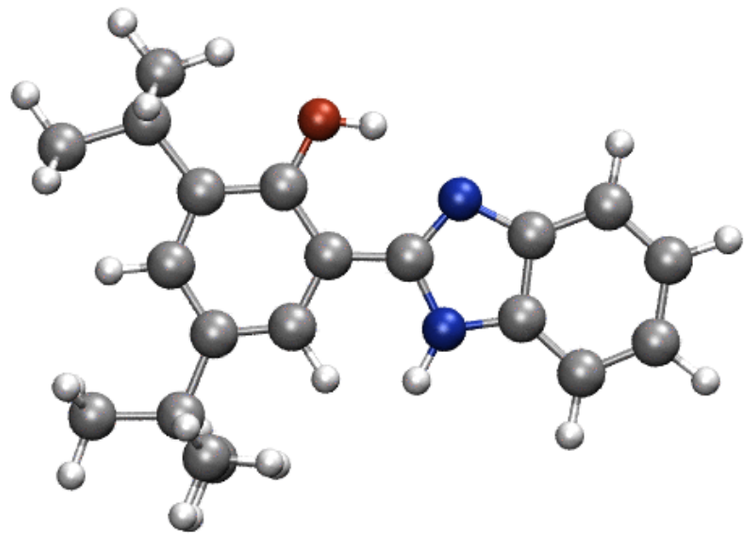
2420 cm^{-1}

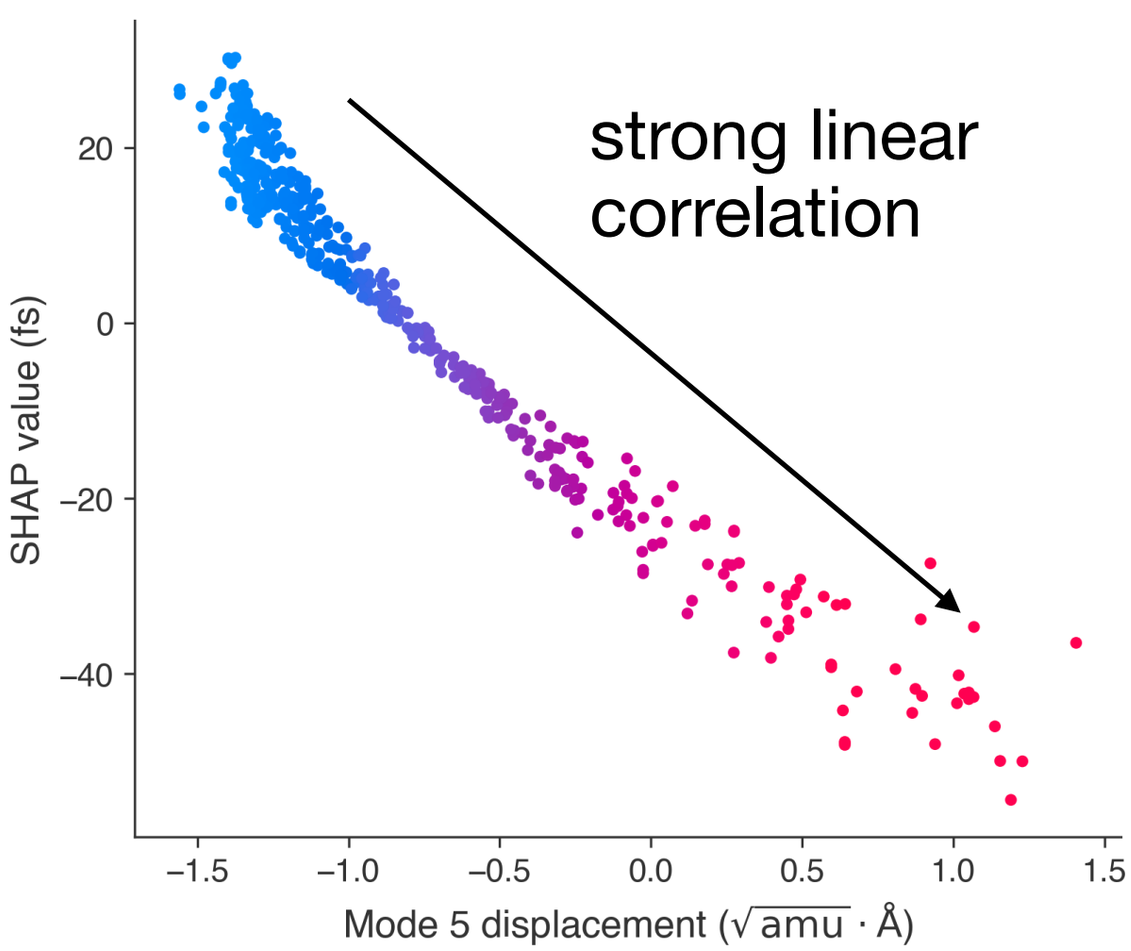
proton stretch

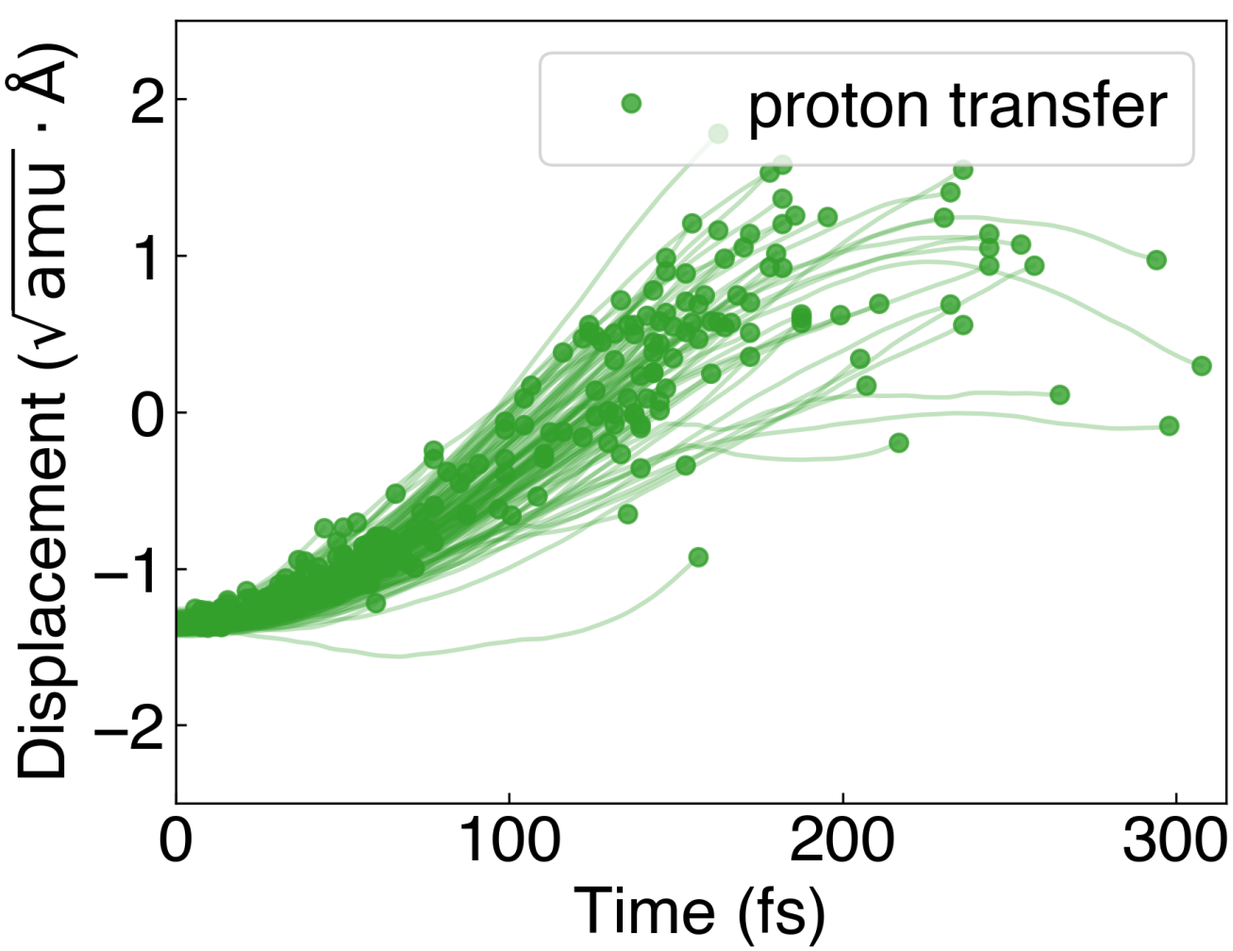
Mode #5: phenol-benzimidazole bend

Trajectories start strongly displaced along this mode (relatively speaking)

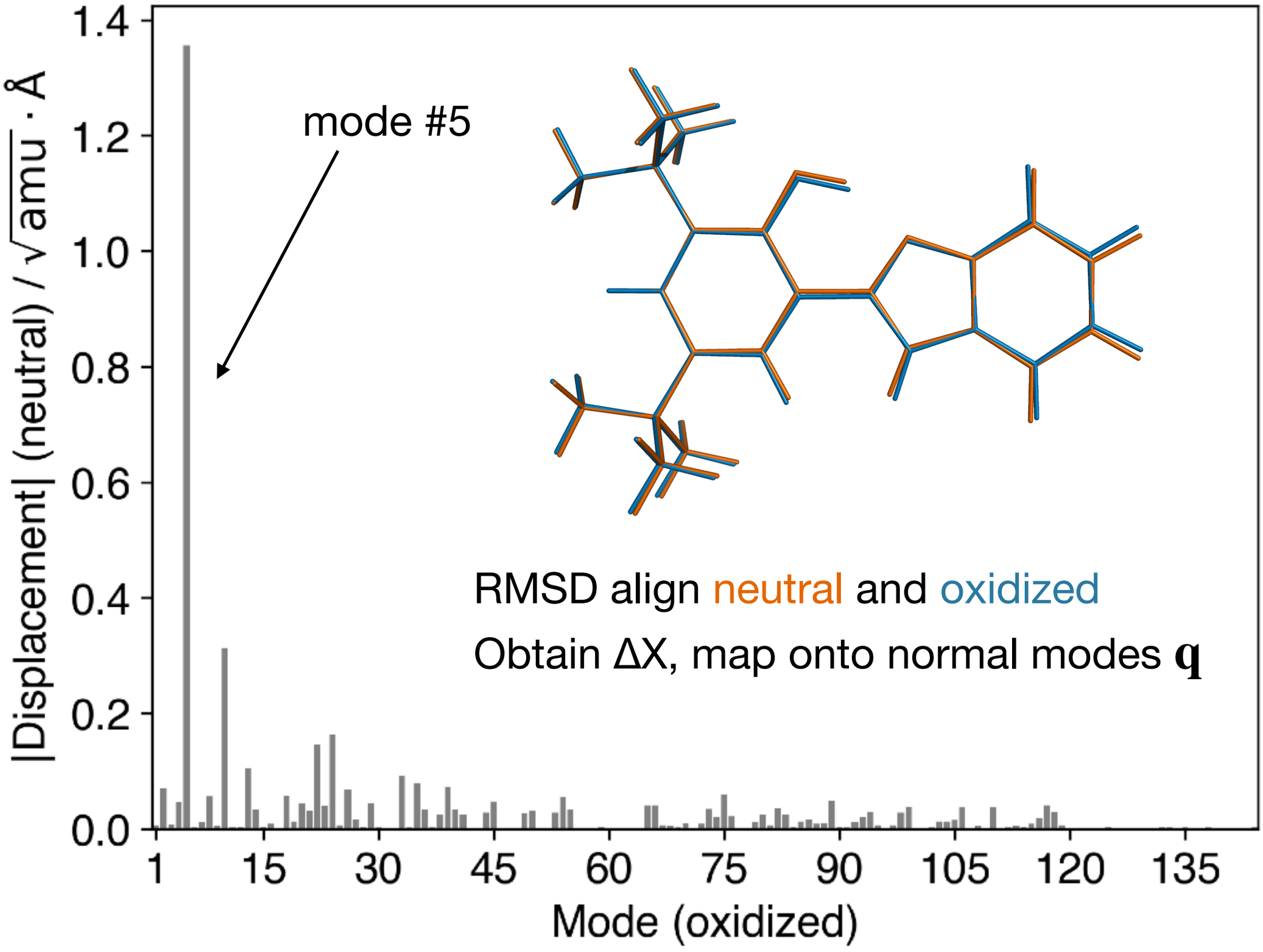
As reaction progresses, molecule reorganizes — model identifies inner sphere reorganization mode



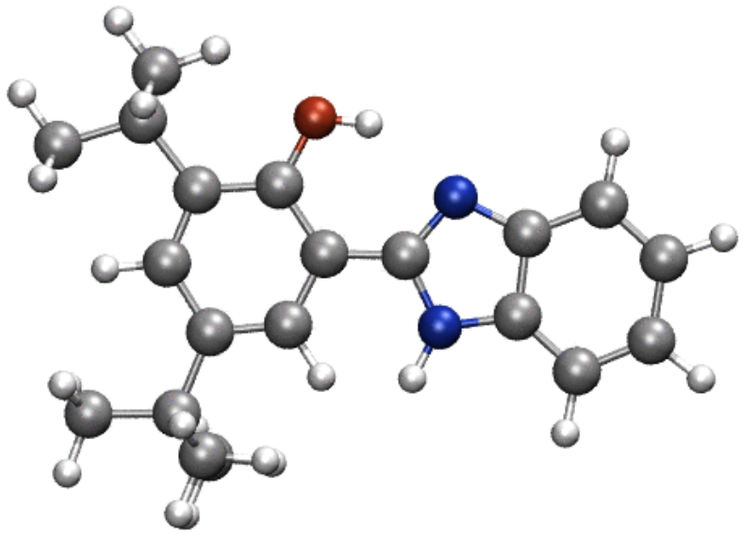


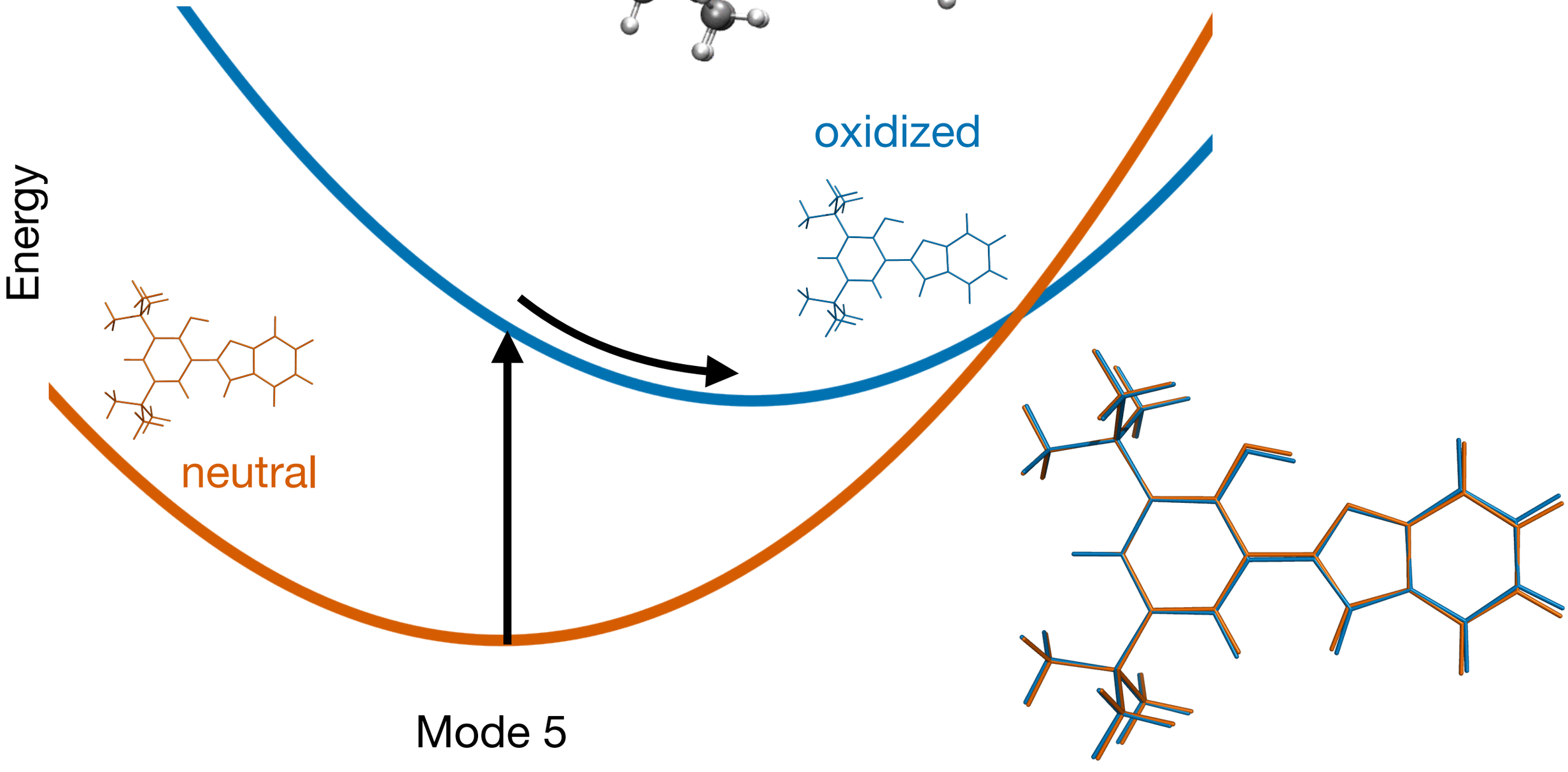


Mode #5 dominates inner sphere reorganization



Inner sphere reorganization

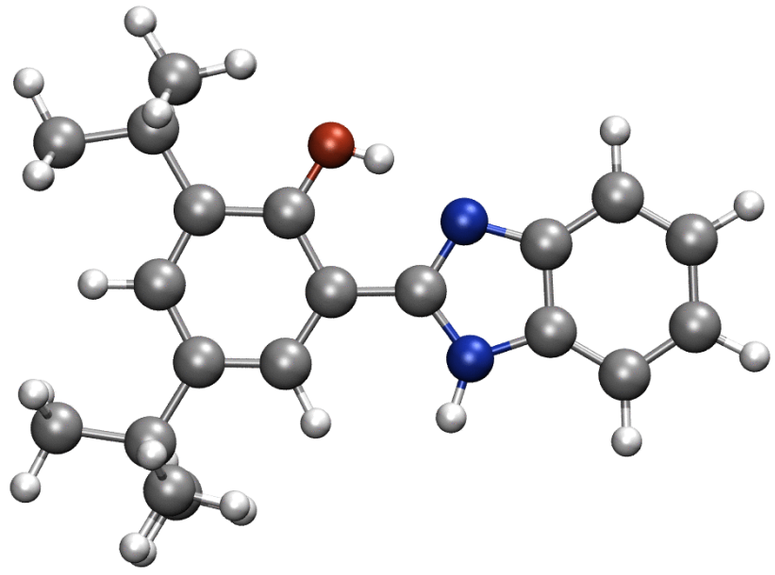


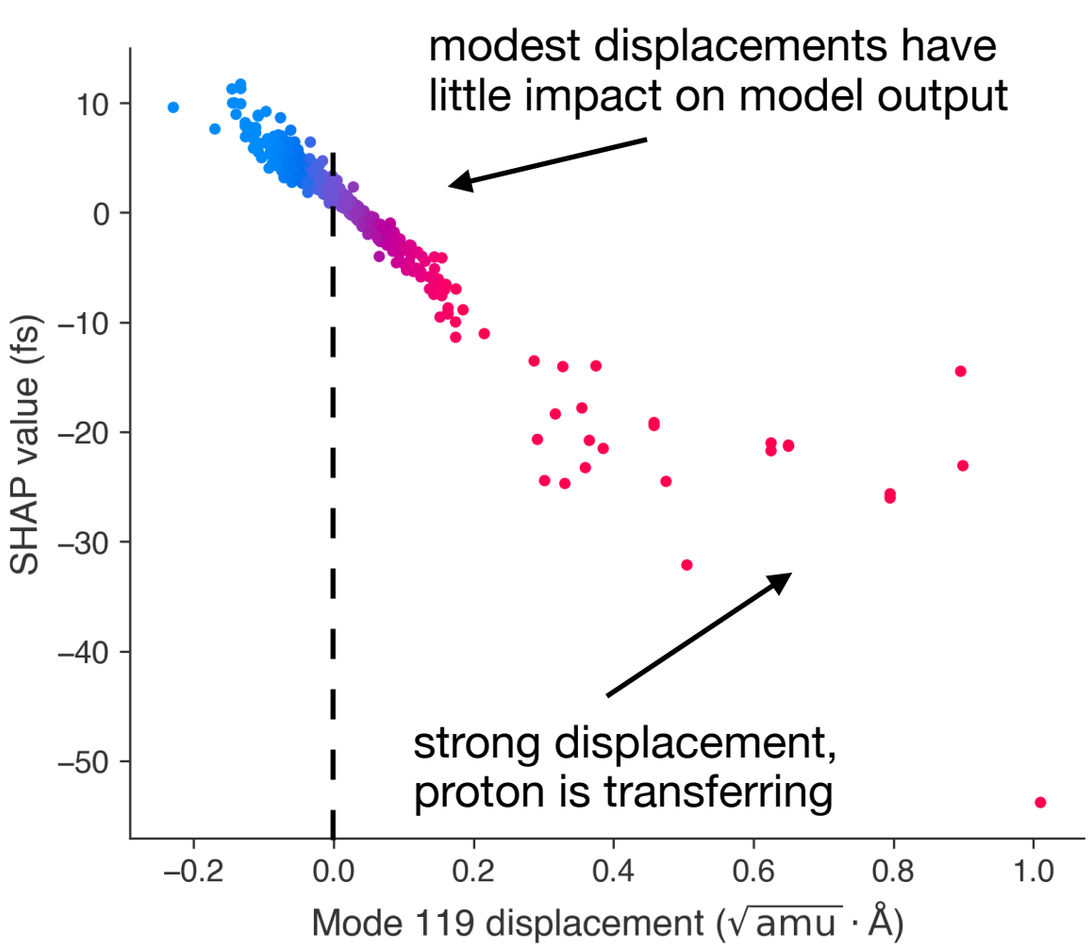


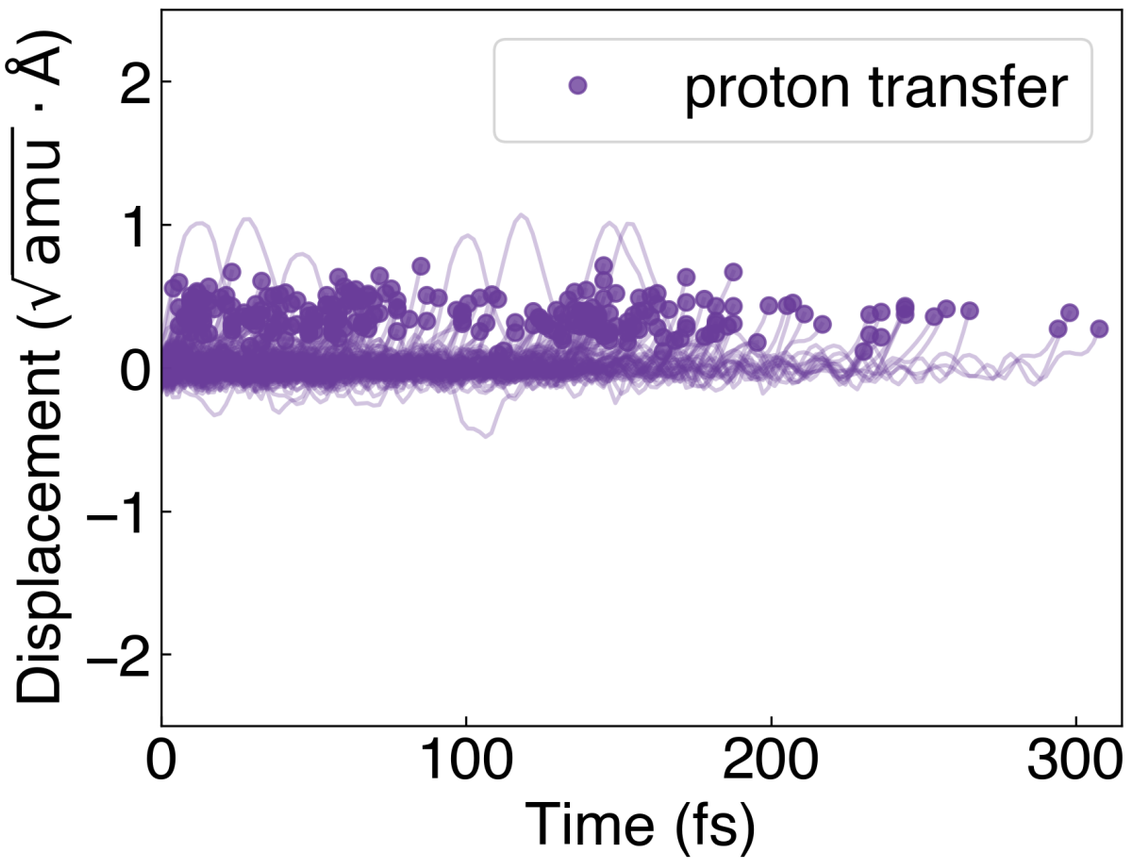
Mode 119: proton stretch

Model identifies proton stretch as significant

Not too surprising, and from the SHAP values we see that it becomes important as the proton is transferring, but not before.



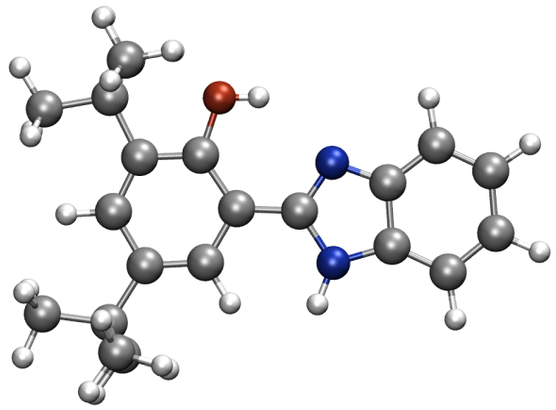




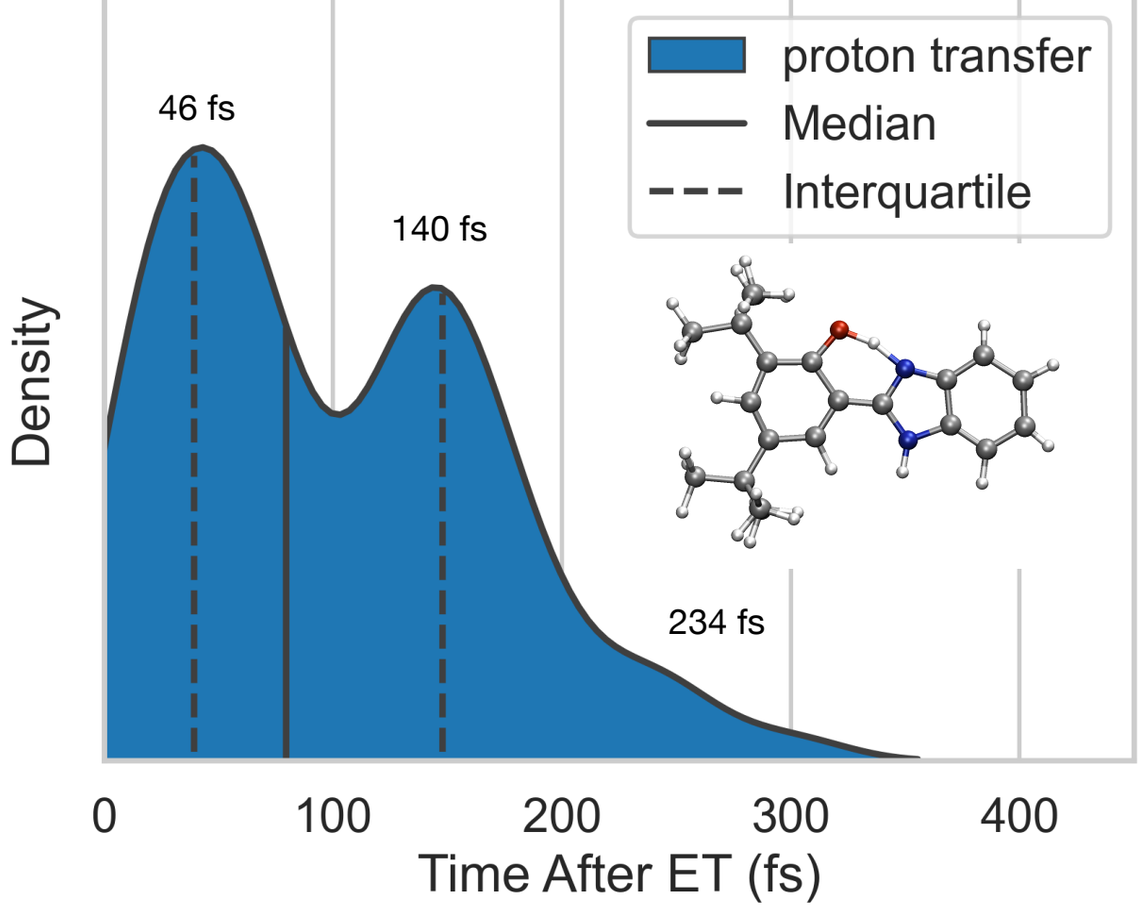
Mode 24: donor-acceptor vibration

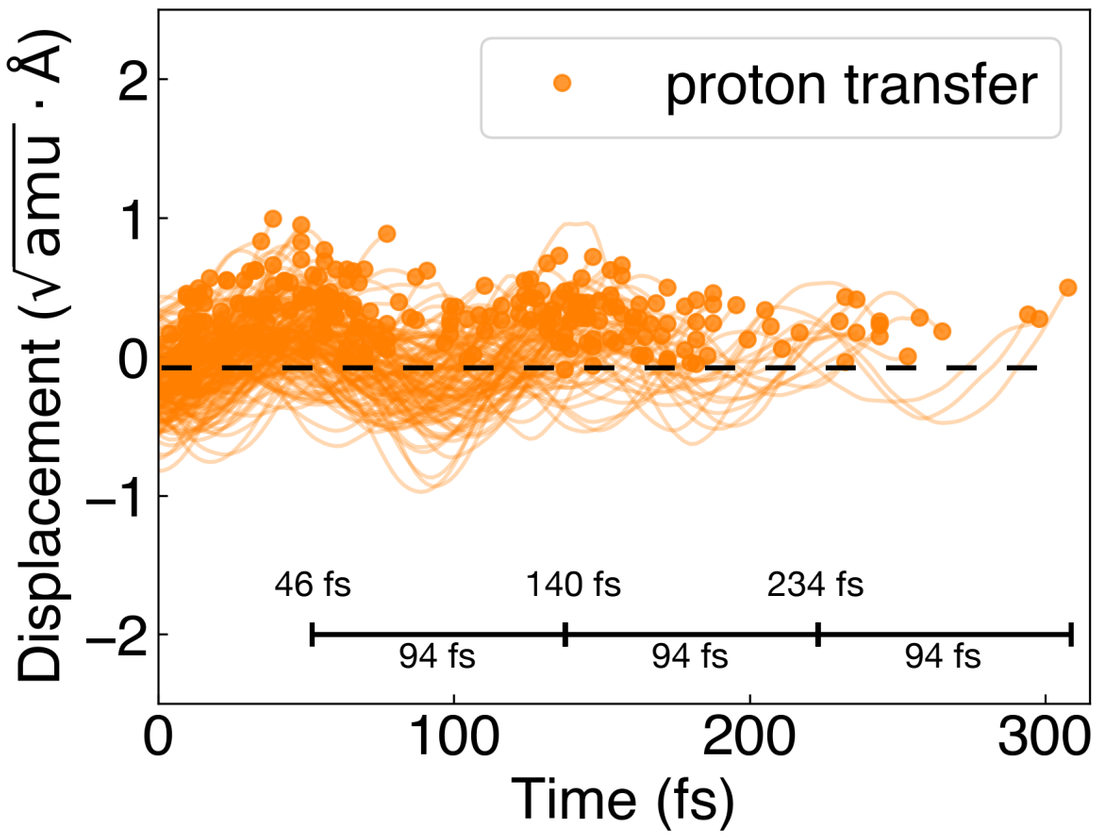
Vibrational coherence along donor-acceptor mode

Oxidation displaces
ensemble along mode 24

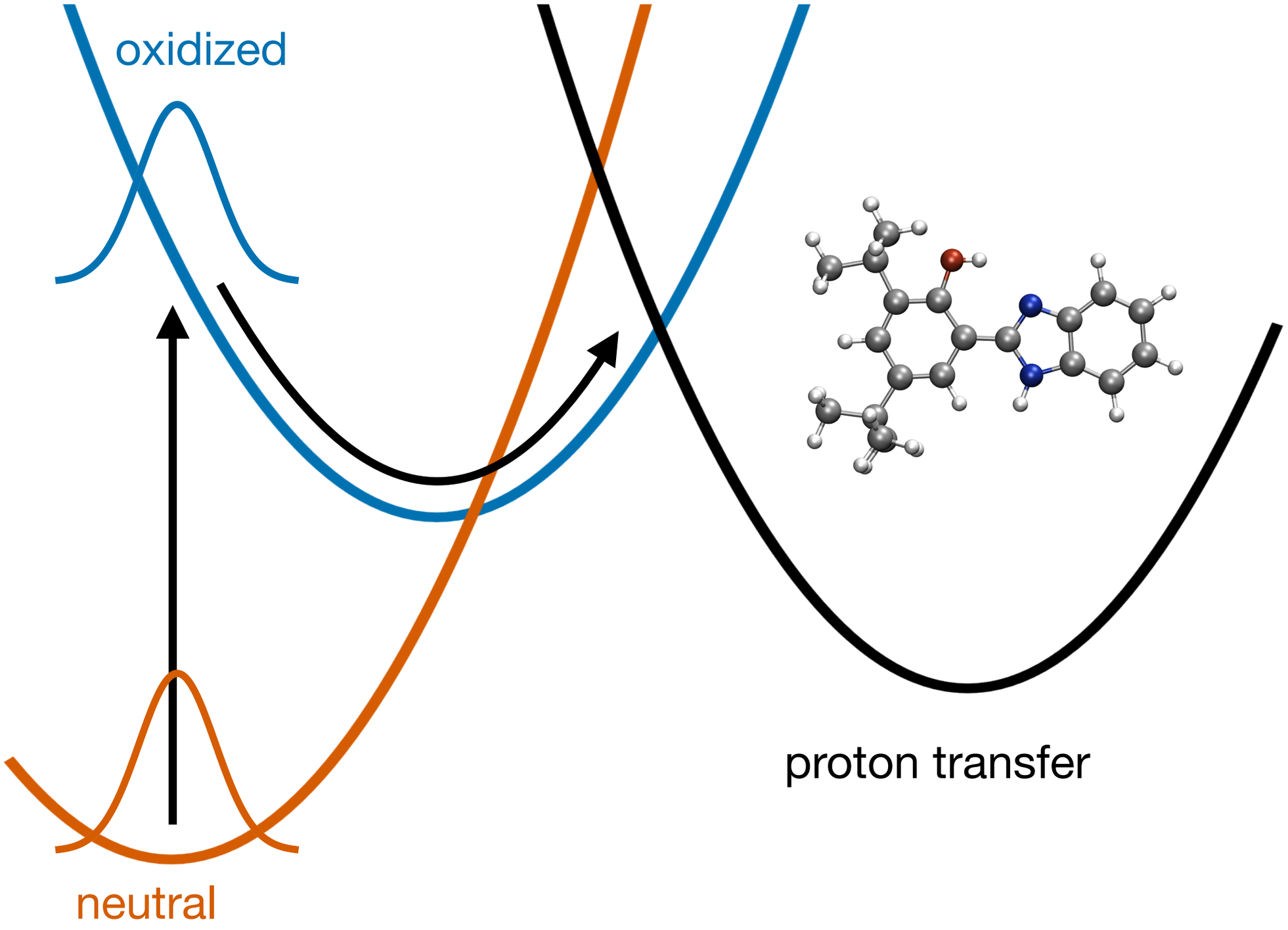


Mode #24: 356 cm^{-1} corresponding to period of ~94 fs

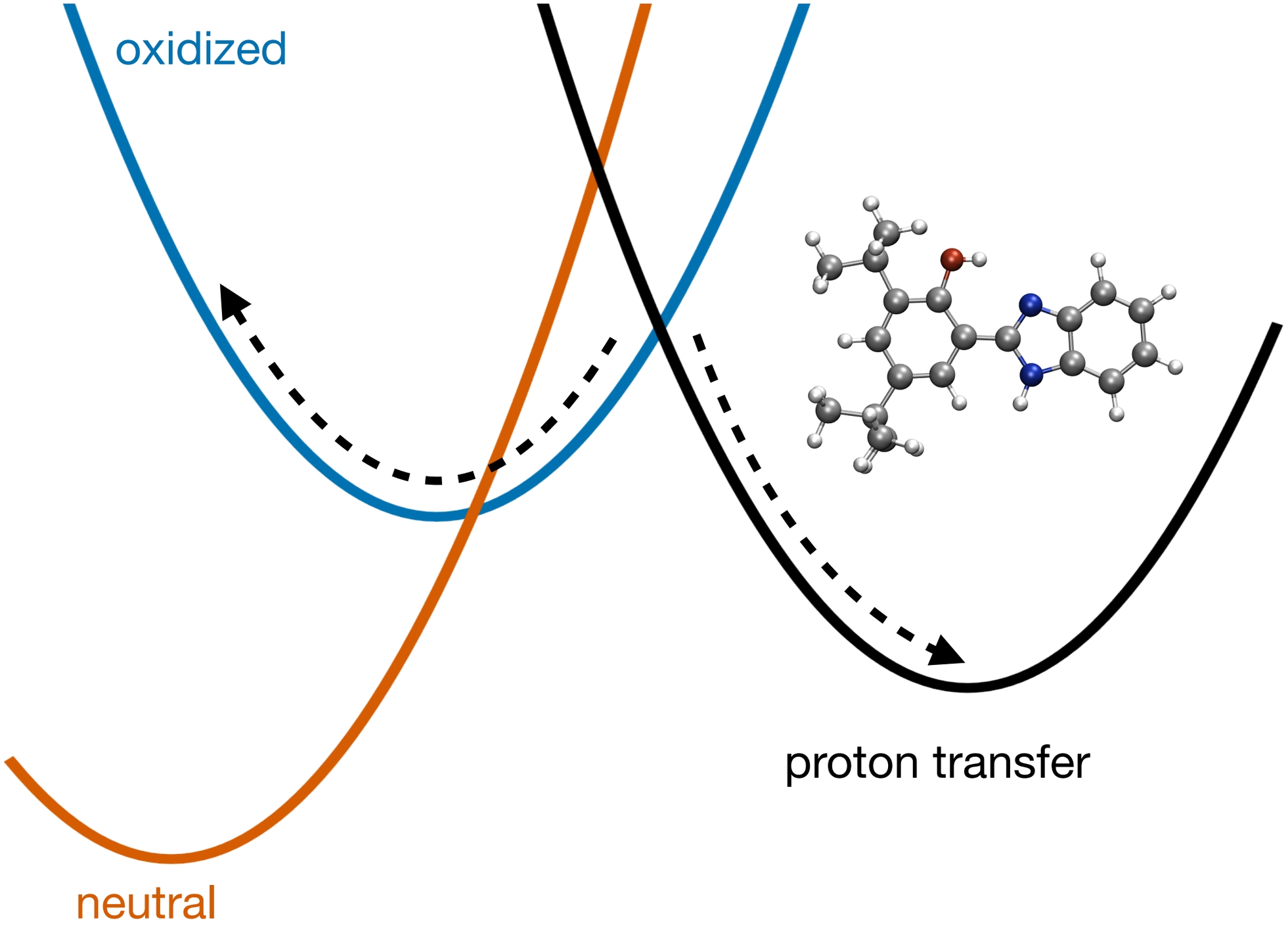




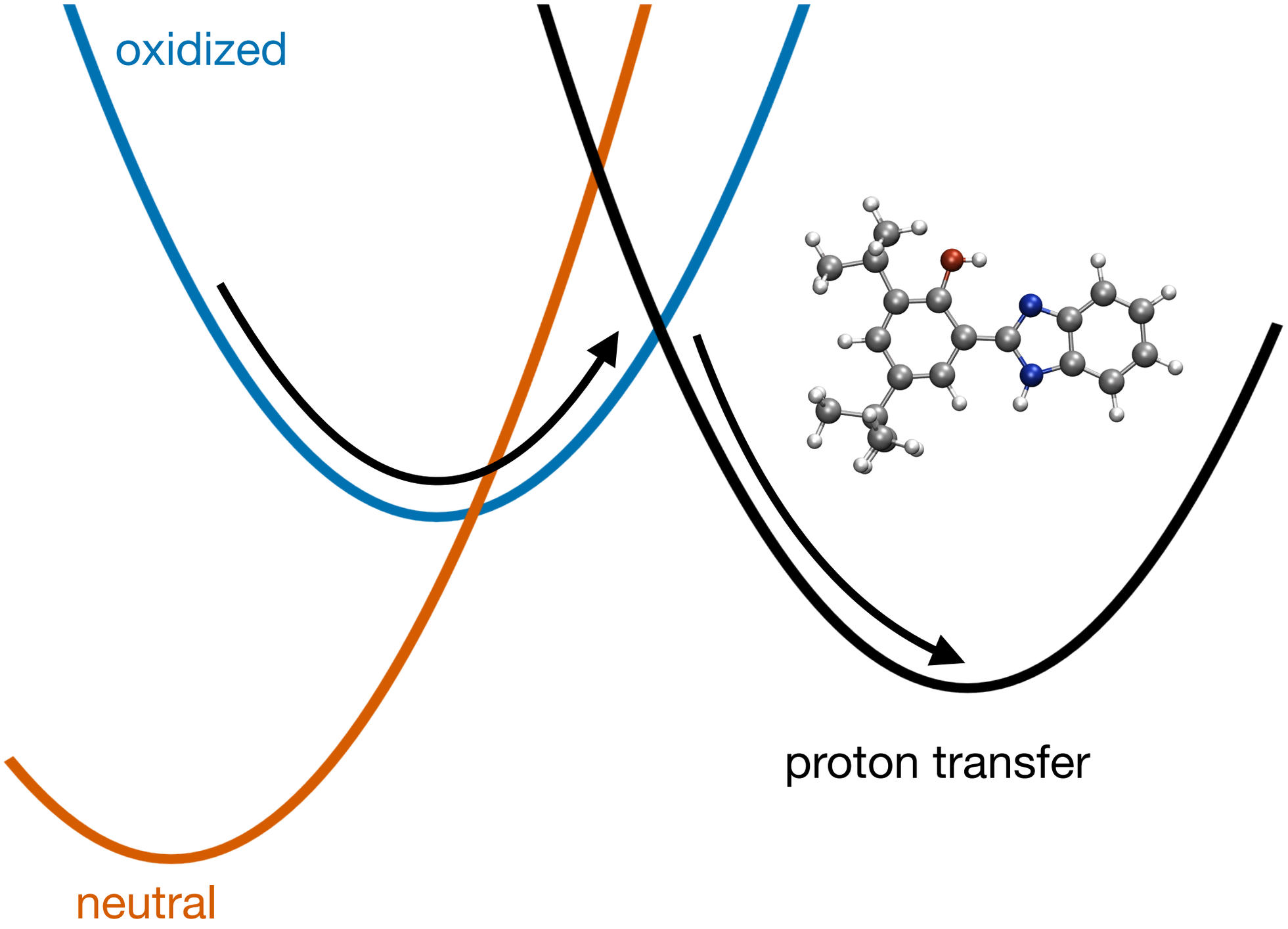
Motion along donor-acceptor mode



Motion along donor-acceptor mode



Motion along donor-acceptor mode



Conclusions

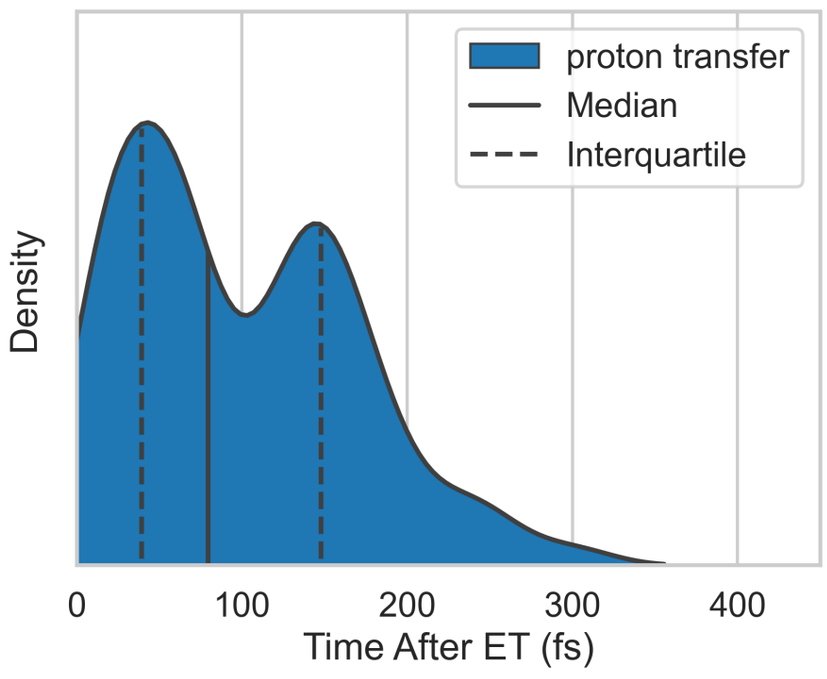
Goal: physical insight into proton transfer dynamics

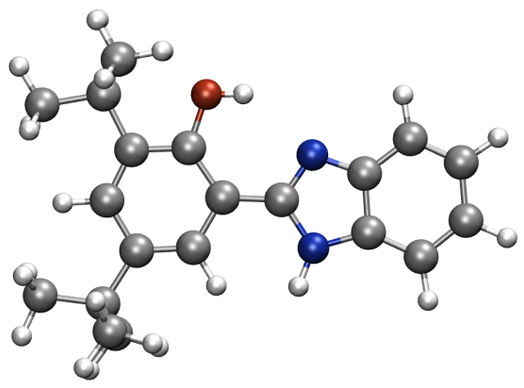
1. *ab initio* molecular dynamics to simulate PCET dynamics

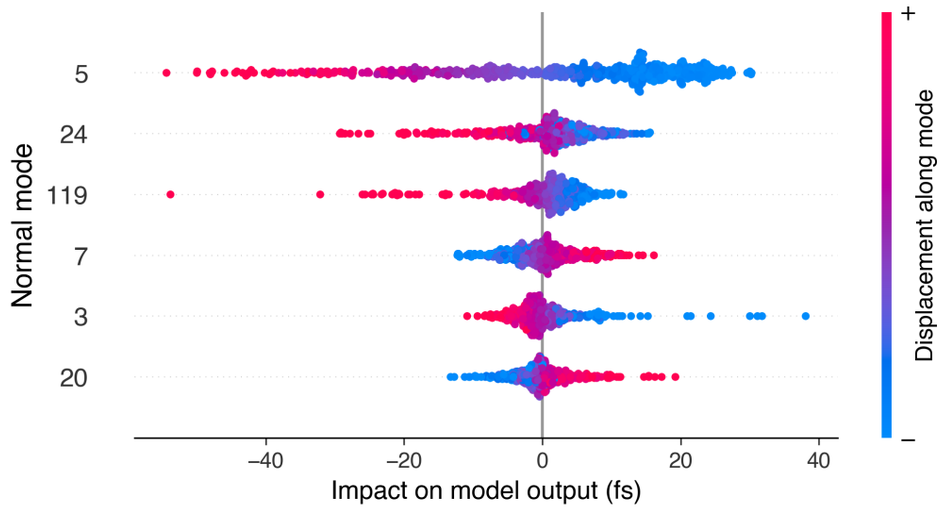
2. Use ML to distill large amount of data into meaningful insights

3. Model to tell us where to look, and then we can follow up with additional questions and investigations

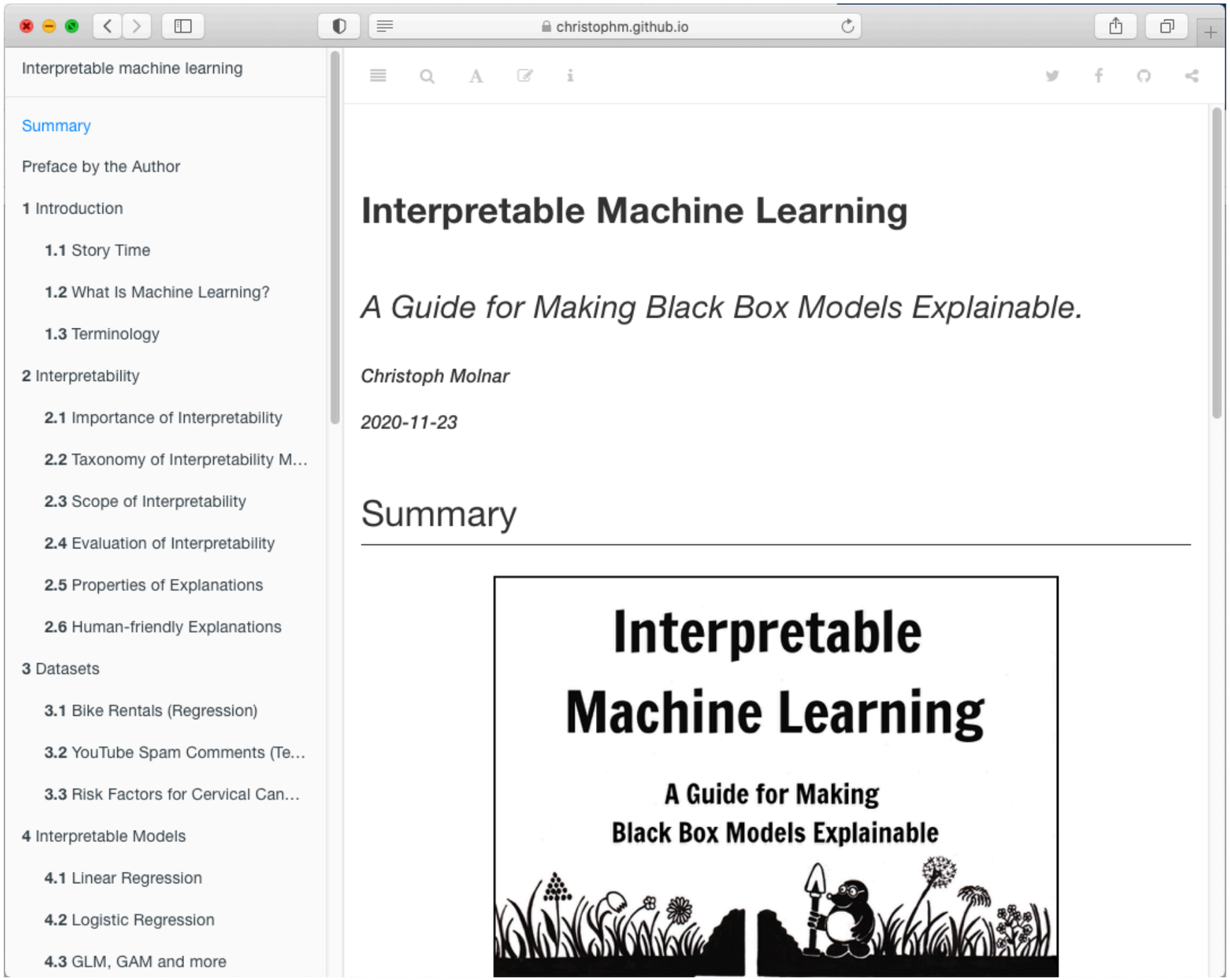
4. Identified inner-sphere reorganization modes and vibrational coherence along the donor-acceptor mode





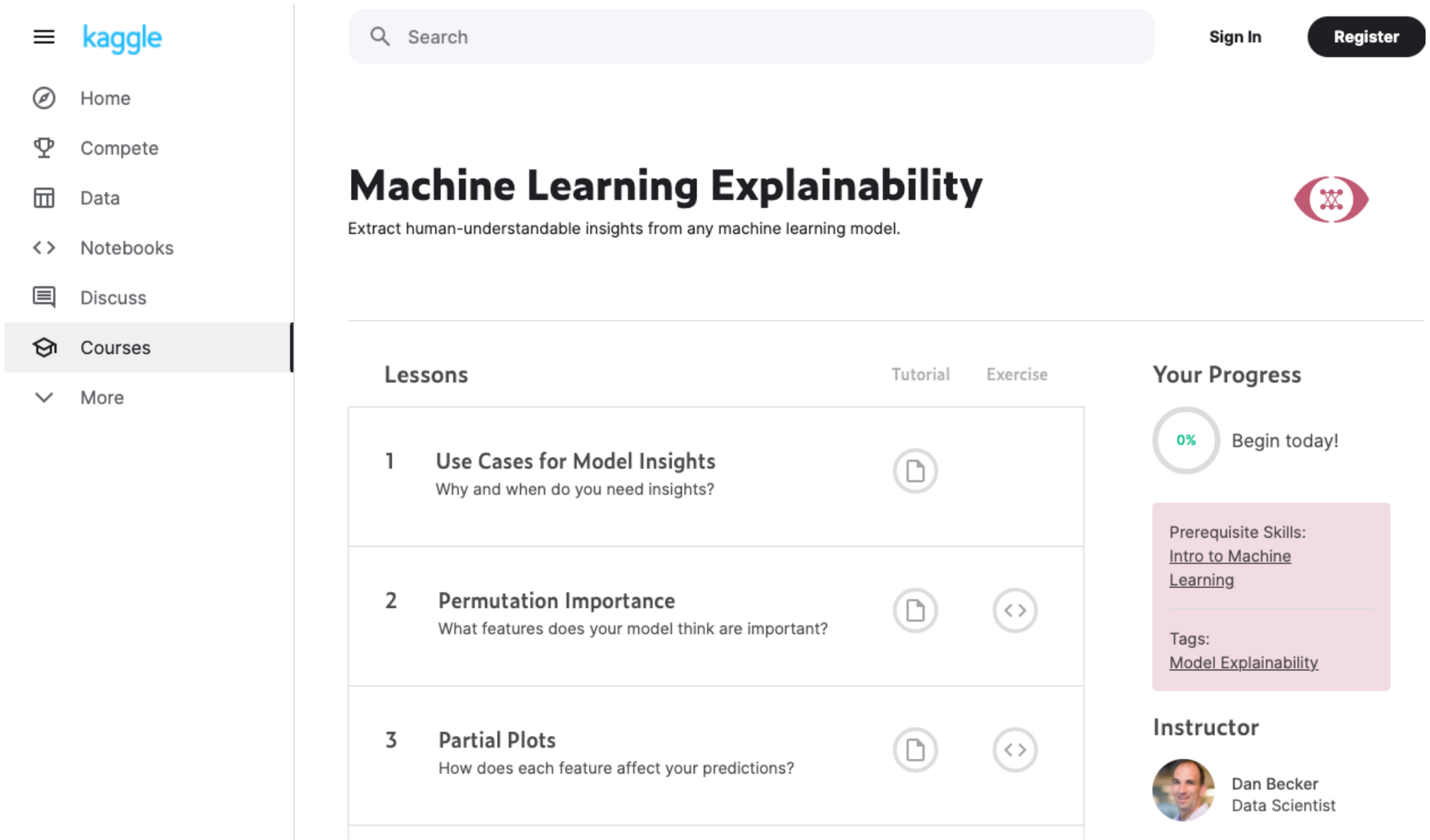


Before we go



https://christophm.github.io/interpretable-ml-book/

Before we go



https://www.kaggle.com/learn/machine-learning-explainability

In case you want to read more…

ACS central science

http://pubs.acs.org/journal/acscii?ref=pdf

http://pubs.acs.org/journal/acscii

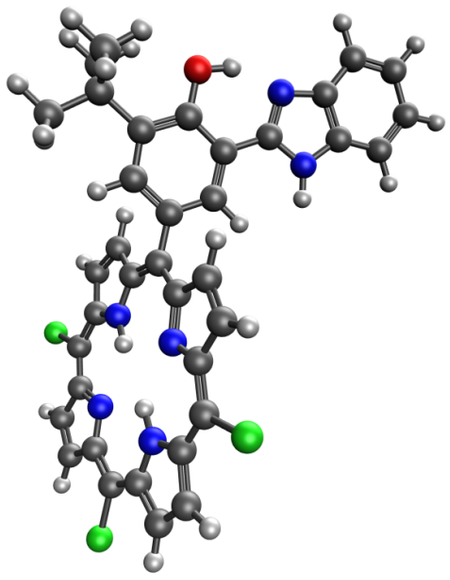
Research Article

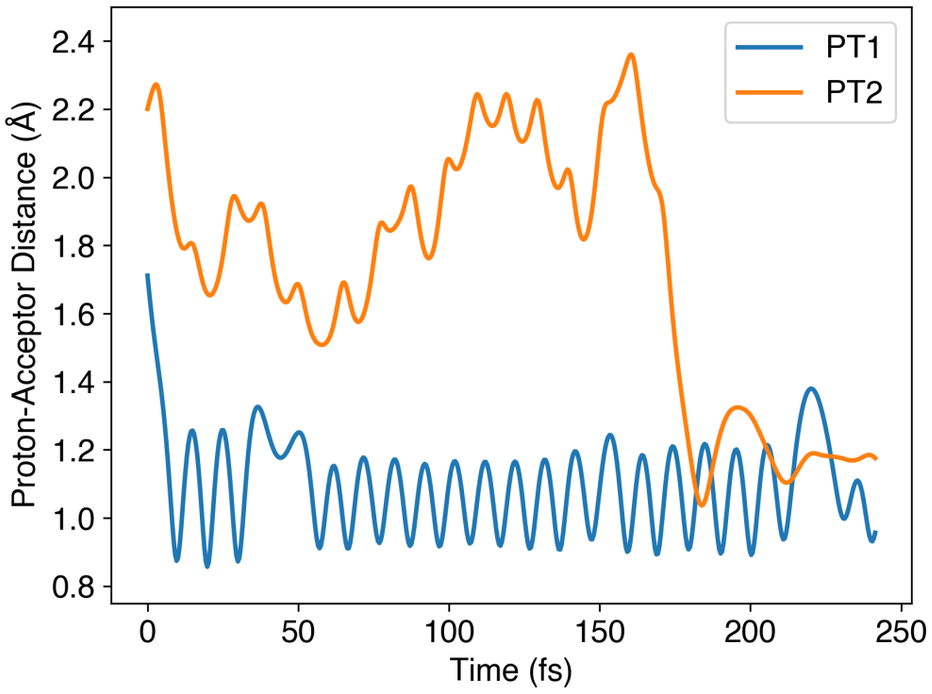
Nonequilibrium Dynamics of Proton-Coupled Electron Transfer in Proton Wires: Concerted but Asynchronous Mechanisms

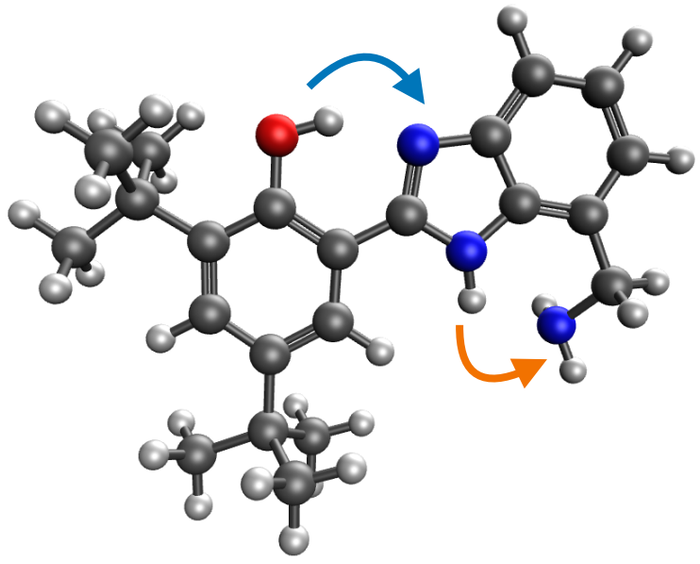
Joshua J. Goings and Sharon Hammes-Schiffer*

Cite This: https://dx.doi.org/10.1021/acscentsci.0c00756

Read Online







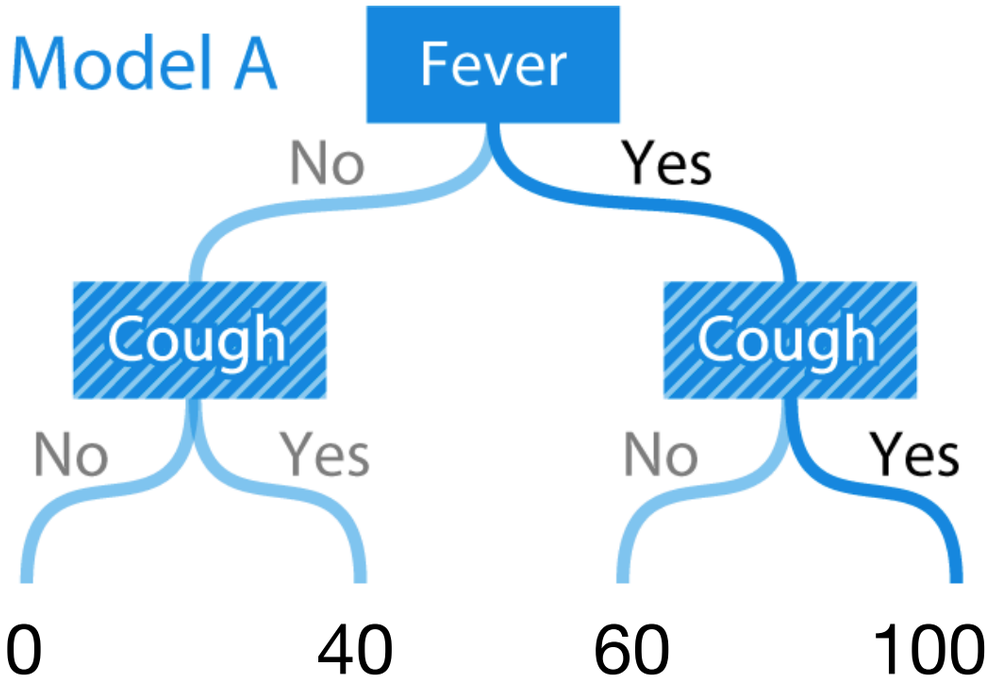
Acknowledgments

XSEDE
Extreme Science and Engineering Discovery Environment

Thank you for your attention!

SHAP values

Do I have COVID-19?



Say you have a fever and cough:

Marginal contributions:

$$\{\} = 50$$

$$\{F\} = 80$$

$$\{C\} = 70$$

$$\{F, C\} = 100$$

$$\phi_{\mathrm{F}} = \frac{1}{2}[\{F\} - \{\}] + \frac{1}{2}[\{F, C\} - \{C\}] \qquad \phi_{\mathrm{F}} = \frac{1}{2}[80 - 50] + \frac{1}{2}[100 - 70] = 30$$

$$\phi_{\mathrm{C}} = \frac{1}{2}[\{C\} - \{\}] + \frac{1}{2}[\{F, C\} - \{F\}] \qquad \phi_{\mathrm{C}} = \frac{1}{2}[70 - 50] + \frac{1}{2}[100 - 80] = 20$$

$$\text{prediction} = \text{baseline} + \phi_F + \phi_C = 50 + 30 + 20 = 100$$