Learning with Graph Kernels in the Chemical Universe

Yu-Hang Tang

Luis W. Alvarez Postdoctoral Fellow in Computing Sciences
Computational Research Division
Lawrence Berkeley National Laboratory

Contents

Predicting Molecular Properties

- **Many molecular properties are functions of their structure**
  - Energy/force
  - Chromatography
  - Reactivity
- **But experimentation/computation to acquire the properties can be expensive**
  - Quantum mechanical computations
  - Large amount of sampling
  - Experiment setup
- **Plus, the search space for chemical elements are combinatorially large**

Need for ML algorithm that can **not only learn from data**, but also can **guide data acquisition**

Gaussian process regression primer

› Conditional distributions of a multivariate normal: given three unit Gaussian random variables A, B, and C, and their covariance matrix Σ, can we **infer the value of C if A and B is known?**

Covariance matrix

	A	B	C
A	**1**	0.5	0.9
B	0.5	**1**	0.8
C	0.9	0.8	**1**

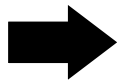
Given	Observe	Intuition
A=2	Cov[A,C]=0.9	C should be close to 2
B=3	Cov[B,C]=0.8	C should also be close to 3

Conclusion: C is probably somewhere between 2 and 3

› The above inference have a **closed-form solution**

$$\mu[C] = \begin{bmatrix}0.9\\0.8\end{bmatrix}^{\mathrm{T}} \begin{bmatrix}1 & 0.3\\0.3 & 1\end{bmatrix}^{-1} \begin{bmatrix}2\\3\end{bmatrix} \approx 2.733$$

$$\mathrm{Var}[C] = 1 - \begin{bmatrix}0.9\\0.8\end{bmatrix}^{\mathrm{T}} \begin{bmatrix}1 & 0.3\\0.3 & 1\end{bmatrix}^{-1} \begin{bmatrix}0.9\\0.8\end{bmatrix} \approx 0.027$$

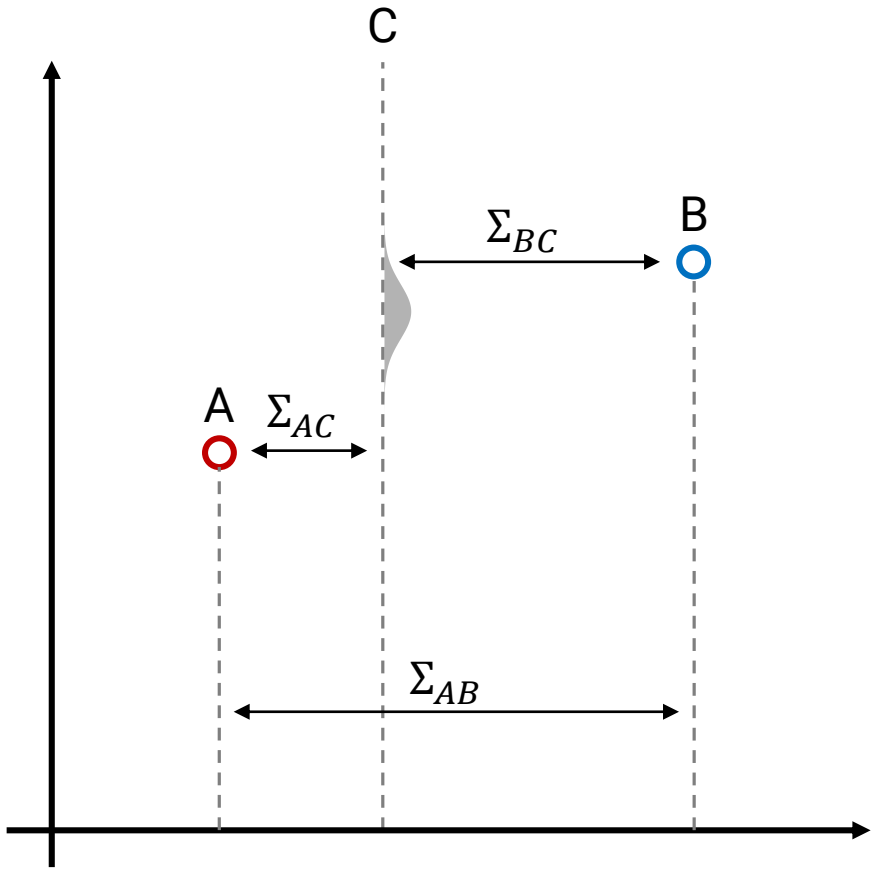


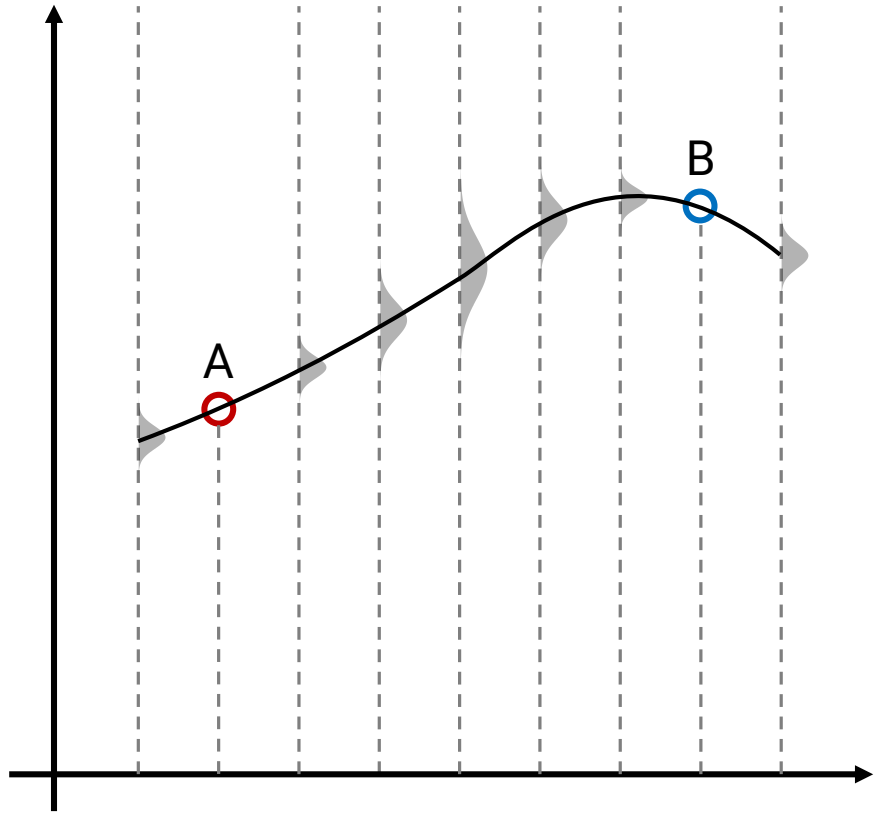
With 95% confidence

$$C = 2.733 \pm 0.054$$

Gaussian process regression (GPR) for supervised learning

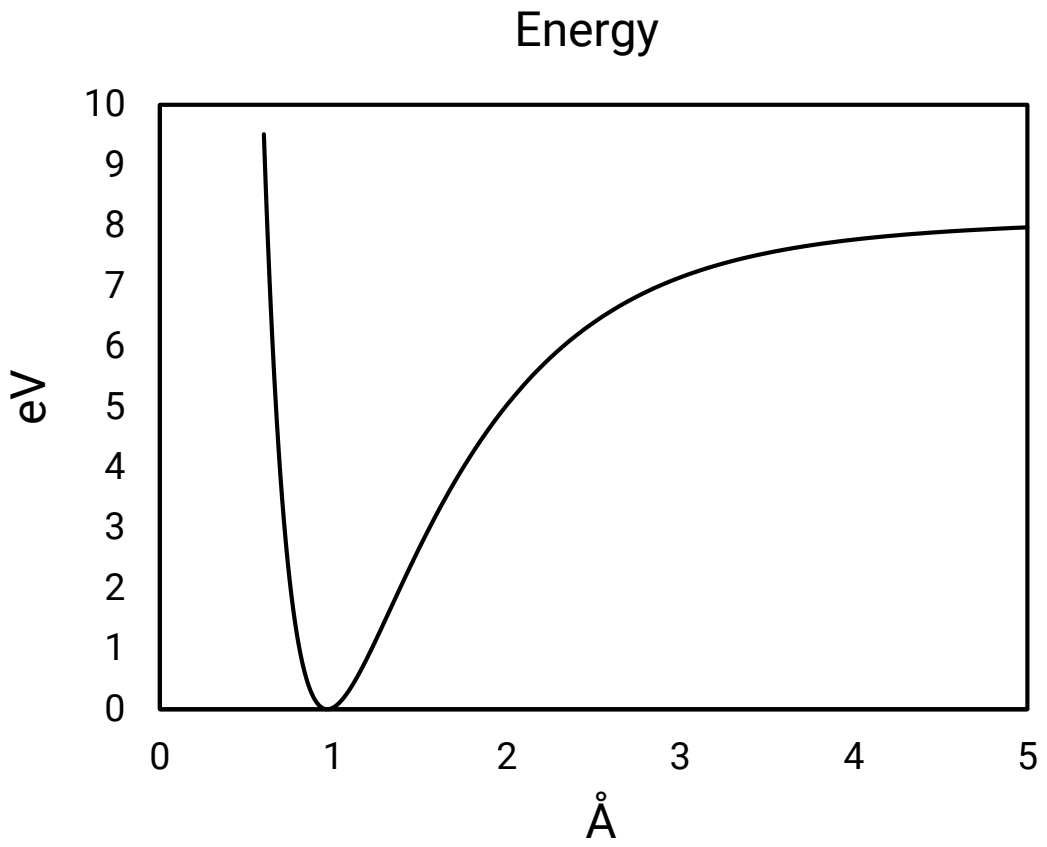
- Given a few sample points (i.e. training data) from a hidden function, can GPR infer what the function is?
  - Yes, assuming **covariance is a function of distance**, e.g. $K(x_1, x_2) = \exp\left[-\frac{1}{2}\frac{(x_1-x_2)^2}{\sigma^2}\right]$

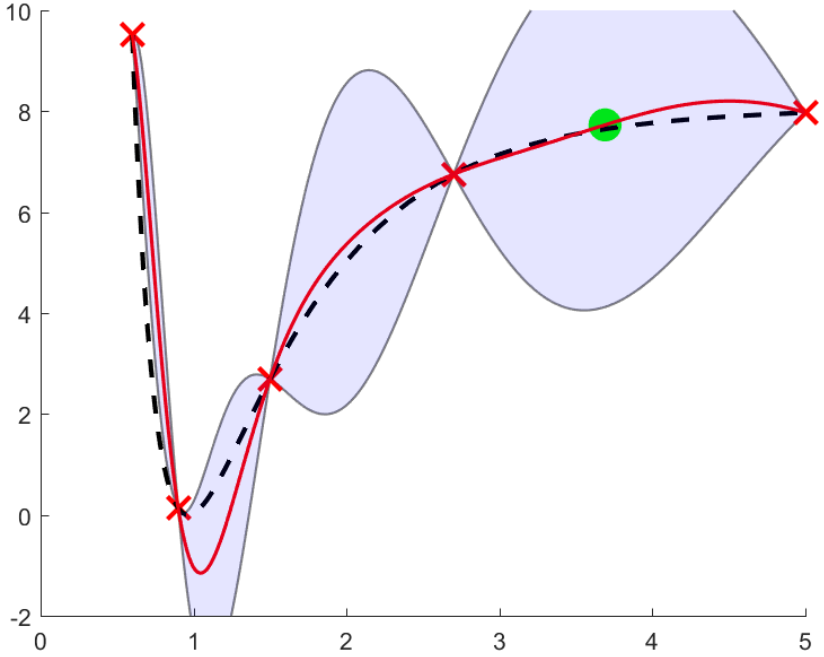




Active learning of potential energy curve using GPR

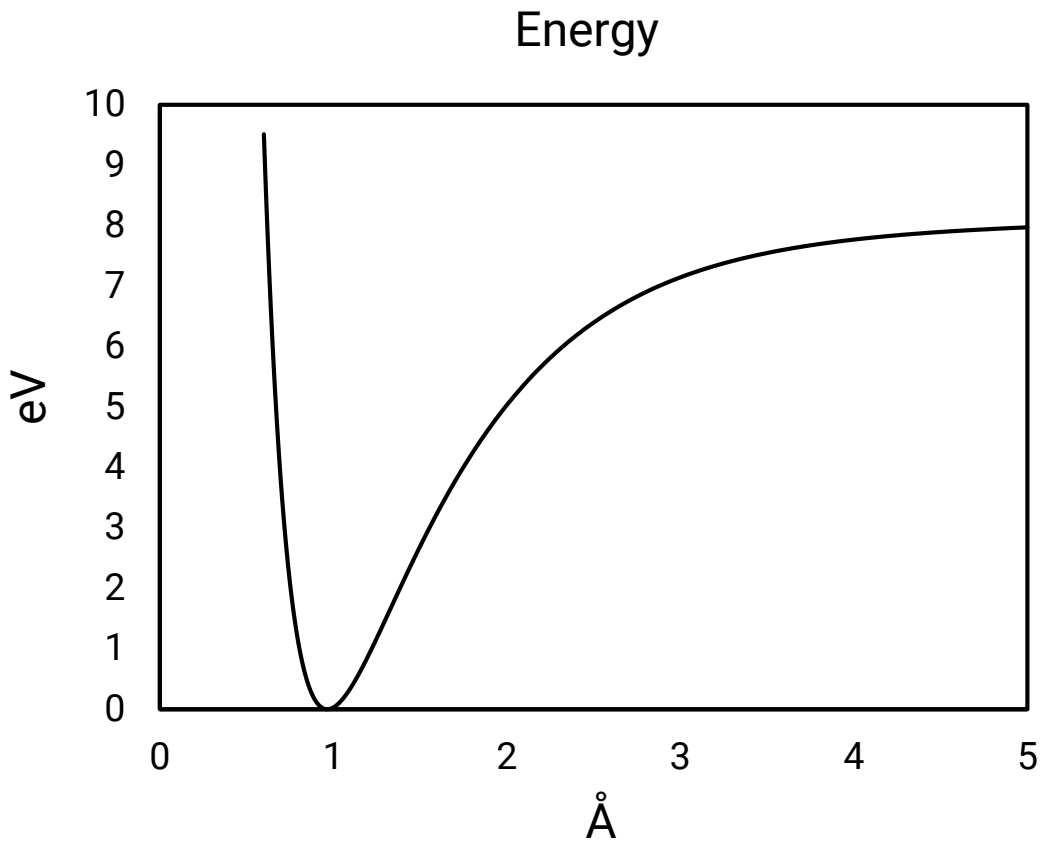
› **Next training point decided on-the-fly, guided by GP's predictive uncertainty**

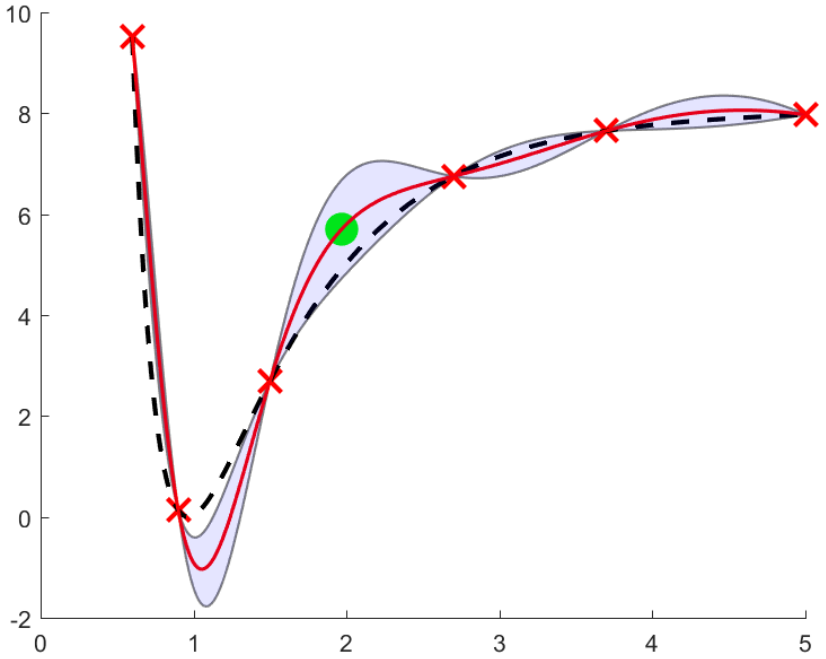




Active learning of potential energy curve using GPR

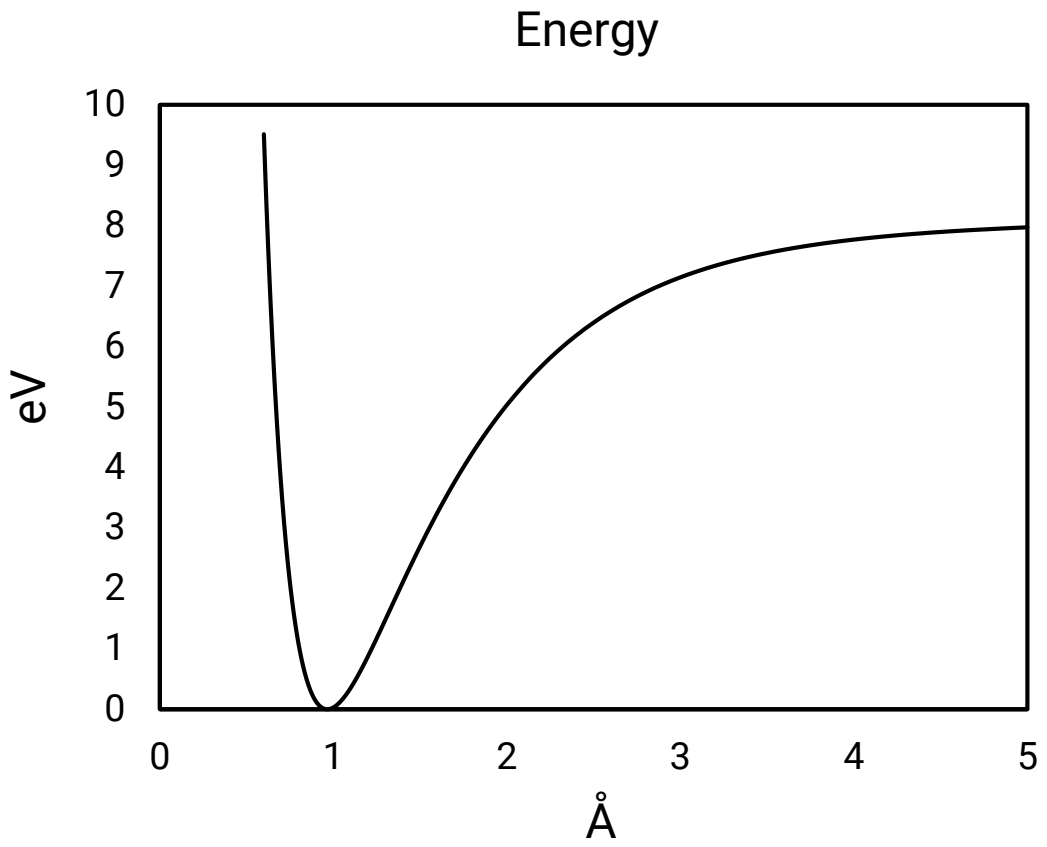
› **Next training point decided on-the-fly, guided by GP's predictive uncertainty**

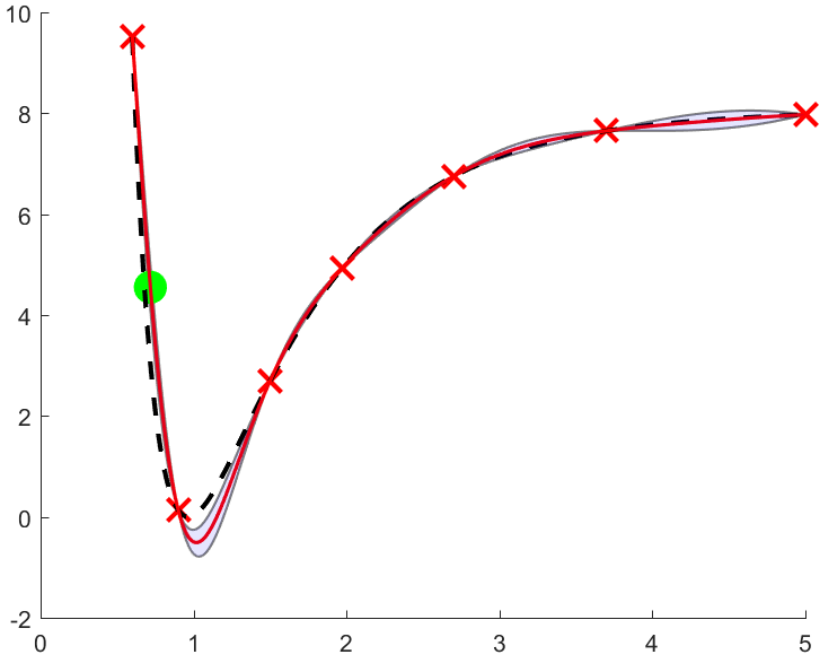




Active learning of potential energy curve using GPR

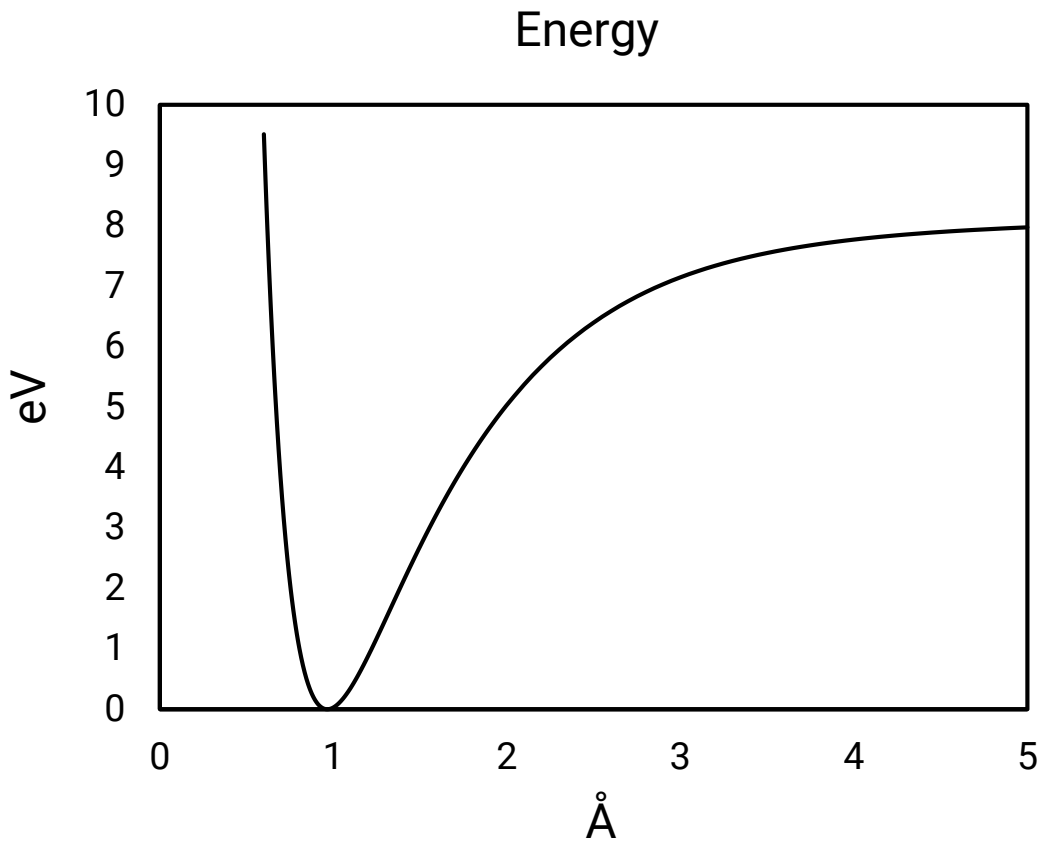
› **Next training point decided on-the-fly, guided by GP's predictive uncertainty**

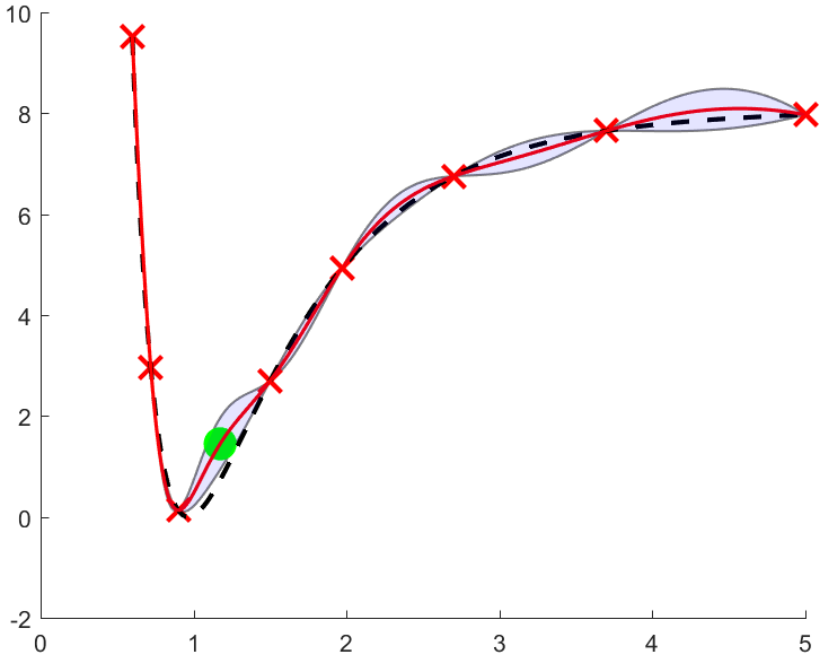




Active learning of potential energy curve using GPR

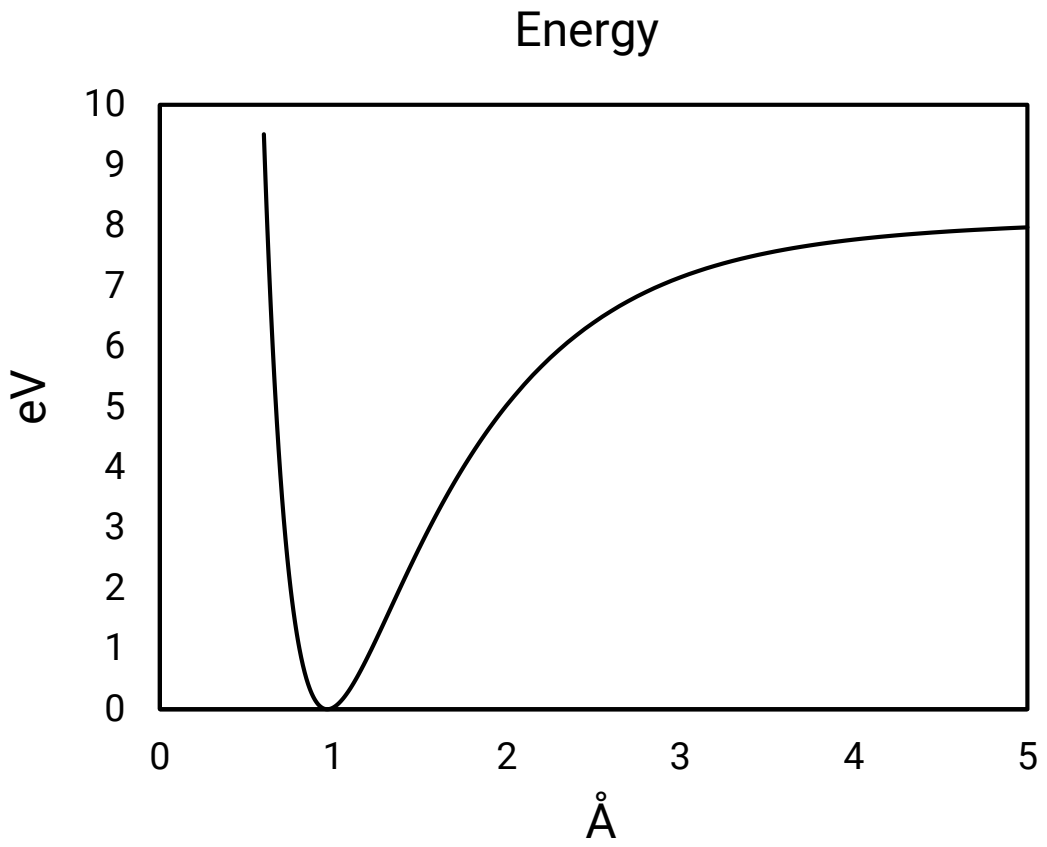
› **Next training point decided on-the-fly, guided by GP's predictive uncertainty**

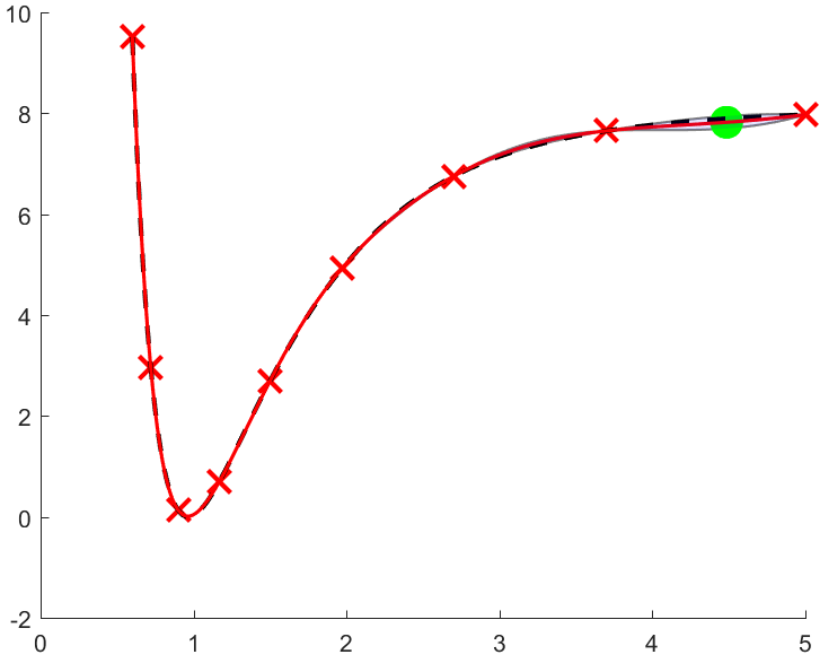




Active learning of potential energy curve using GPR

› **Next training point decided on-the-fly, guided by GP's predictive uncertainty**





The previous example was cheating

- Carried out as a 1D GPR on the real line
- Problem

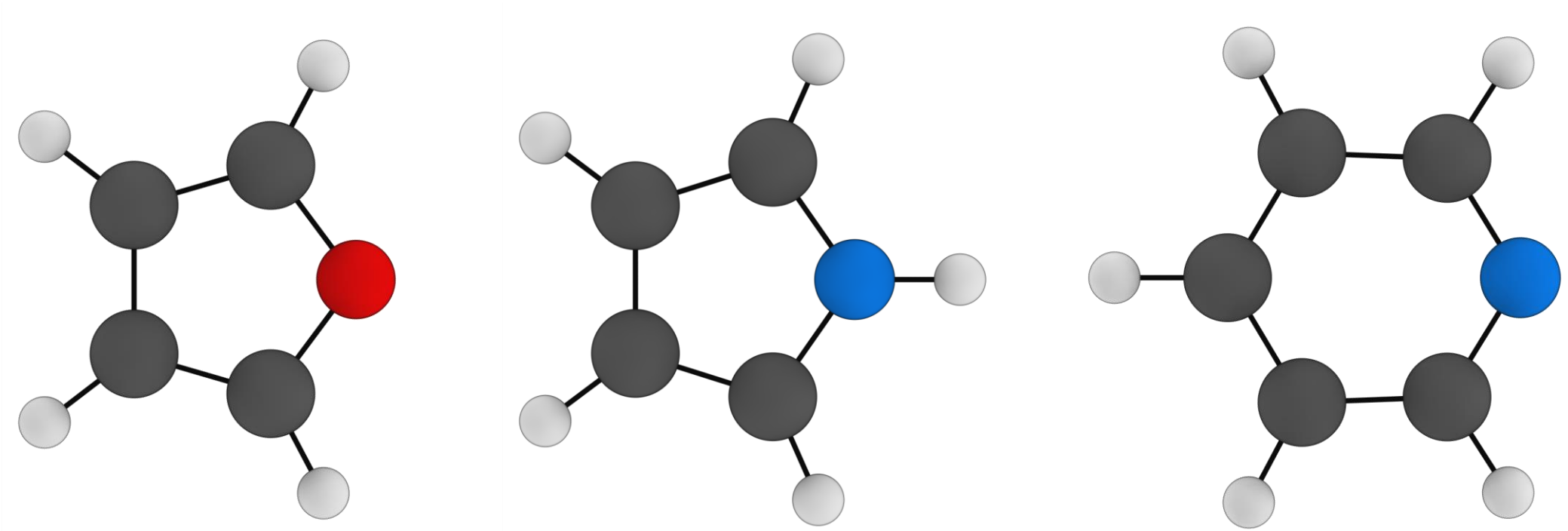
How to define **covariance functions between molecules**?

- Hint: it is our belief that **similar molecules have covariate properties**
  - covariance is statistician's way for describing 'similarity' between random variables
- Need for **similarity quantification between atomistic configurations**

Similarity functions between molecules: challenges

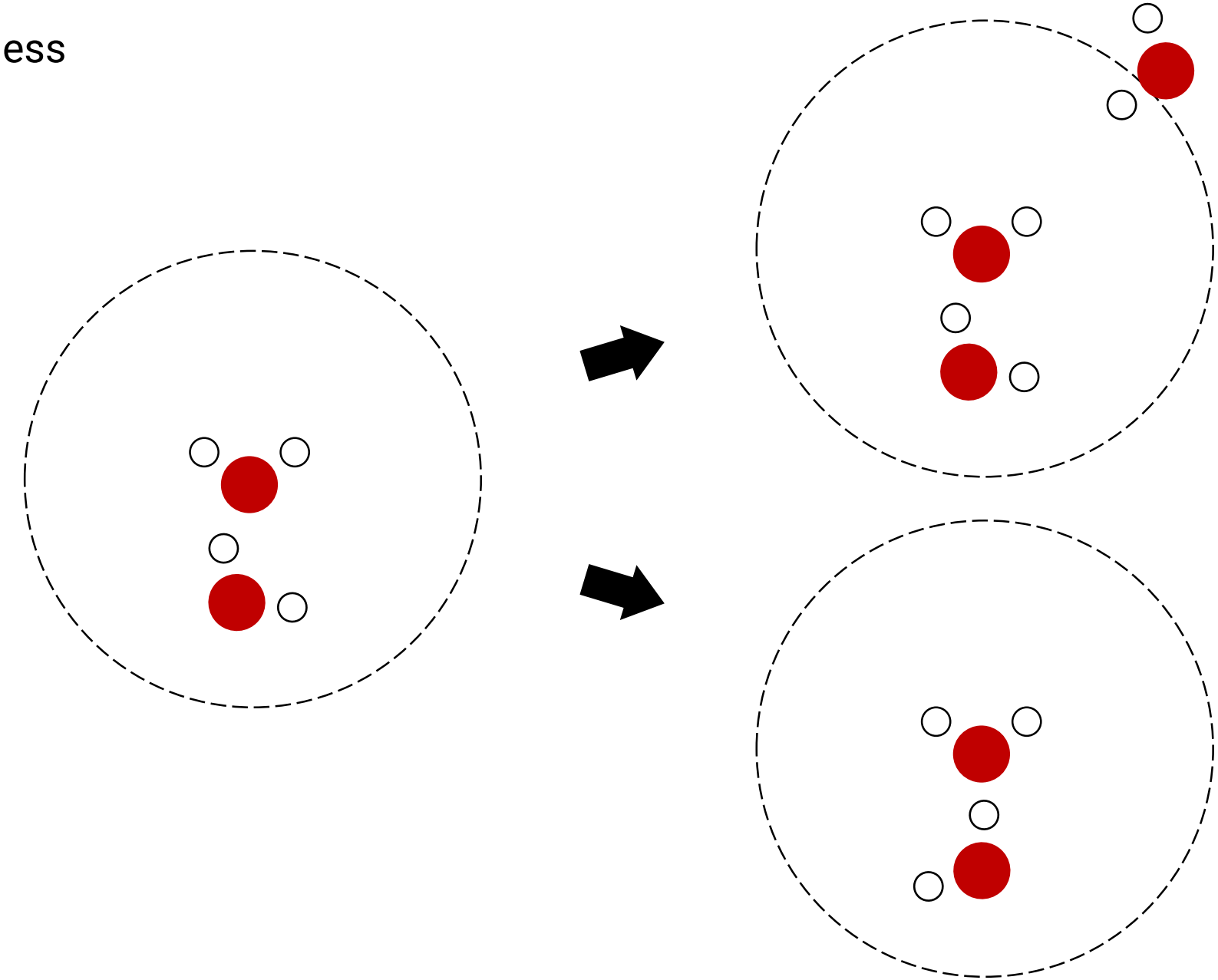
› Variable degrees of freedom

› Discrete label/topology space



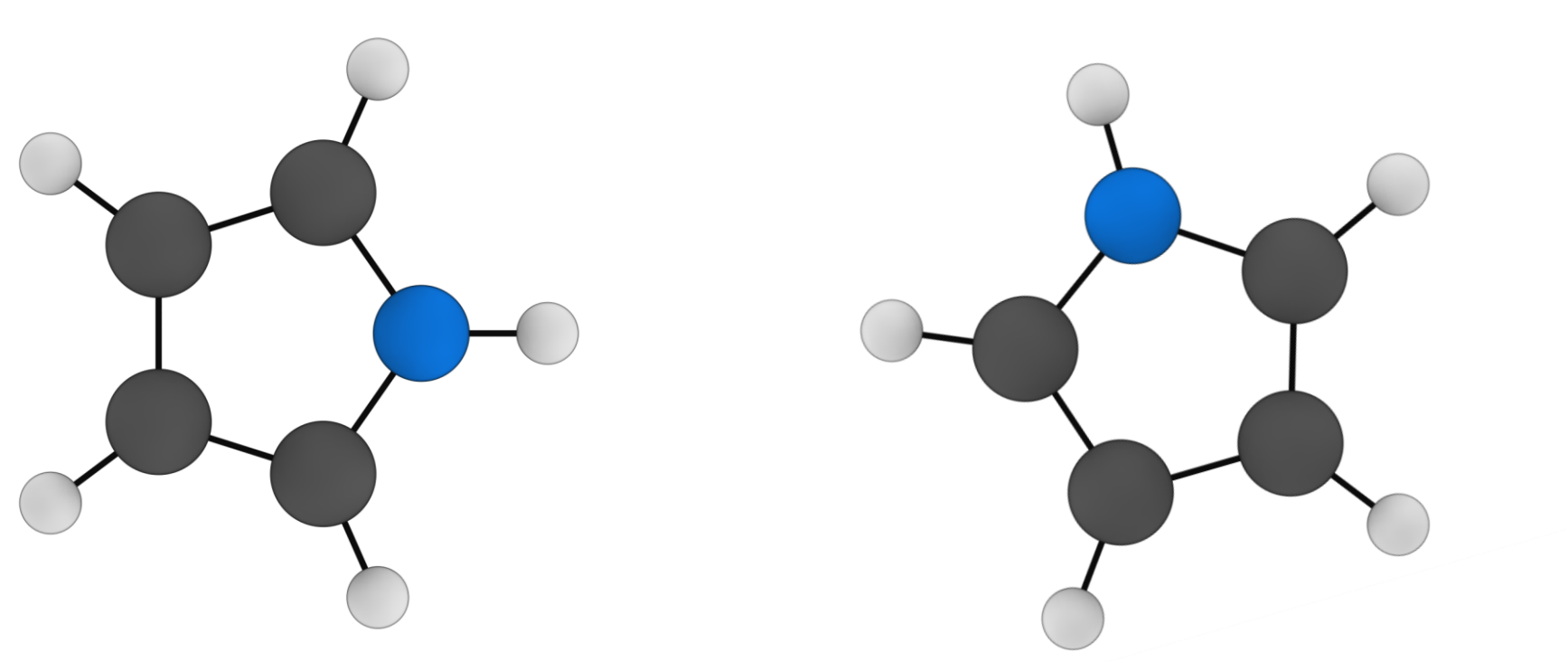
Similarity functions between molecules: challenges

› Smoothness



Similarity functions between molecules: challenges

› Symmetry adaptation



Similarity comparison via feature vectors: detour?

- Well-known fundamental similarity functions
  - The **cosine similarity**: based on angle (similar if pointing in the same direction)
  - Square exponential RBF: based on L_2 distance (similar if close in space)
- For molecules: apply the cosine/Gaussian similarity function on a molecular feature vector
  - Behler-type symmetry functions: Behler. J Chem Phys. 2011
  - Eigenspectrum of coulomb matrix: Rupp et al. PRL. 2012
  - SOAP: spherical harmonics expansion of density. Bartók et al. PRB. 2013
  - Bispectrum of mass density. Bartók et al. PRL. 2010
  - DECAF: optimal quadrature expansion of density + canonical alignment
    - Y.-H. Tang et al. JCP. 2018: An atomistic fingerprint algorithm for learning ab initio molecular force fields https://doi.org/10.1063/1.5008630
    - J Chem Phys 2018 Editors' Choice
  - and hundreds more...

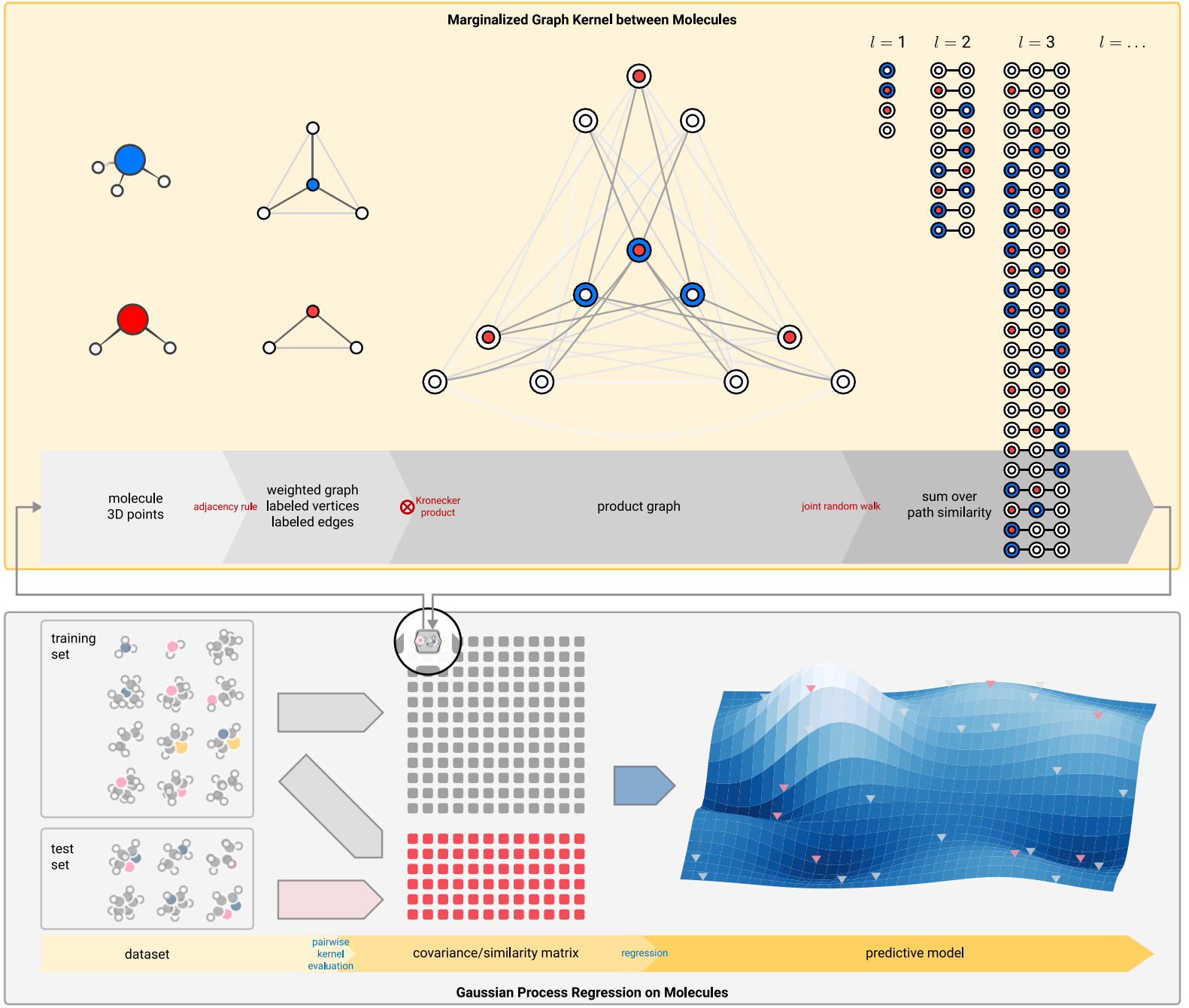
Similarity between structured data

- Molecules are intrinsically graphs with
  - **Variable numbers of nodes and edges**
  - Non-sequential connectivity between components
- Explicit feature vectors might be a **detour**, since eventually only a single number (the covariance) is needed.
- **The marginalized graph kernel** is specifically designed to overcome the above issues
  - Construct implicit feature space formed by joint random walks on the graphs
  - Built-in symmetry invariance
  - Scales to arbitrary number of atom/bond types

Kashima, Hisashi, Koji Tsuda, and Akihiro Inokuchi. “Marginalized Kernels between Labeled Graphs.” In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 321–328, 2003.

Ferre, Haut, Barros. “Learning molecular energies using localized graph kernels.” J. Chem. Phys. 146, 114107 (2017)

Gaussian Process Regression using the Marginalized Graph Kernel



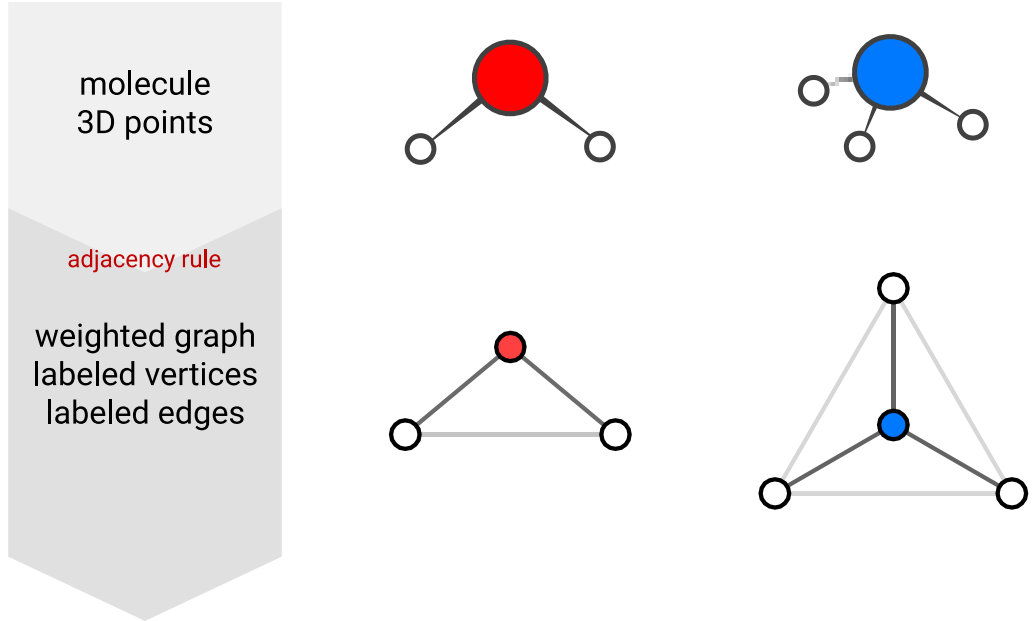
Tang & de Jong, J Chem Phys, 2019: Prediction of atomization energy using graph kernel and active learning
https://doi.org/10.1063/1.5078640

Convert 3D molecular geometry to an undirected, weighted graph

- **Atoms as vertices**
- **Use an adjacency rule to create edges with weights decaying by distance**
  - For example, a Gaussian adjacency rule

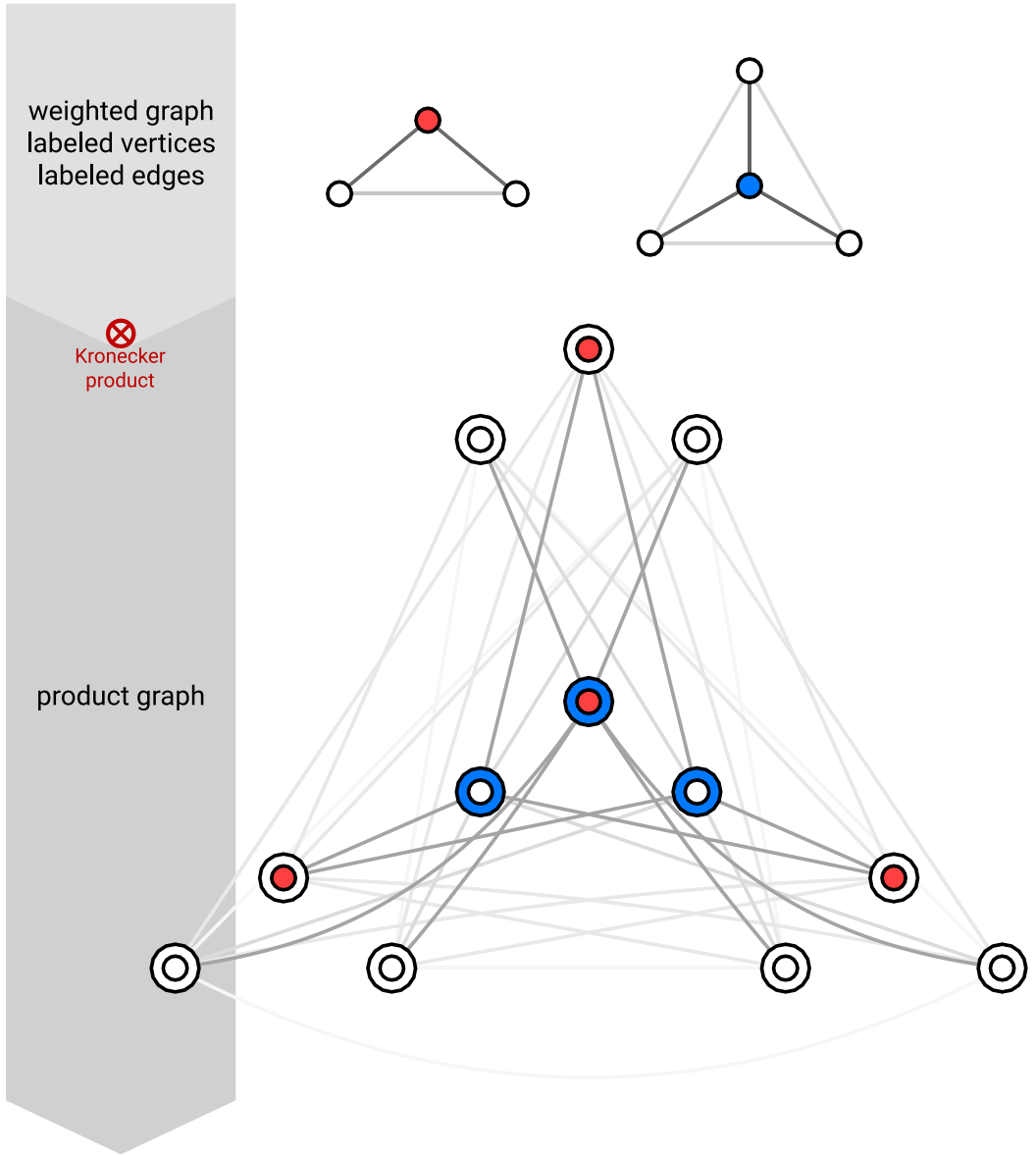
$$w_{ij} = \exp\left[-\frac{1}{2}\frac{\left(\boldsymbol{r}_i - \boldsymbol{r}_j\right)^2}{\left(\lambda\, b_{ij}\right)^2}\right]$$

  - b_{ij} is the average bond length between elements
  - λ is a linear scaling factor



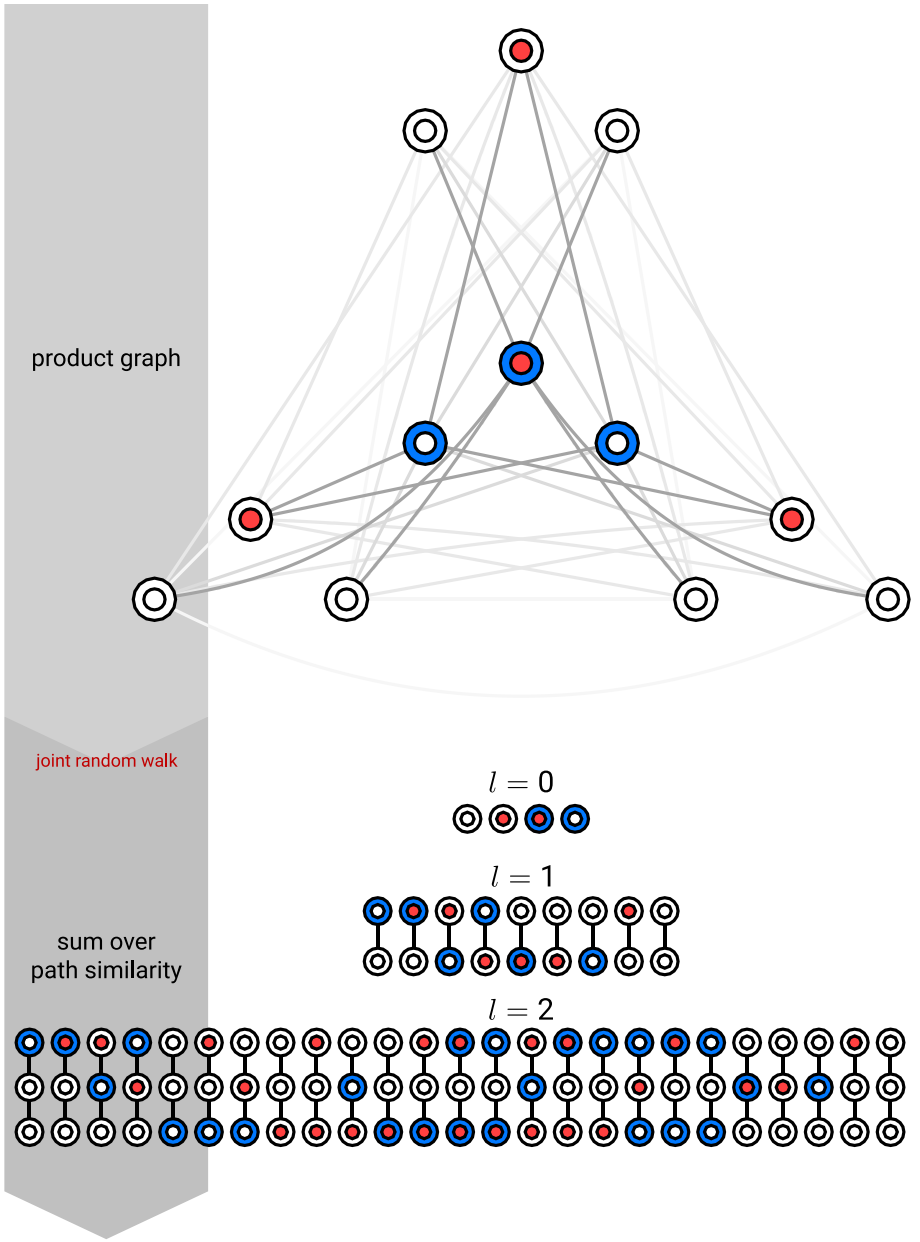
Formation of Product Graph

- **A product graph is a graph where**
  - a vertex is a pair of vertices, each from a smaller graph
  - an edge exists if the two pairs of constituting vertices are both connected in the smaller graph



Perform random walk on the graph, and sum over path similarity

- Jump probability proportional to edge weight
- Stopping probability determines average path length
- Sum over all possible paths of potentially infinite length



Marginalized graph kernel: computation

$$K(G,G') = \sum_{l=1}^{\infty} \sum_{\boldsymbol{h}} \sum_{\boldsymbol{h}'} p_{\mathrm{s}}(h_1) p'_{\mathrm{s}}(h'_1) K_{\mathrm{v}}\left(v_{h_1}, v'_{h'_1}\right) \prod_{i=2}^{l} p_{\mathrm{t}}(h_i|h_{i-1}) p_{\mathrm{q}}(h_l) \prod_{j=2}^{l} p'_{\mathrm{t}}(h'_j|h'_{j-1}) p'_{\mathrm{q}}(h'_l) \prod_{k=2}^{l} K_{\mathrm{e}}\left(e_{h_{k-1}h_k}, e_{h'_{k-1}h'_k}\right) K_{\mathrm{v}}\left(v_{h_k}, v'_{h'_k}\right)$$

A (slightly) more friendly version of the kernel is

$$K(G,G') = \mathbf{s}_{\times} \cdot \mathbf{R}_{\infty},$$

where R_∞ can be solved from

$$\left[\mathbf{D}_{\times}\mathbf{V}_{\times}^{-1} - \mathbf{A}_{\times} \odot \mathbf{E}_{\times}\right] \mathbf{R}_{\infty} = \mathbf{D}_{\times}\, \mathbf{q}_{\times}.$$

$\mathbf{D}_{\times}$: vertex degree matrix
$\mathbf{V}_{\times}$: vertex label similarity matrix
$\mathbf{A}_{\times}$: adjacency matrix
$\mathbf{E}_{\times}$: edge similarity matrix
$\mathbf{q}_{\times}$: stopping probability

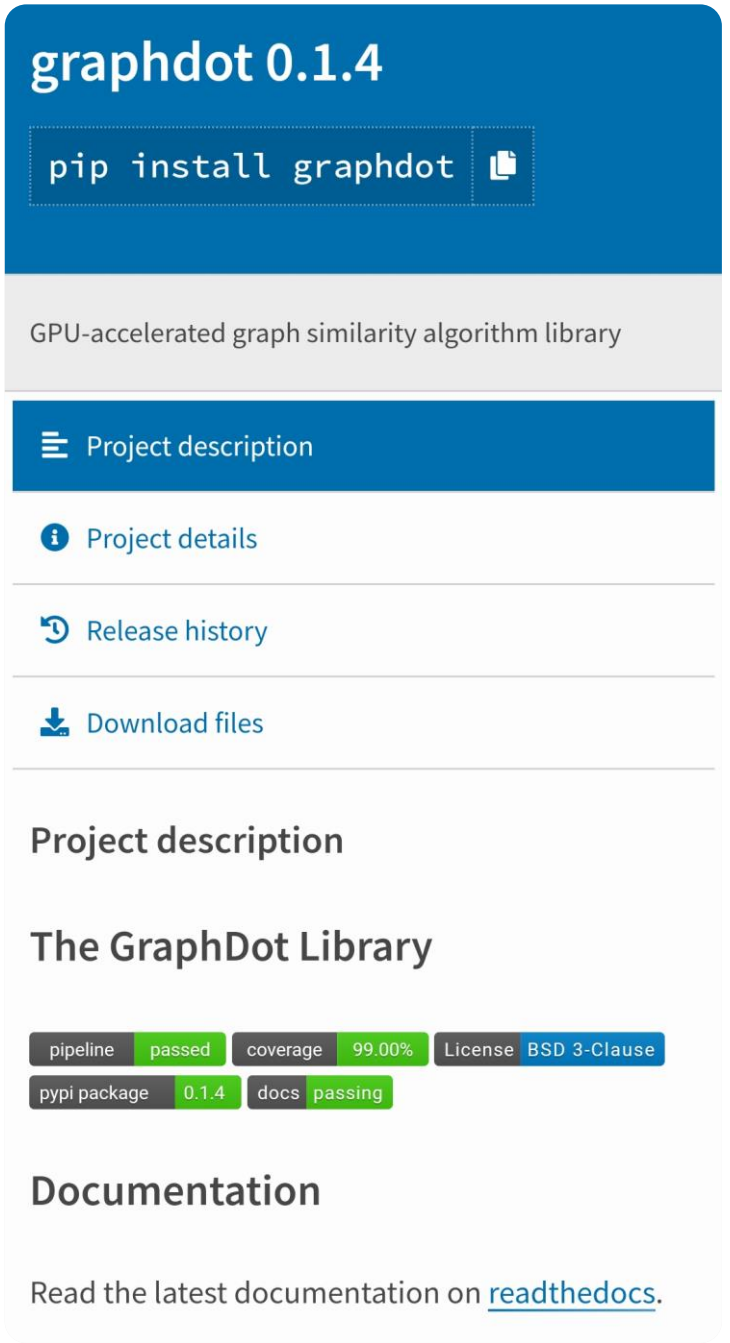
GraphDot: graph kernel made easy

Repository: https://gitlab.com/yhtang/graphdot

PyPI: https://pypi.org/project/graphdot/

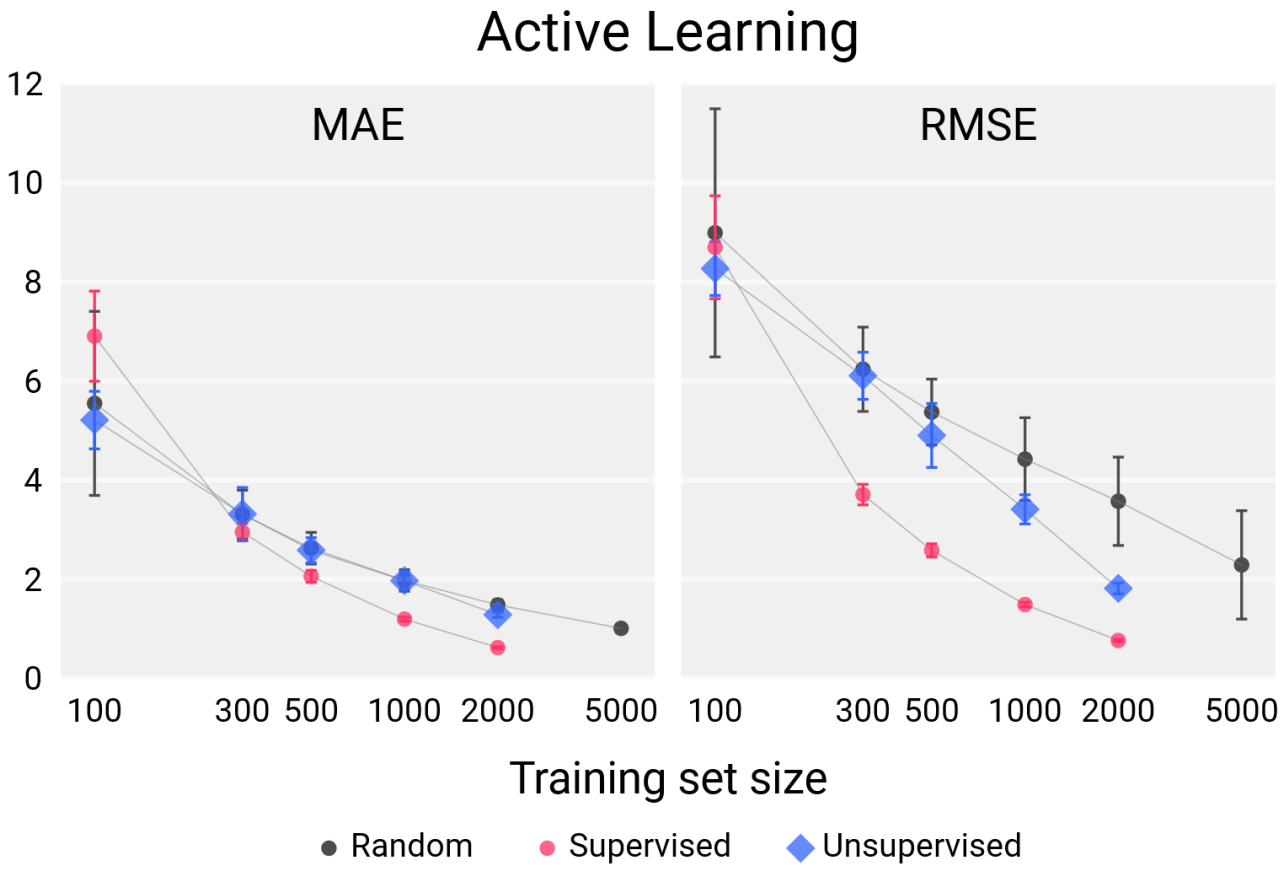
Documentation: https://graphdot.readthedocs.io/en/latest/

- Fully featured: **for and beyond molecules!**
  - Weighted graphs with both nodes and edges labeled
  - Arbitrary attributes and custom base similarity kernels
- **GPU-accelerated**
  - Just-in-time code generation and compilation
  - 100x speedup compared to existing CPU packages such as GraKeL and graphkernels
- **Interoperable** with ASE, NetworkX, pymatgen
  - Scikit-learn compatible python interface



Example & benchmark

- **QM7: 7165 small organic molecules consisting of H, C, N, O, S, up to 23 atoms**
  - From scratch training time: N = 1000: 10 s training, 0.018 s/sample predicting, N = 2000: 40 s training, 0.034 s/sample predicting
- **Supervised learning: use predictive error to determine the next sample**
- **Unsupervised active learning: use predictive variance**



Summary

- **Active learning using GPR can be powerful for predicting molecular properties**
- **The marginalized graph kernel is an ideal covariance function for Gaussian process regression of molecular energy**
- **The GraphDot library is a high-performance and easy-to-use python package for graph kernel computations**

Thank you!

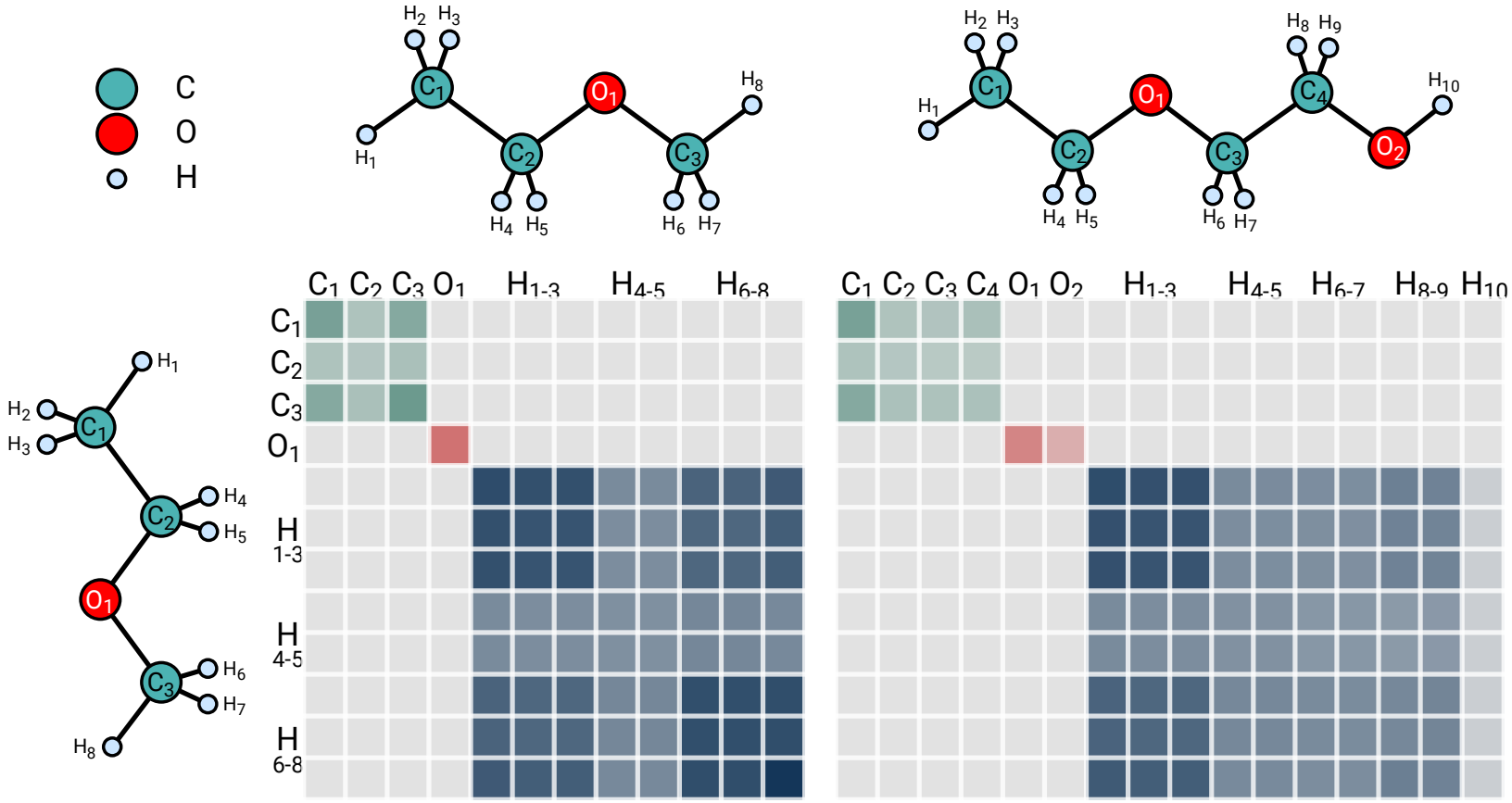
Acknowledgment

- LBNL LDRD Project “Active Learning of Ab Initio Force Fields with Applications to Large-Scale Simulations of Materials and Biophysical Systems”
- Work also supported in part by the Applied Mathematics program of the DOE Office of Advanced Scientific Computing Research under Contract No. DE-AC02-05CH11231, and in part by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

Marginalized graph kernel: application

› The elements of $\mathbf{R}_\infty$ can be interpreted as an **atom-wise similarity matrix**

› The **sum of the elements** of $\mathbf{R}_\infty$, before normalization, defines a kernel that allows **automatic scaling** when predicting extensive variables



Example & benchmark

- **QM7: 7165 small organic molecules consisting of H, C, N, O, S, up to 23 atoms**
  - From scratch training time: N = 1000: 10 s training, 0.018 s/sample predicting, N = 2000: 40 s training, 0.034 s/sample predicting