Atomic Descriptors For Machine Learning Applications

Ali Sadeghi

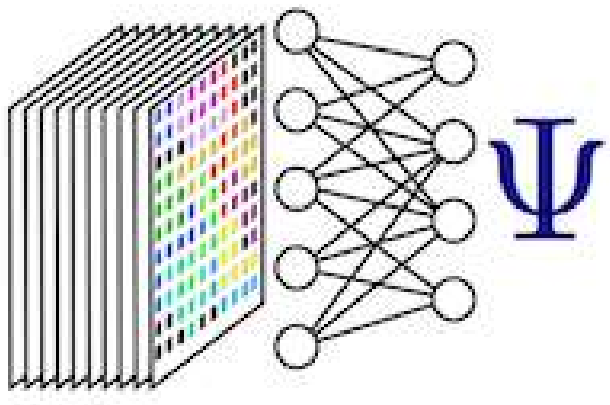
Department of Physics
Shahid Beheshti University

2nd Workshop on Machine Leaning in Physics
University of Tehran
Mehr 11-13, 1397

Outline

CMP: Fundamental model-based understanding (conventional/emergent) states of matter: properties ... transitions.
ML: Approx. data-driven potability distributions p: map input to output $x \xrightarrow{f} y$.

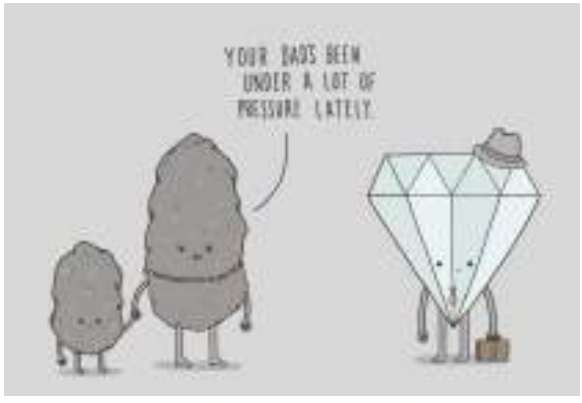
ML for CMP:

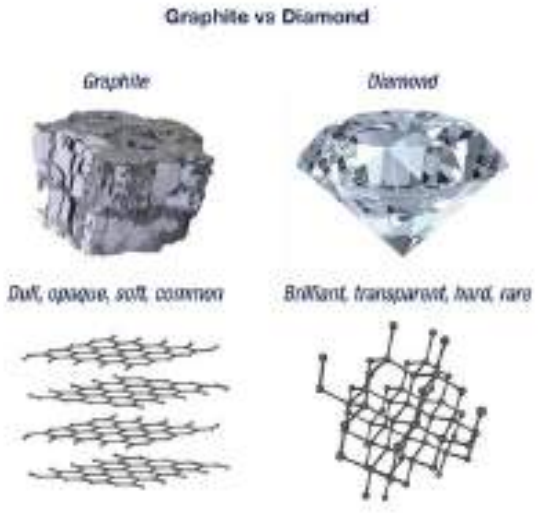


- $p(x, y)$: Unsupervised learning for e.g. WF representation and compression by Boltzmann machine
- $p(x|y)$: Classification for e.g. matter phase transition
- $p(y|x)$: Prediction for e.g. material properties and IP

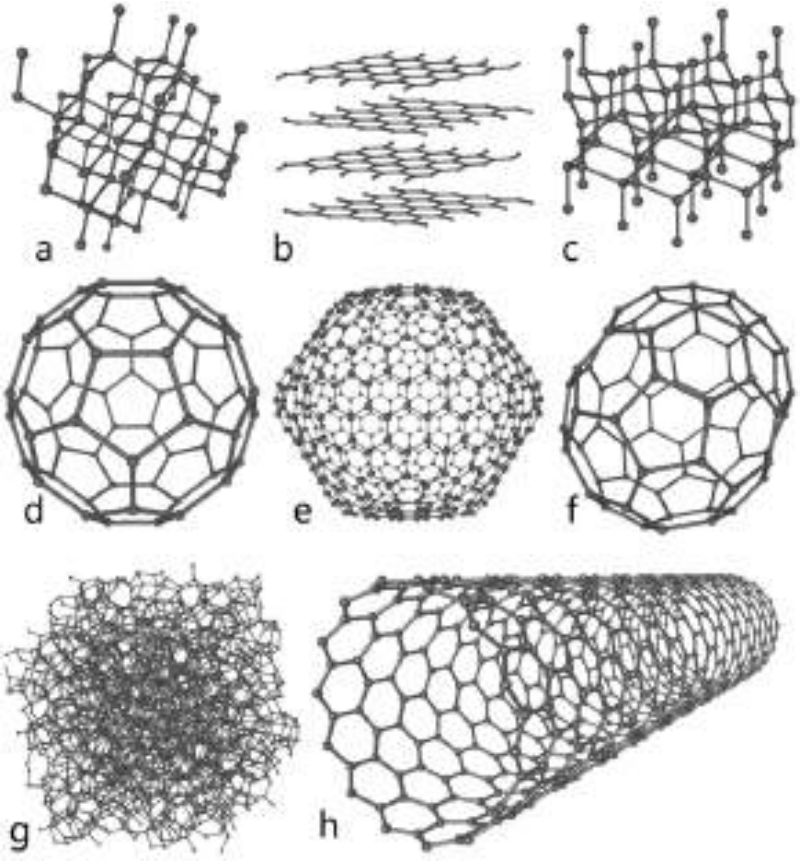
Atomic configuration

Materials properties depend on their atomic structure.





Several allotropes exist



Atomic configuration matters a lot!

What is Atomic Configuration?

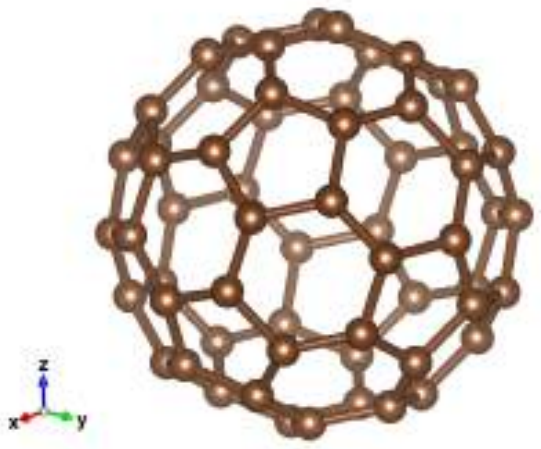
Arrangement of atoms in molecules, clusters, crystals, ...

Clusters:

$$\begin{cases} \mathbf{Z} \equiv \{Z_i\} \in \mathbb{R}^{N_{\text{atom}}} \\ \mathbf{R} \equiv \{x_1, y_1, z_1; x_2, y_2, \cdots z_{N_{\text{atom}}}\} \in \mathbb{R}^{3N_{\text{atom}}} \end{cases}$$

XYZ format:

```
60
Fullerene C60
C     0.504958     3.505671     0.648885
C     1.942663     3.068513     0.574739
C     3.126844     1.588262    -1.157148
C     3.726664     0.781137    -0.037884
C     3.429356     1.124146     1.350137
C     3.755443    -0.660625    -0.466759
C     3.173969    -0.744798    -1.853158
C     2.785029     0.646138    -2.279243
...
```



What is Atomic Configuration?

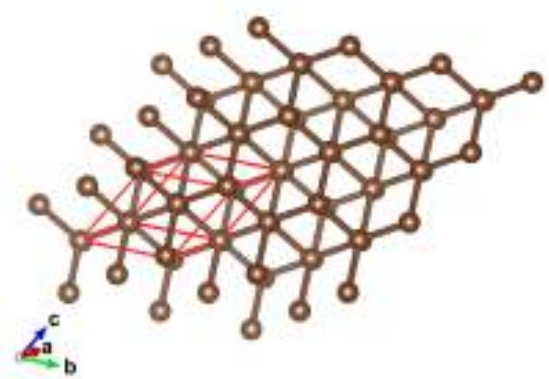
Arrangement of atoms in molecules, clusters, crystals, ...

Crystals:

$$\begin{cases} \mathbf{Z} \equiv \{Z_i\} \in \mathbb{R}^{N_{\text{atom}}} \\ \mathbf{R} \equiv \{x_1, y_1, z_1; x_2, y_2, \cdots z_{N_{\text{atom}}}\} \in \mathbb{R}^{3N_{\text{atom}}} \\ \mathbf{a} \equiv \{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\} \in \mathbb{R}^{3\times 3} \end{cases}$$

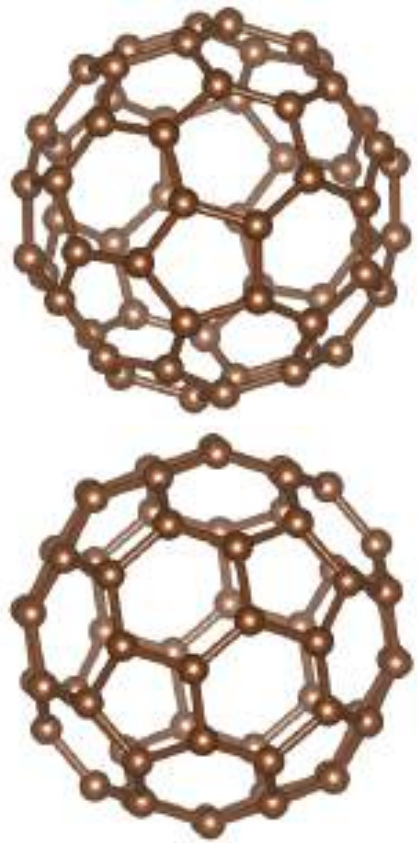
VASP format:

```
Diamond              comment line
3.7                  universal scaling factor
0.5  0.5  0.0        1st Bravais lattice vector
0.0  0.5  0.5        2nd Bravais lattice vector
0.5  0.0  0.5        3rd Bravais lattice vector
C                    atomic types
2                    # of atoms per species
direct               Direct or Cartesian
0.0  0.0  0.0        positions
0.25  0.25  0.25
```



Naive Distances based on $\mathbf{R} \in \mathbb{R}^{3N}$

- The worst: the Frobenius norm $d(A,B) = \|\mathbf{R}^A - \mathbf{R}^B\|$ depends on
  - choice of origin
  - orientation
  - index order

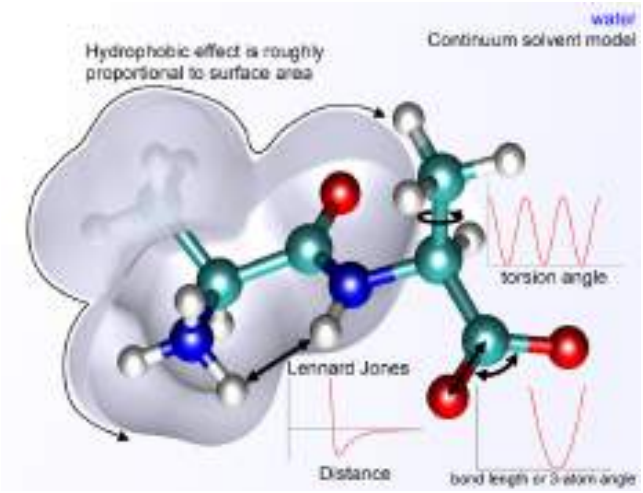


- The best: $\text{RMSD}(A,B) = \frac{1}{\sqrt{N}} \underbrace{min}_{\pi,U,\mathbf{d}} \|\mathbf{R}^A + \mathbf{d} - U\mathbf{R}^B\pi\|$

RMSD is Computationally too costly

- alignment (rigid shift+rotation) is easy (analytic!)
- index matching (permutation) is very expensive:
  - For similar structures: Hungarian algorithm $(O(N))$
  - For distinct structures: MC gives $N! \Rightarrow \exp(N)$

Material Property y



CMP

Compute property y from $\mathbf{Z}, \mathbf{R}, \mathbf{a}$

Cartesian $\mathbf{R}$: perfect for calculations:
Get energy from

$$E = E_0 + \sum_{i,j} E_2(r_{ij}) + \sum E_3(r_{ijk}) + \dots$$

Get everything! by solving numerically

$$H\Psi(\mathbf{R}) = E\Psi(\mathbf{R})$$

Outline

Material Property y

ML

Mapping $(\mathbf{Z}, \mathbf{R}, \mathbf{a}) \xrightarrow{f=?} y$

Train f on m samples, by minimizing
Err. $= \sum_{j=1}^{m} (y^{(j)} - y^{(j)}_{\text{pred}})^2$

$$\begin{pmatrix} (\mathbf{Z}, \mathbf{R}, \mathbf{a})^{(1)} \\ (\mathbf{Z}, \mathbf{R}, \mathbf{a})^{(2)} \\ \vdots \\ (\mathbf{Z}, \mathbf{R}, \mathbf{a})^{(\mathrm{m})} \end{pmatrix} \xrightarrow{f} \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{pmatrix}$$

Cartesian $\mathbf{R}, \mathbf{a}$: useless input **feature**!
Not invariant to:

- Index permutation
- Rigid translation
- Rotation
- choice of cell shape $\mathbf{a}$

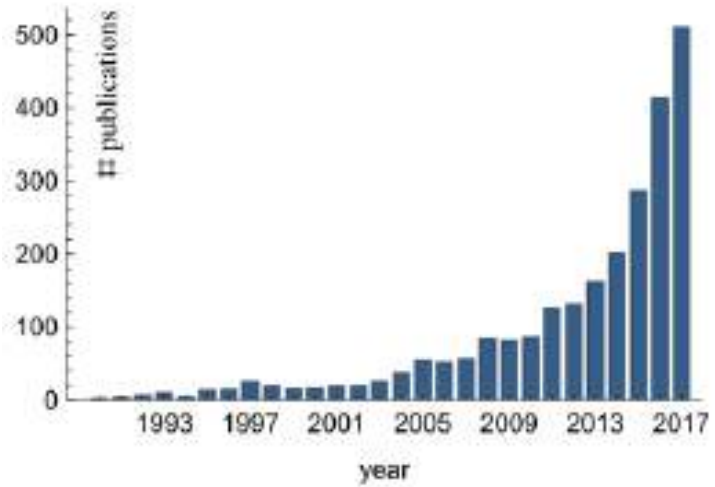
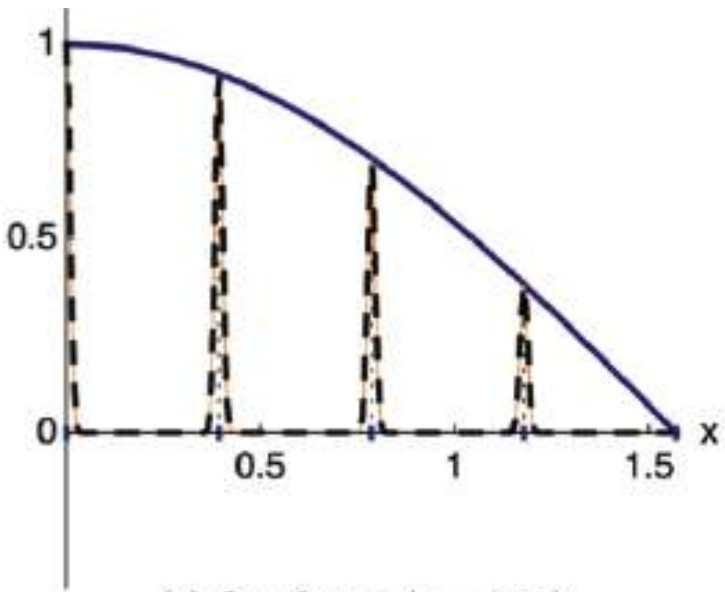


FIG. 1. *Number of publications per year* from a web of science search for articles with topics of machine learning and either chemistry or materials, taken June 5, 2018. The average number of citations per article is 12.

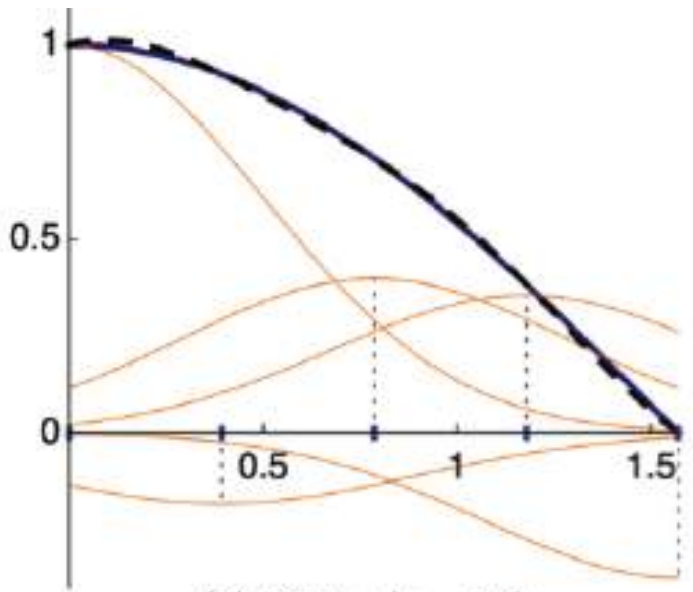
JCP 148,241401 (2018)

Kernel Ridge Regression

- Machine learning: $\mathbf{x} \xrightarrow{f=?} y$
- For example: $y \simeq f(\mathbf{x}) = \sum_{j=1}^{m} w_j k(\mathbf{x}, \mathbf{x}_j)$
- similarity metric $k(\mathbf{x_i}, \mathbf{x_j}) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}\right)$



(a) Overfitting ($\sigma = 0.01$)



(b) Fitting ($\sigma = 0.5$)

IJQC 115, (2015)

Kernel Ridge Regression

- $f(\mathbf{x}) = \sum_{j=1}^{m} w_j k(\mathbf{x}, \mathbf{x}_j)$
- **Feature** vector $\mathbf{x} \in \mathbb{R}^{l_{\text{FP}}}$ is our **fingerprint** of length l_{FP}
- **Label** $y \in \mathbb{R}$ is the desired **property**
- Model $f = \underset{f}{\text{argmin}} \sum_{j=1}^{m} \left(y^{(j)} - f(\mathbf{x}^{(j)}) \right)^2 + \lambda \mathbf{w}^{\mathbf{T}} \mathbf{K} \mathbf{w}$ is trained via parameters $\{w_i\}$ as $(\mathbf{K} + \lambda \mathbf{I})\mathbf{w} = \mathbf{y}$
- Hyper-parameters λ, σ prevent overfitting via Reg.
- **Train set** consists of m samples $(\mathbf{x}^{(\text{j})}, y^{(\text{j})})$ used to train
- **Test set** of samples is used to verify the accuracy of the machine output.

Structure Fingerprint

ML

Training $f(\mathbf{x}) = \sum_{j=1}^{m} w_j k(\mathbf{x}, \mathbf{x}_j)$ depends critically on the quality of the similarity metric

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\sigma^2}\right)$$

$$\begin{pmatrix} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \\ \vdots \\ \mathbf{x}^{(m)} \end{pmatrix} \xrightarrow{\text{train } f} \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{pmatrix}$$

$$\mathbf{x}^{\text{new}} \xrightarrow{\text{use } f} y^{\text{pred}}$$

A fingerprint, i.e. a **feature** for ML, should be invariant under

- Index permutation
- Rigid translation
- Rotation
- choice of cell shape $\mathbf{a}$

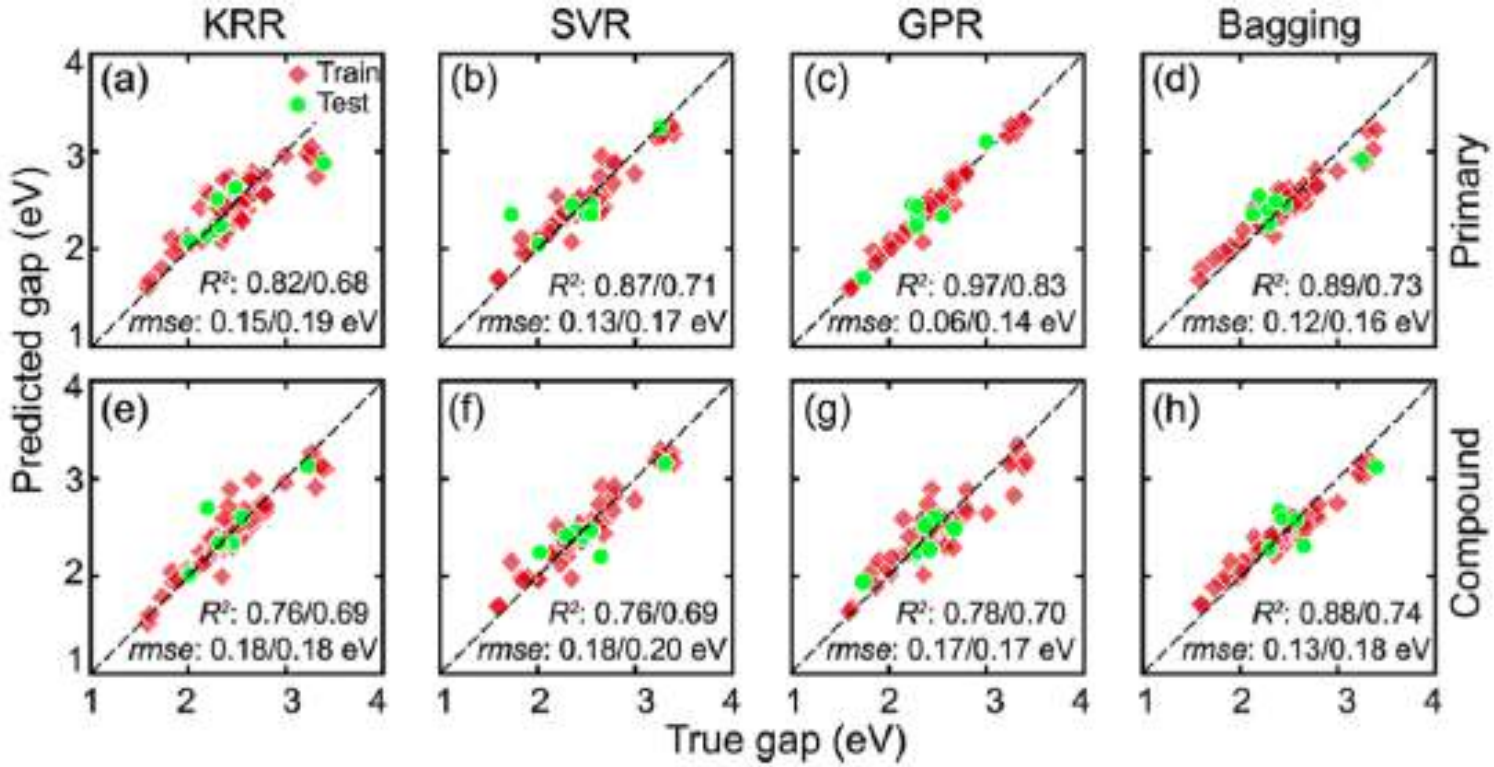
Configuration Fingerprints

To describe a complex compound, two kinds of properties can be employed:

- Physical and chemical, and elemental properties
  - melting point, boiling point, thermal conductivity, ...
  - atomic number, atomic mass, row and columns position the periodic table, electro negativity, ...
  - covalent radius, ionic radius, vdW radius, ...
  - number of single, double and triple bonds, ...
  - the statistical mean of the above quantities
  - ...
- Structural properties
  - radial distribution function
  - average of spherical harmonics for neighbours
  - Coulomb matrix
  - ...

Configuration Fingerprints

ML Band Gap Predictions of Functionalized MXene



Chem Mater, Rajan et al. (2018)

Efficiency

Trade-off between: prediction accuracy and computational costs

Feature matrix, weights vector, FP:

$$\begin{pmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & \cdots & x_{l_{\text{FP}}}^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & \cdots & x_{l_{\text{FP}}}^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{(\text{m})} & x_2^{(\text{m})} & x_3^{(\text{m})} & \cdots & x_{l_{\text{FP}}}^{(\text{m})} \end{pmatrix}_{m \times l_{FP}} , \begin{pmatrix} w^{(1)} \\ w^{(2)} \\ \vdots \\ w^{(m)} \end{pmatrix}_{m \times 1} , (x_1, \cdots x_{l_{\text{FP}}})$$

- Effective sampling (reduce m)
  - Diversity
  - Scoring, duplicates elimination
- Dimensionality reduction (reduce l_{FP})
  - Naturally short FP
  - LASSO, Corr (x_i, y) & Corr(x_i, x_j), CUR, $\pm$ PCA, SVD
- Parallelism

Outline

ML Atomic Properties

Atomic Property

Mapping $(Z_i, \mathbf{r_i}) \xrightarrow{f=?} y$

$$\begin{pmatrix} (Z,\mathbf{r})^{(1)} \\ (Z,\mathbf{r})^{(2)} \\ \vdots \\ (Z,\mathbf{r})^{(\mathsf{m})} \end{pmatrix} \xrightarrow{f} \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{pmatrix}$$

$m = N_{\mathsf{atom}} N_{\mathsf{conf.}}$ samples.

* Cartesian $\mathbf{r}$: useless input **feature**!
* $\mathbf{w}_A, \mathbf{w}_B, \ldots$ can be trained separately for different species A, B, ... to speed up the training and prediction (smaller kernel matrices $k_{m' \times m'}$).

$$\begin{pmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & \cdots & x_{l_{\mathsf{FP}}}^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & \cdots & x_{l_{\mathsf{FP}}}^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{(\mathsf{m})} & x_2^{(\mathsf{m})} & x_3^{(\mathsf{m})} & \cdots & x_{l_{\mathsf{FP}}}^{(\mathsf{m})} \end{pmatrix}_{(m_A+m_B+\ldots)\times l_{FP}}$$

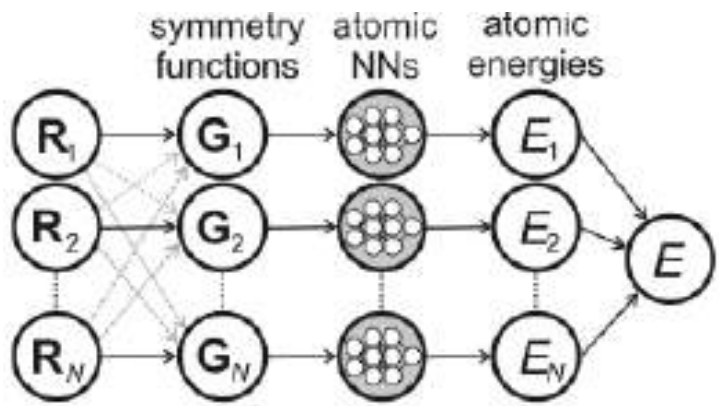
ML Atomic Properties

Advantages/applications:

- Accessing local and **atomic** physical quantities (e.g. atomic charges) and their trends
- Constructing accurate structural FP
- Expressing global quantities in terms of **atomic** contributions: Train to small molecules/cell, use for large systems

$$E = \sum_i E_i$$

$$E = E(\{q_i\}) \text{ or } E(\{\chi_i\})$$



PRL 98, (2007), PRB 92, (2015)

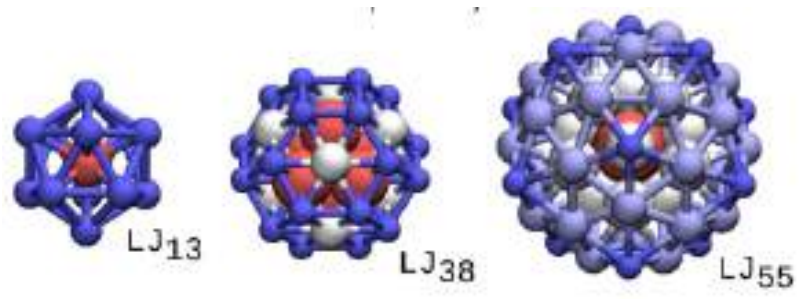
Atomic Fingerprints

- Geometric: encapsulating environmental information: Coordination #, RDF, ADF, ...

Two examples of simple scaler FPs:

- Coordination number: integer (discontinuous)
- Graph Theory inspired: Social Permutation INvarianT

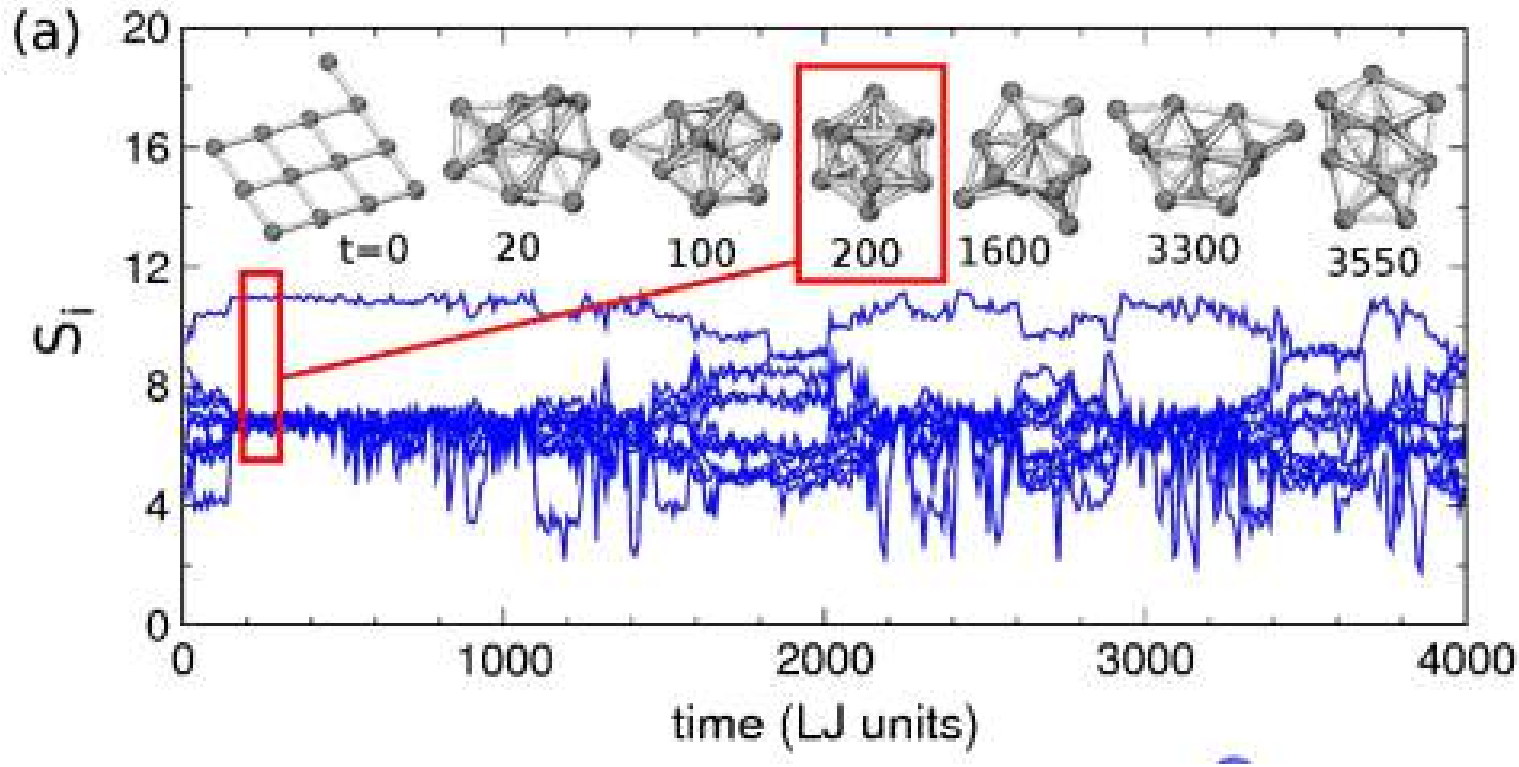
SPRINT: $S_i = \sqrt{N}\lambda_{max}\nu_i^{max}$



ν^{max} is the principal vector of the adjacency (contact) matrix

Atomic Fingerprints

SPRINT:



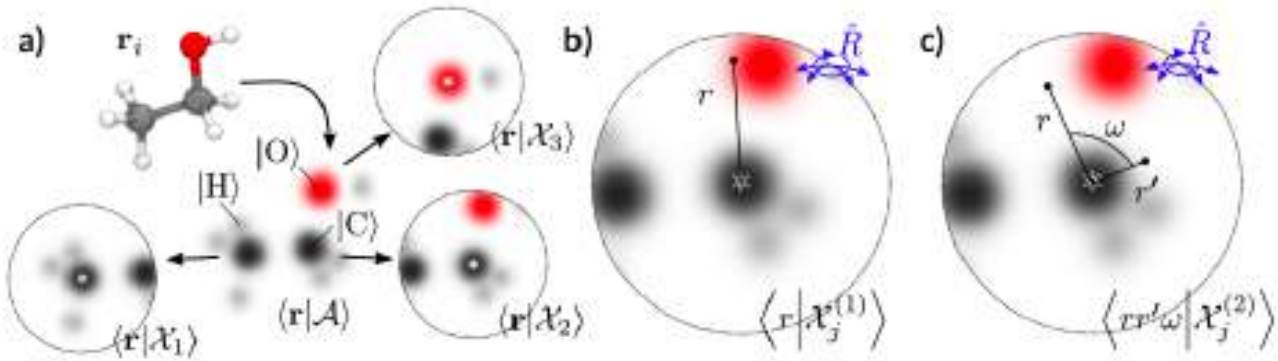
PRL 107, 085504 (2011)

Long Atomic Fingerprints

Atom-centered descriptors:

- G^2, G^5: Atom-centered Behler-Parrinello Sym. Func.
- SOAP: Smooth Overlap of Atomic Positions
- SGO: Spectrum of atom-weighted GTO's Overlap

$\mathbf{x}_i \in \mathbb{R}^{l_{\text{FP}}}$ where $L_{\text{FP}} > 1$



arXiv1807.00408

SGO

$\mathbf{x}_{\mathsf{SGO}}$ = sorted, truncated list of the largest eigenvalues of the overlap matrix of atom-centered GTO's

Atom-weighted SGO:

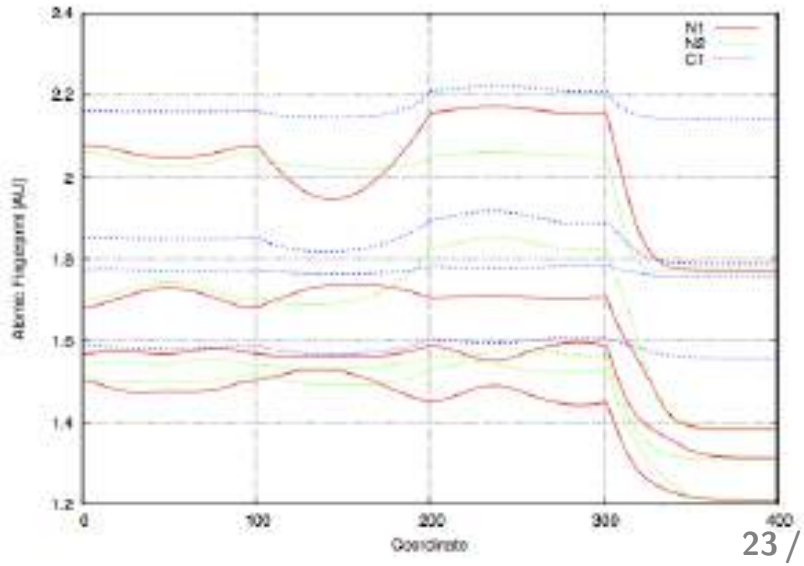
- Construct OM & cutoff matrix $f_{ij}^{k} \equiv f^{c}(r_{ik}) f^{c}(r_{jk})$
- Atom k weighted OM: element-wise multiply as $\mathsf{OM}_{ij} \times f_{ij}^{k}$
- Diagonalize; Collect l_{FP} largest eigenvalues as FP of atom k

Deforming azobenzene in 4×100 steps,

1. C-N-N-C dihedral angle
2. N-C-C angle
3. N-N-C-C dihedral angle
4. N-N bond length

Changes in $2, 3, \ldots N$-body characters.

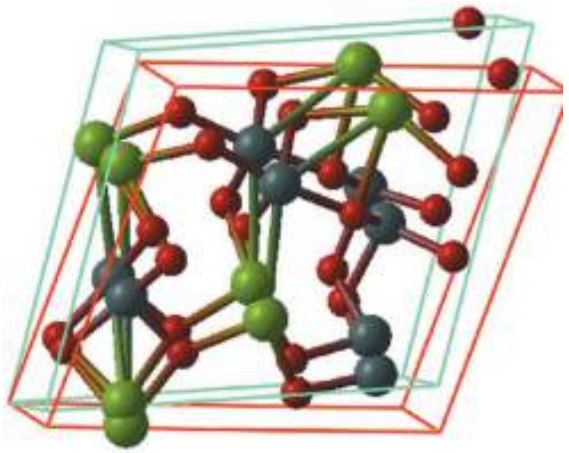
Prediction of atomic charges



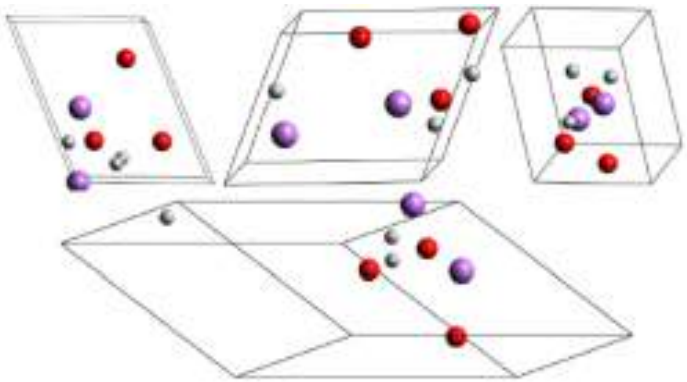
SGO FP for Crystals

Crystal

$(\mathbf{Z}, \mathbf{R}, \mathbf{a})$ determines the crystal structure



Comparing crystal structures needs considering both lattice vectors and atomic positions



4 distinct descriptions of the same structure

(Com Phys Commun 183.3 (2012))

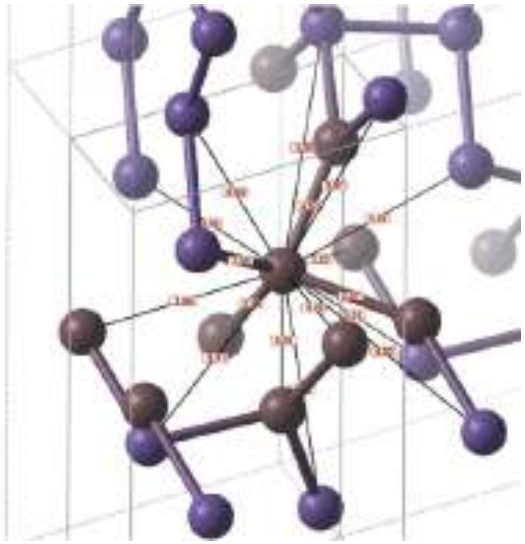
No unique representation of crystalline structure $\Rightarrow$ One should avoid comparing lattice vectors.

Crystal Fingerprints

Hint!

- Exclude lattice vectors from FP construction
- Take a spherical piece centered on each atom

A per-atom histogram can be served as structure fingerprint
$\Rightarrow$ Comparing histograms or RDF
Sensitive to parameters (bin widths) Acta Cryst. (2010) A66



SGO Crystal FP

Extending the cluster SGO FP to crystal case

Golden idea

Crystal $\Rightarrow$ a set of clusters

- Take a spherical piece centered on each atom in the unitcell
- Construct OM & cutoff matrix $f_{ij}^{k} \equiv f^{c}(r_{ik}) f^{c}(r_{jk})$
- Atom k weighted OM: element-wise multiply as $\mathrm{OM}_{ij} \times f_{ij}^{k}$
- Contract the matrix
- Diagonalize; Collect l_{FP} largest eigenvalues as FP of atom k

JCP 144, (2016)

Outline

Summary

- Arrangement of atoms determines the properties of materials
- Atomic descriptors can be constructed from local information
- Including multi-atomic character is possible via matrix methods
- Efficiency of ML properties depends critically on the quality of atomic descriptors

Thanks for your attention!

Outline

Hands-on Session

- Visualize the trajectory of the structure: VMD
- Generate atomic fingerprints:
  - Compile the code fingerprint.x
  - Run it and take the generated file
- Shuffle the data set and split it to 3 parts: train, validate, test
- Run the ckrr.x code and get the error verification on train and validate sets
- Scan the hyper-parameter space
- Take the best hyper-parameters, then run on the test set
- Run the plot.sh script to generate the scattering plots