UNIVERSITY OF NOTRE DAME

Neural Network Model Chemistries

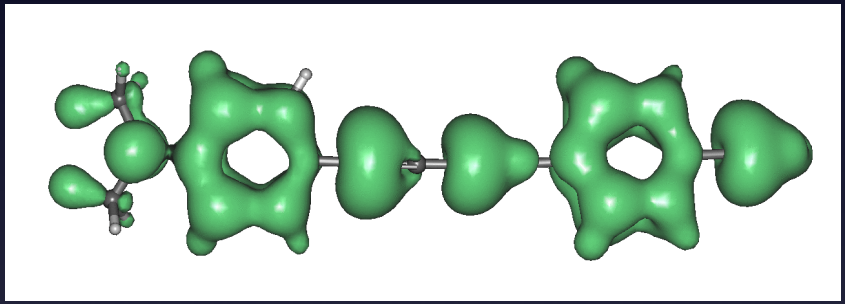
RIKEN 3/25/2017

John Parkhill
Department of Chemistry and Biochemistry, Notre Dame

Parkhill Group

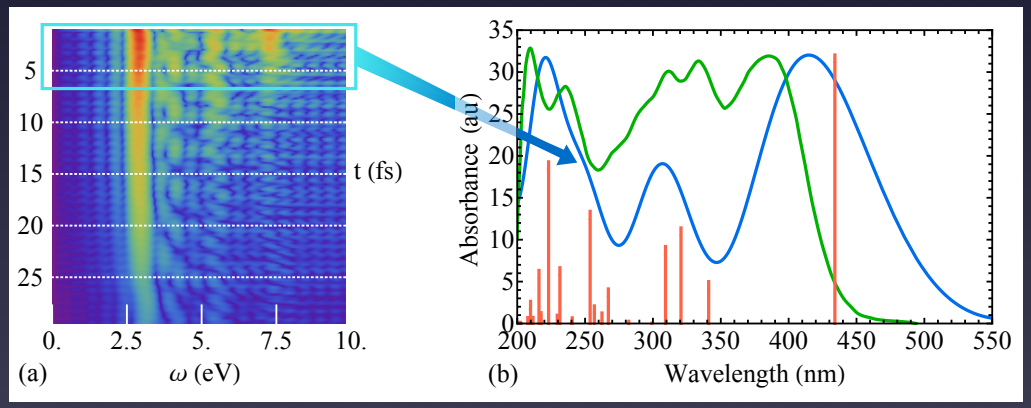
5 Students (K Yao)

Research Areas:

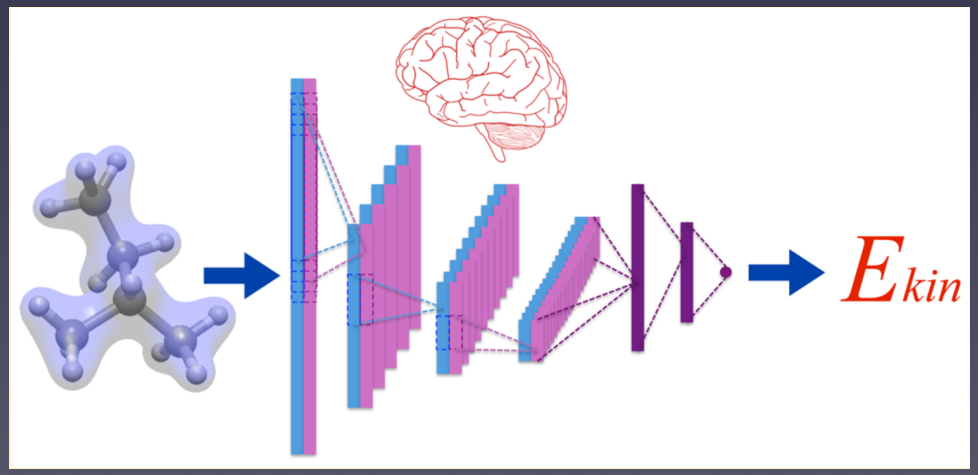


Non-equilibrium realtime quantum electronic dynamics.

$$\frac{d}{dt}\gamma_p = \frac{1}{2}\{\gamma_s\gamma_t V(s)^{pr}_{st} V(t)^{st}_{pr}\eta_p\eta_r + \gamma_s\gamma_t V(s)^{st}_{pr} V(t)^{pr}_{st}\eta_p\eta_r$$
$$-\gamma_p\gamma_t V(s)^{pt}_{rs} V(t)^{rs}_{pt}\eta_r\eta_s - \gamma_p\gamma_t V(s)^{rs}_{pt} V(t)^{pt}_{rs}\eta_r\eta_s\}$$

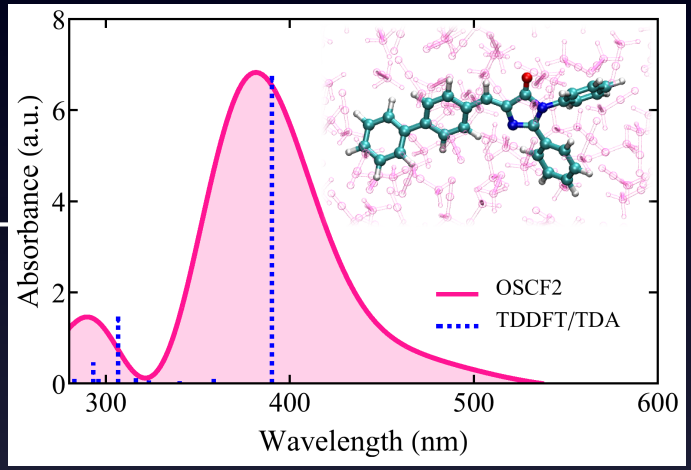


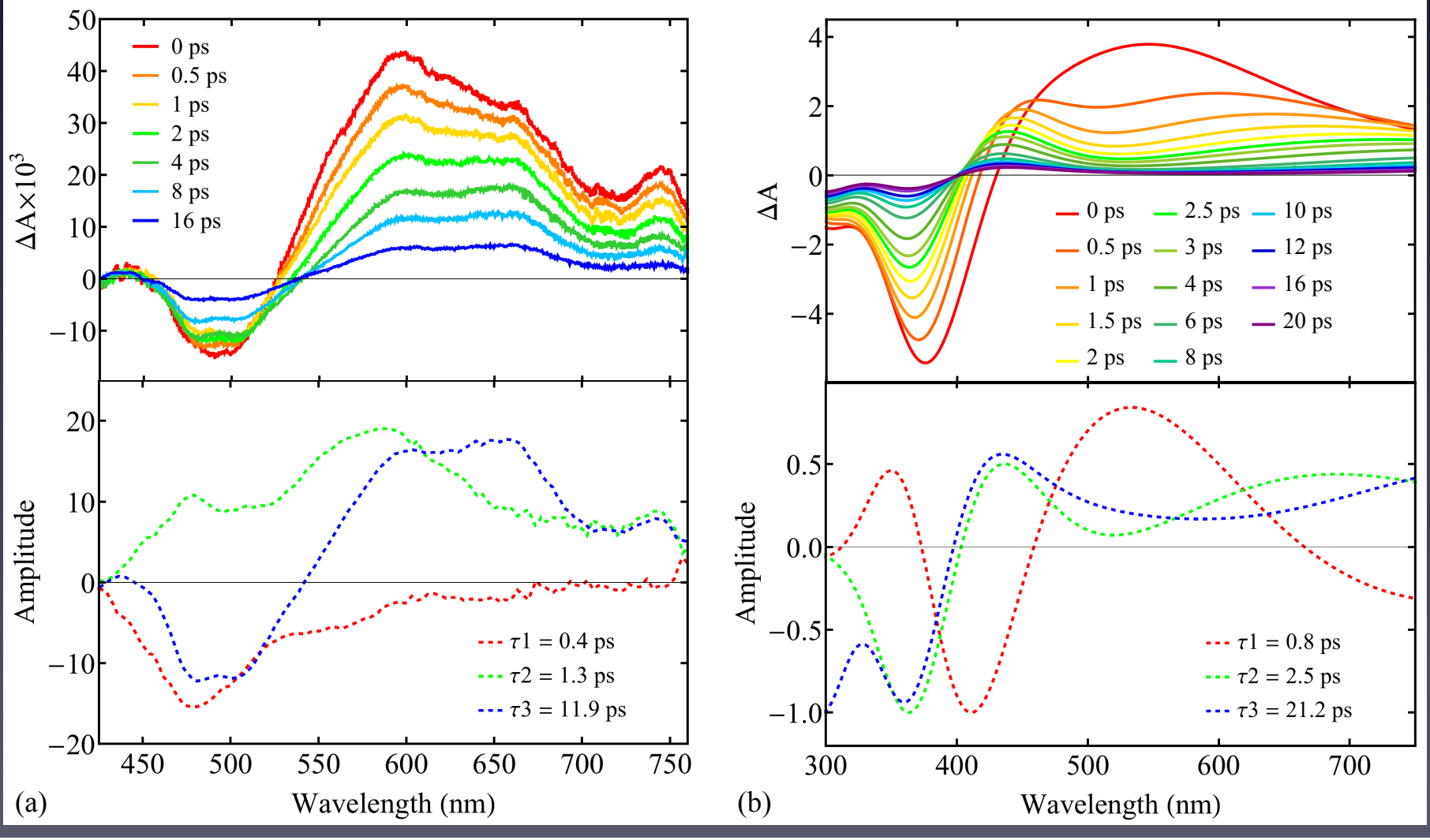
Applications of Neural Networks to electronic structure theory.



Realtime Spectra

Realtime Transient Abs. spectra
(Expt, Vauthey left)



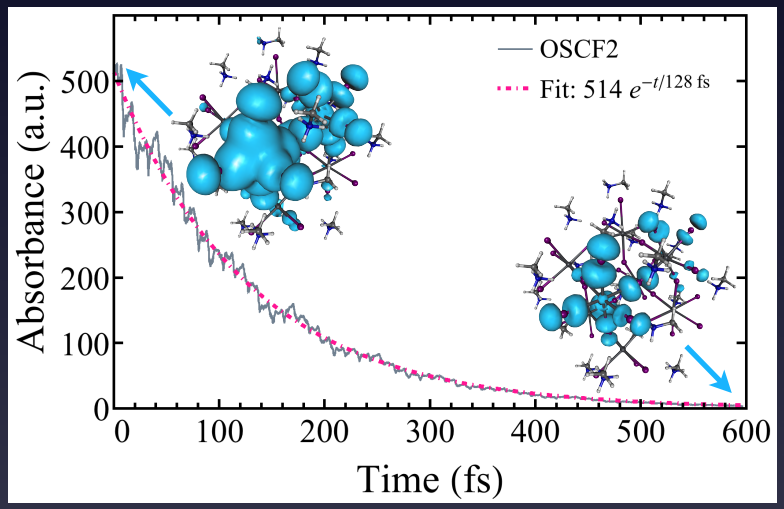


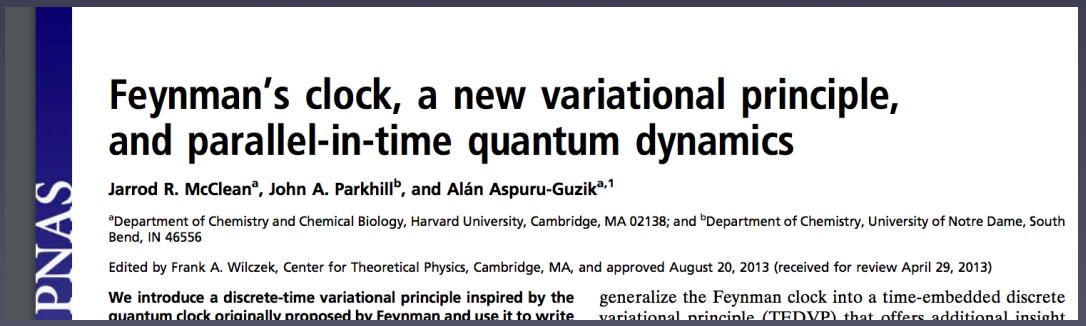
Begging for Breakthroughs

Electronic timescale
~ $4*10^{-4}$ fs

1ps = 10^7 fock builds
114 days at 1 build/second

Dynamics is sequential
and parallelization in time
is weak.





PNAS

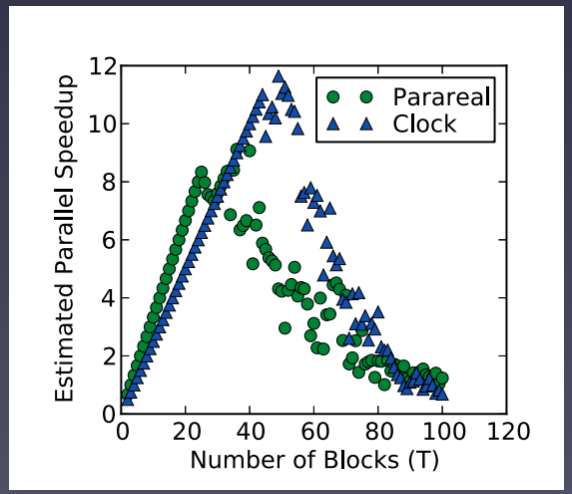
Feynman's clock, a new variational principle, and parallel-in-time quantum dynamics

Jarrod R. McClean[a], John A. Parkhill[b], and Alán Aspuru-Guzik[a,1]

[a]Department of Chemistry and Chemical Biology, Harvard University, Cambridge, MA 02138; and [b]Department of Chemistry, University of Notre Dame, South Bend, IN 46556

Edited by Frank A. Wilczek, Center for Theoretical Physics, Cambridge, MA, and approved August 20, 2013 (received for review April 29, 2013)

We introduce a discrete-time variational principle inspired by the quantum clock originally proposed by Feynman and use it to write generalize the Feynman clock into a time-embedded discrete variational principle (TEDVP) that offers additional insight



3 Models

- Orbital Free DFT
- Neural Networks + Many Body Expansion.
- A Diatomics in Molecules NN

Today

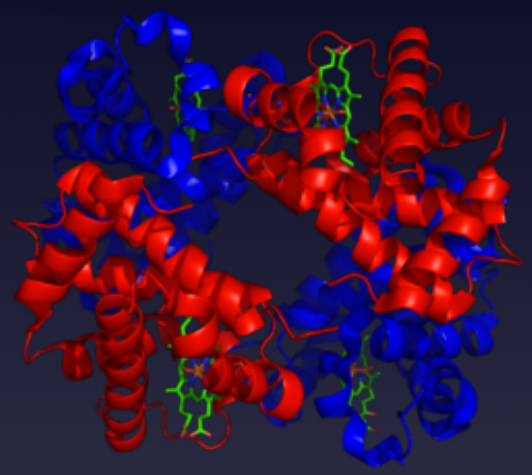
2000+ Atoms	<100 Atoms	<40 Atoms
Force-Fields	Density Functional	Ab-Initio

Soon

2000+ Atoms	<100 Atoms	<40 Atoms
Force-Fields	Density Functional	Ab-Initio

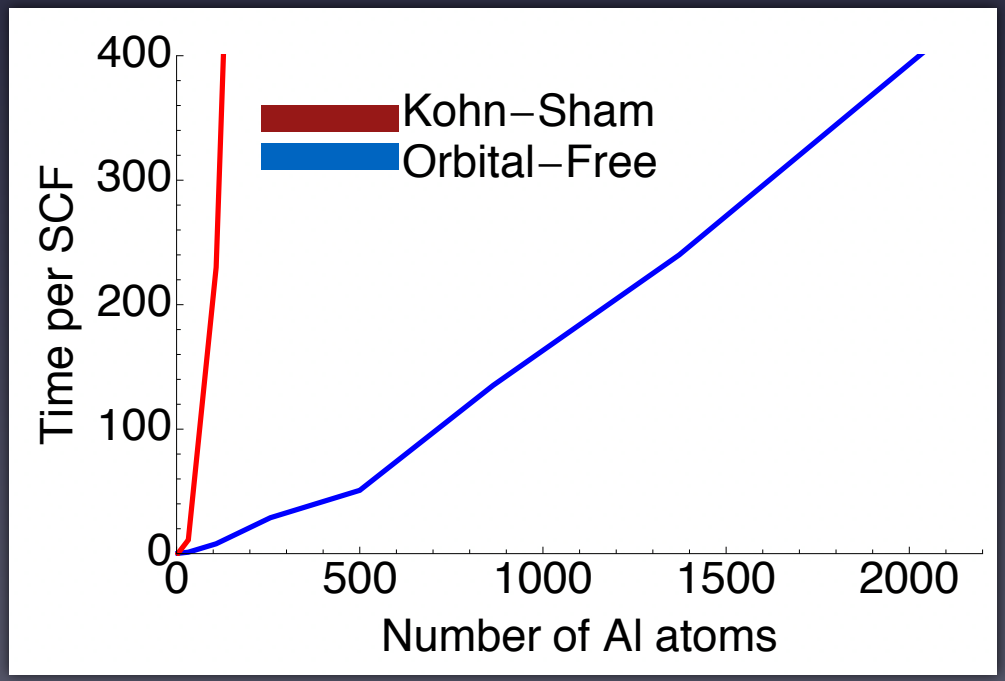
Neural Network

Orbital Free DFT



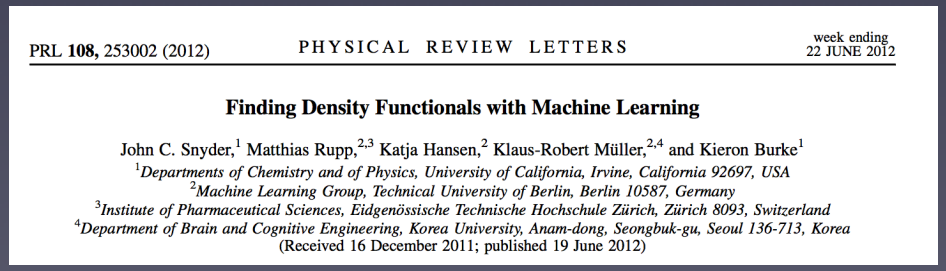
Hemoglobin : 16000 Daltons
~ 16000 orbitals vs Limit of Most KS ~ 3000

With orbital free-DFT you only need 1 orbital, and get a 10x speedup.



But you must know a mysterious 'functional' which maps the density to kinetic energy....

$$E_{\text{kin}} = \int T(n(r))dr$$



PRL **108,** 253002 (2012) PHYSICAL REVIEW LETTERS week ending 22 JUNE 2012

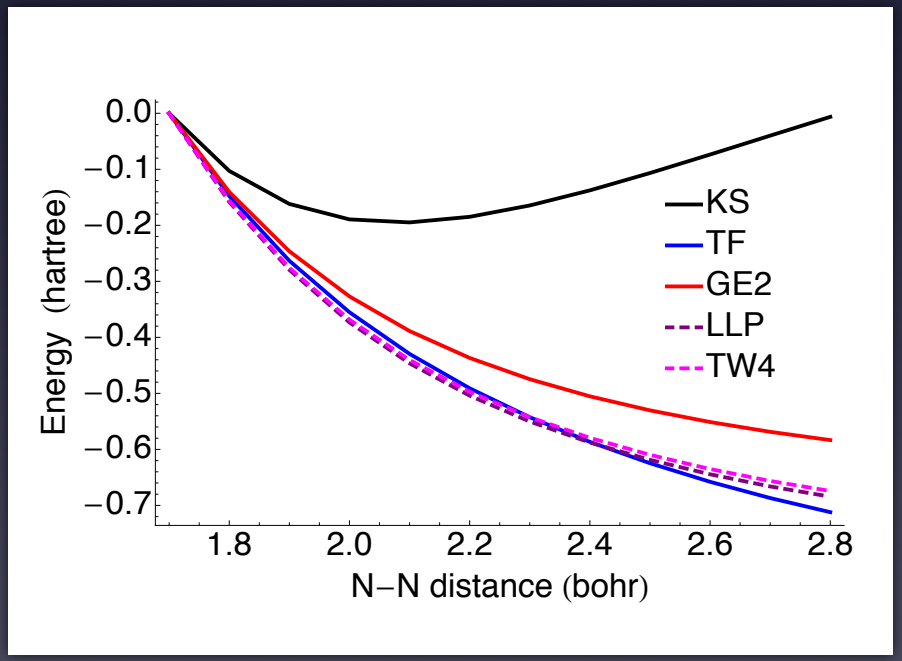
Finding Density Functionals with Machine Learning

John C. Snyder,[1] Matthias Rupp,[2,3] Katja Hansen,[2] Klaus-Robert Müller,[2,4] and Kieron Burke[1]

[1]*Departments of Chemistry and of Physics, University of California, Irvine, California 92697, USA*
[2]*Machine Learning Group, Technical University of Berlin, Berlin 10587, Germany*
[3]*Institute of Pharmaceutical Sciences, Eidgenössische Technische Hochschule Zürich, Zürich 8093, Switzerland*
[4]*Department of Brain and Cognitive Engineering, Korea University, Anam-dong, Seongbuk-gu, Seoul 136-713, Korea*
(Received 16 December 2011; published 19 June 2012)

Chemistry Starts in the 4th digit.

$$T_{\mathrm{GGA}} = \int \tau_{\mathrm{TF}}(n(r)) F\left(\frac{|\nabla n(r)|}{n(r)^{3/4}}\right)$$



Accuracy ~ 1 %

No shell structure

<u>No Bonding !</u>

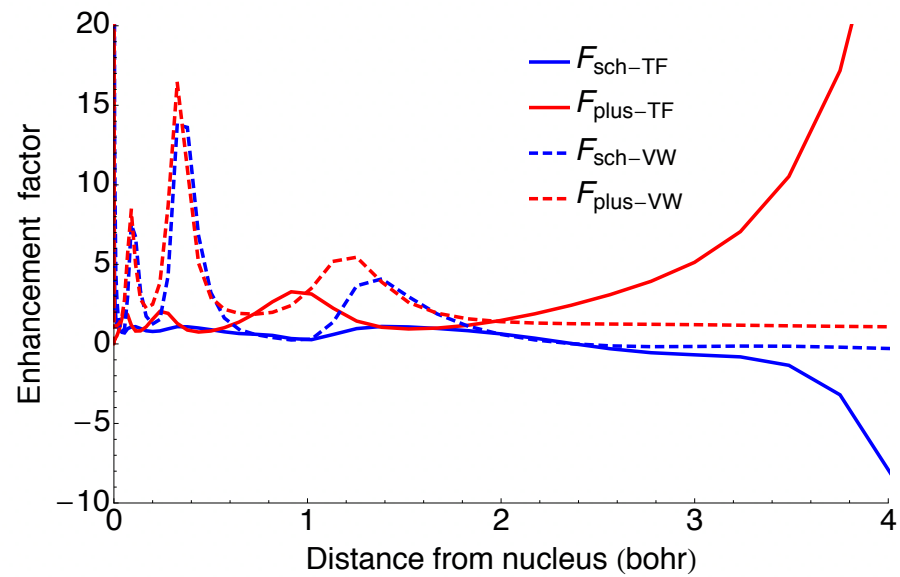
D Garcia-Aldea, JE Alvarellos *Journal of chemical physics* 127.14 (2007): 144109.

A Kohn-Sham Kinetic Energy Density

$$T = \int F(\rho(r), \rho(r_1), \rho(r_2), \cdots) t_{TF(VW)}(\rho; r) dr$$

Local kinetic energy $t_{plus} = \frac{1}{2}\sum_{i=1}^{N} |\nabla\phi_i(r)|^2$ $\quad t_{sch} = -\frac{1}{2}\sum_{i=1}^{N} \phi_i^*(r)\nabla^2\phi_i(r)$

Thomas Fermi and Von Weisacker $\quad T_{TF} = \int C_{TF}\rho^{5/3} dr \quad T_{VW} = \frac{1}{8}\int \frac{|\nabla\rho|^2}{\rho} dr$

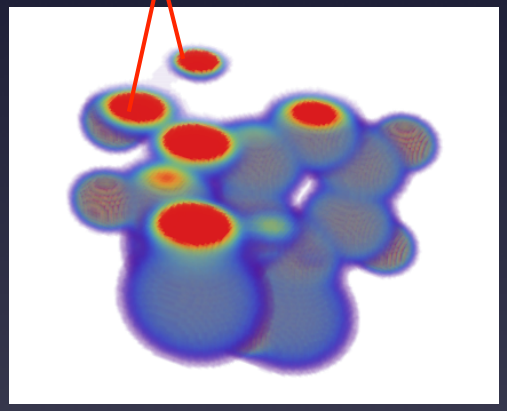


- Four types of F
- They all display the shell structure
- TF based F diverge at long distance, while VW based converge

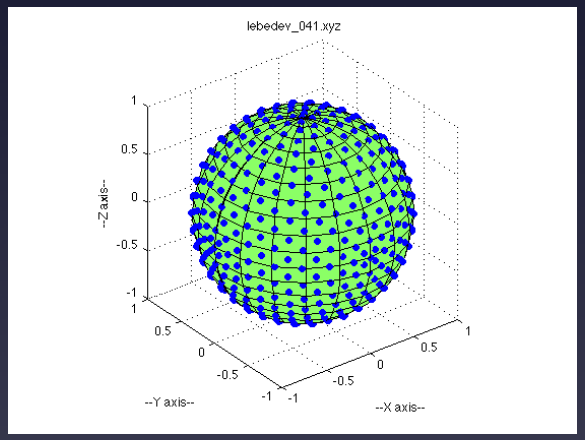
CNN version 0.1

Pseudo 2-d input motivated by computational limitations

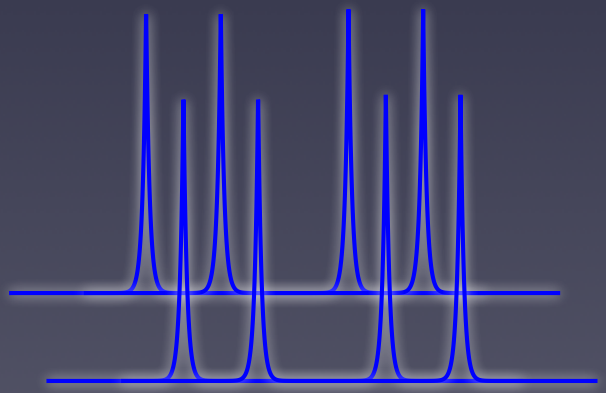
quadrature point



Density along lines fed into convolutional neural network.



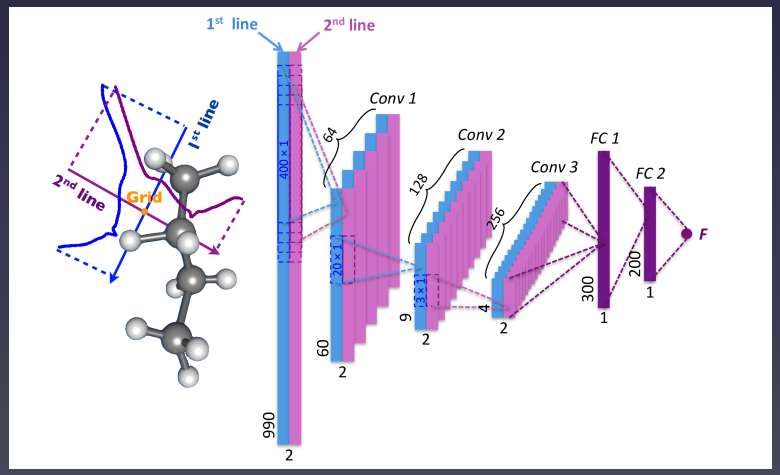
~1 million quadrature points per atom.
~2000 inputs per quadrature point.
~Barely tractable



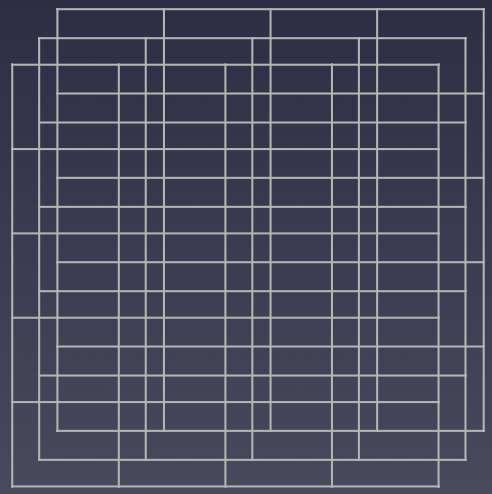
Our Network

$$T\{n(r)\} = \int F\{n(r), r'\}\tau_{tf}(r')dr' \qquad \tau_{TF} = \frac{3}{10}(3\pi^2)^{5/3}n(r)^{2/3}dr$$

~4000 samples per grid point
10^6 samples per atom



Future thoughts:
3D Convolutional Networks
Basis sets

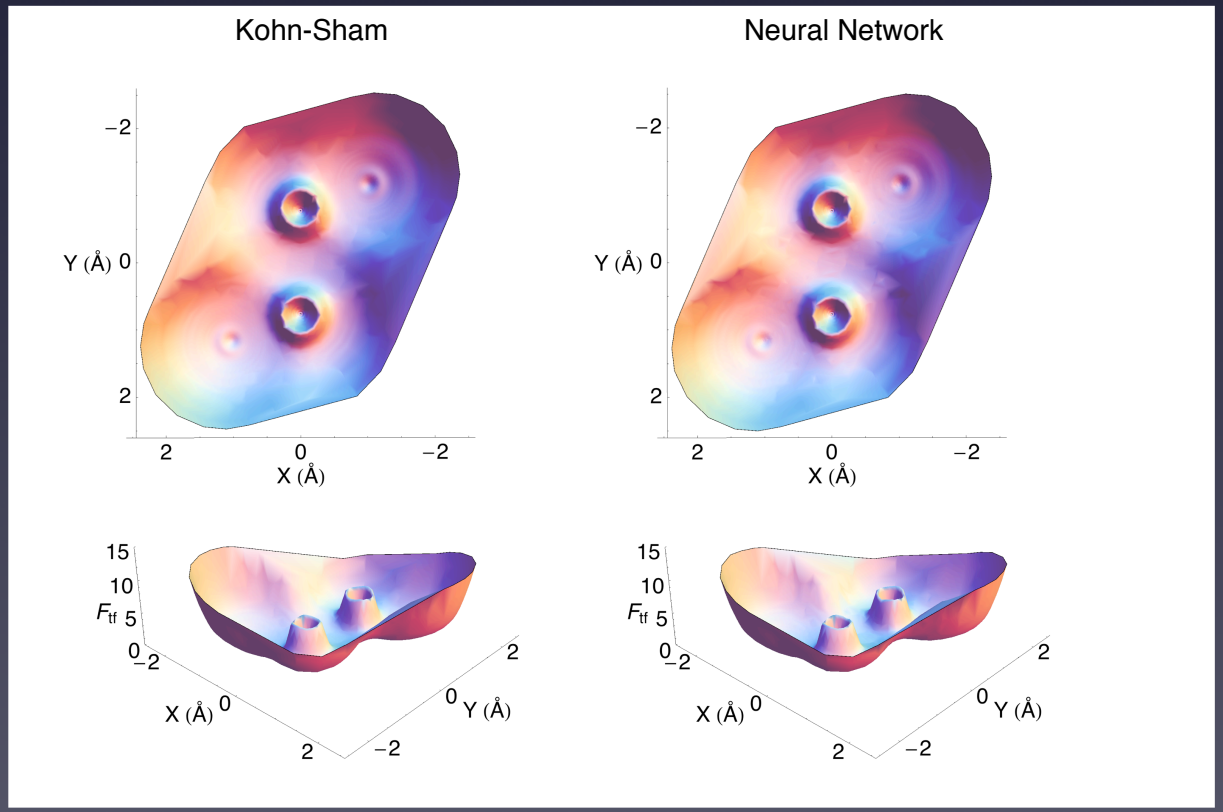


$$y^i_{mn} = f(b^i + \sum_{a=0}^{a=p}\sum_{b=0}^{b=q} y^{i-1}_{(m+a)(n+b)} w^{i-1,i}_{ab}) \qquad f(x) = \max(0, x)$$

Finding a functional

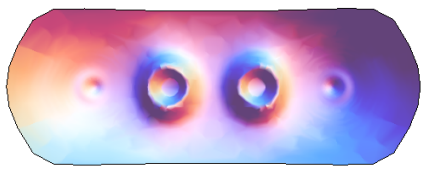
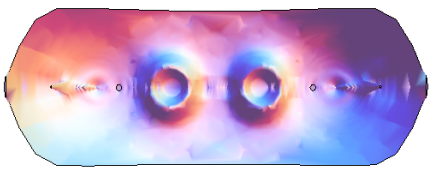
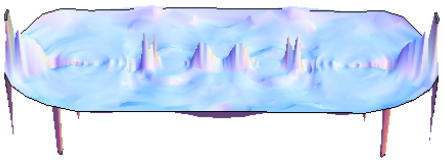
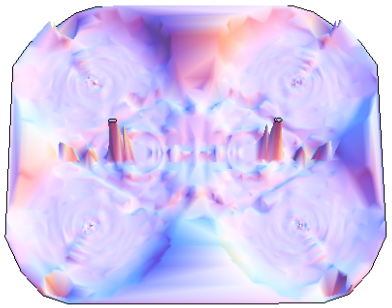
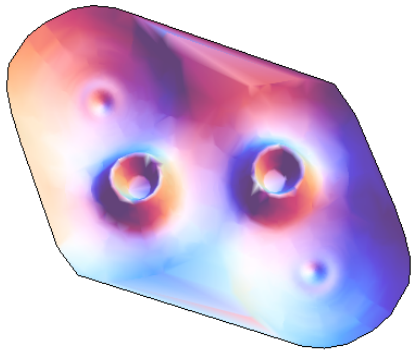
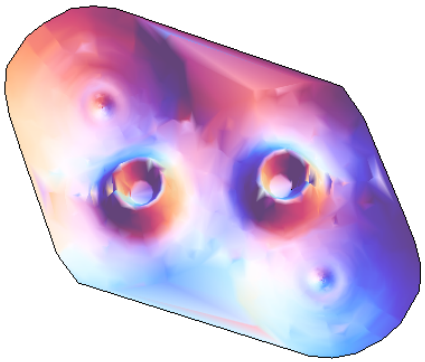
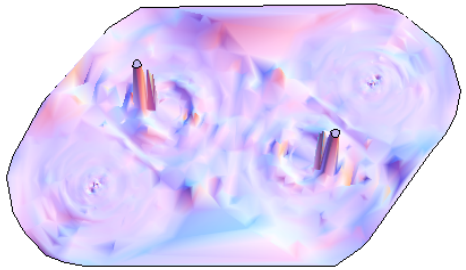
Learn F as a function of n(r), given as a slice of the surface.

$$T\{n(r)\} = \int F\{n(r), r'\}\tau_{tf}(r')dr'$$

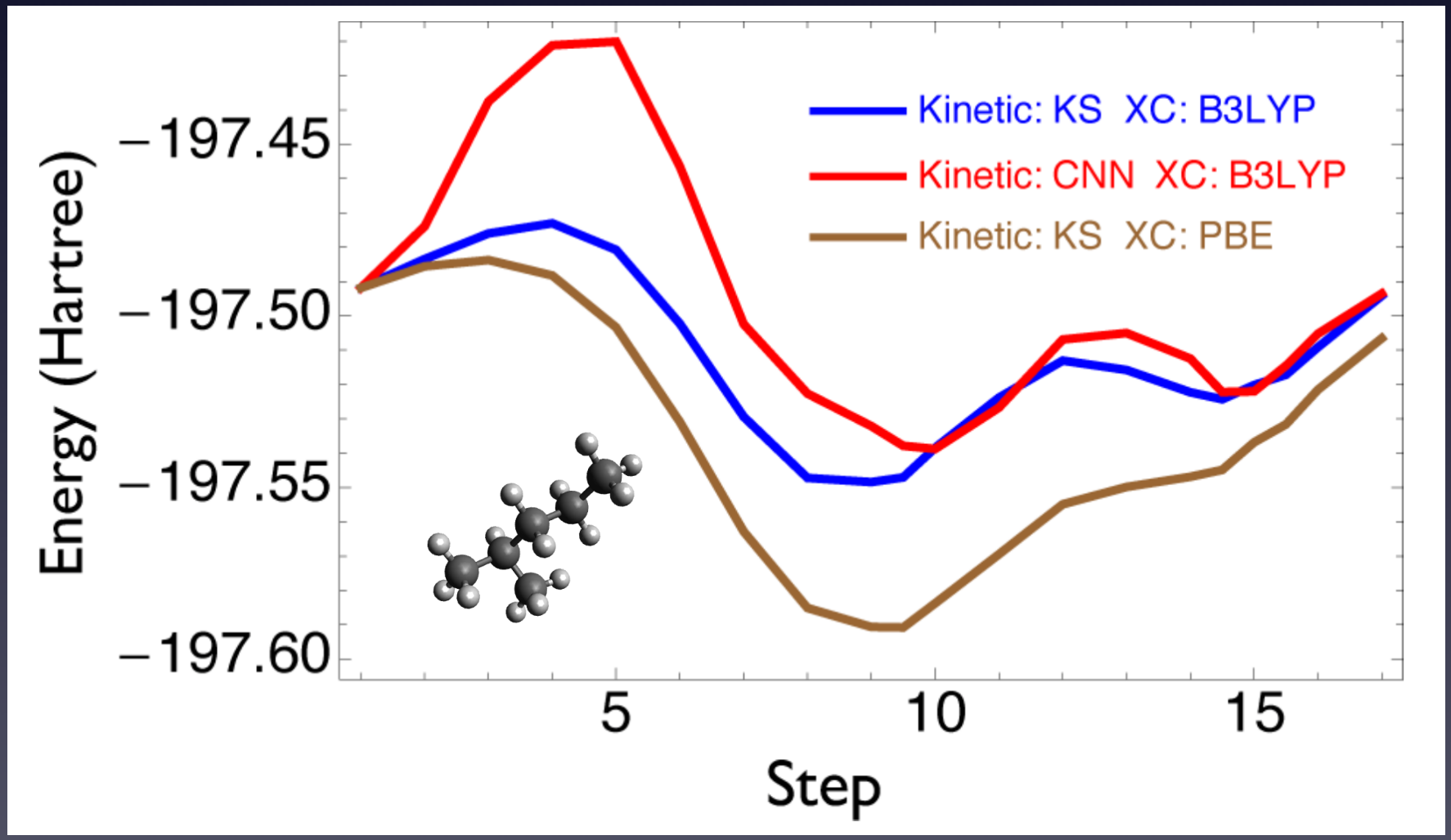


The shape of the error

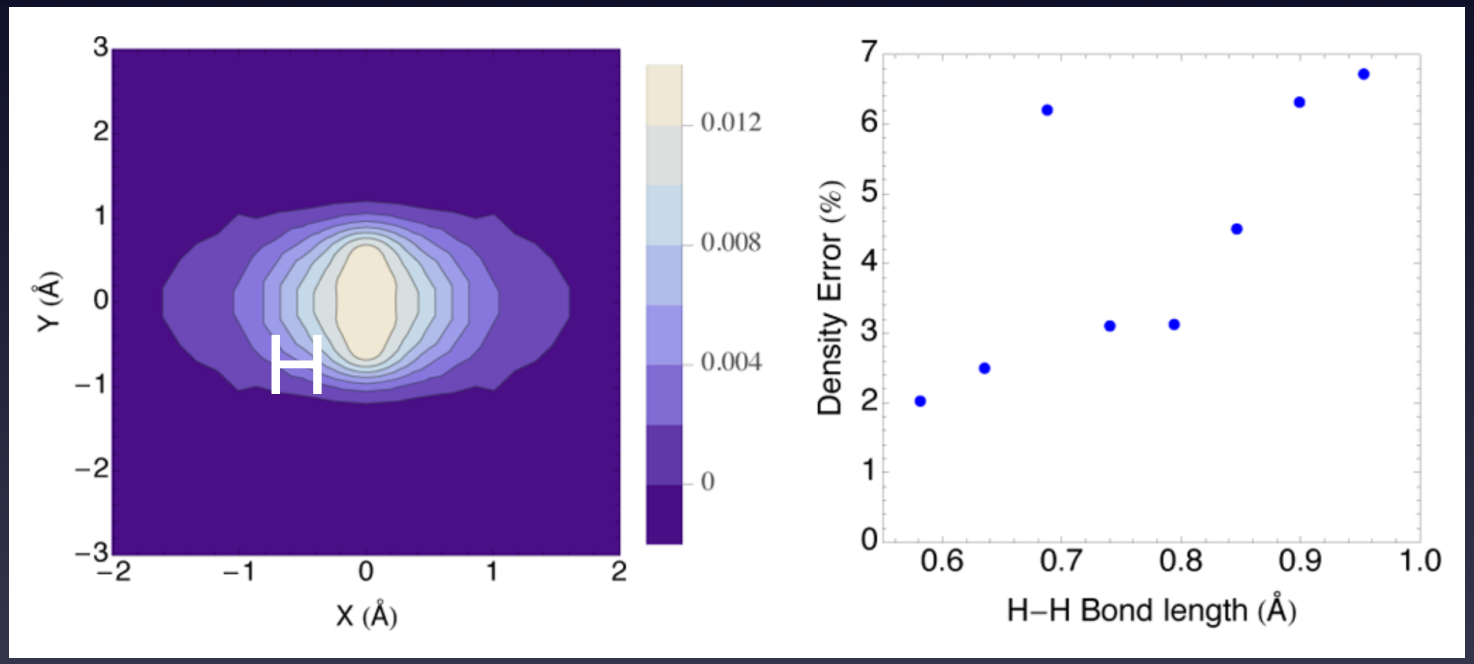
Enhancement Factor in C-C bonding plane

	Accurate (KS)	Neural Net	Error(NN-KS)
C_2H_2			
C_2H_4			
C_2H_6			

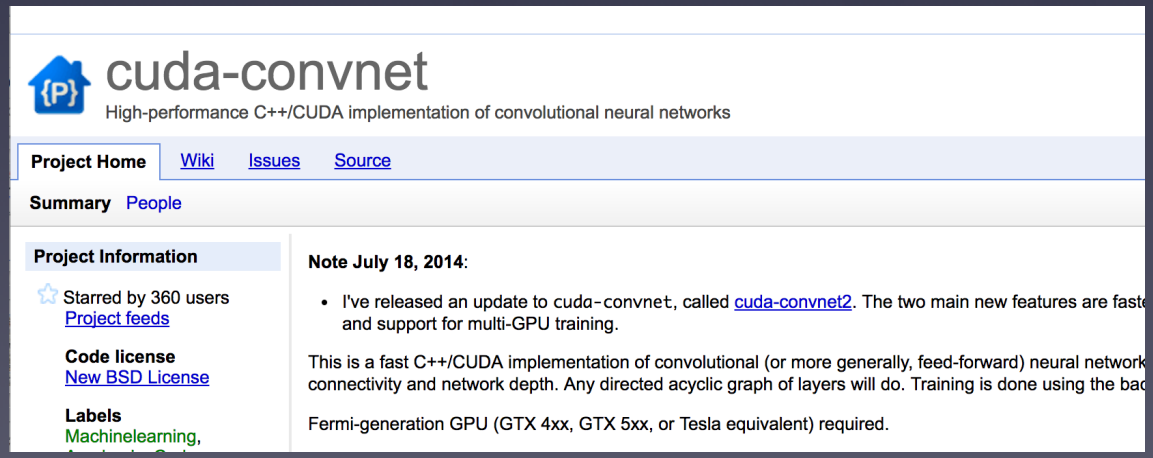
Reproducing Potential Energy Surfaces



Self-Consistent Densities.



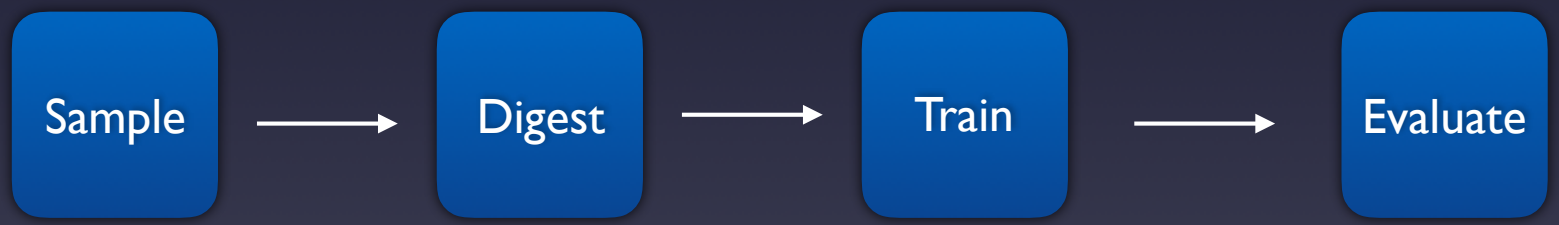
Errors of the NN's optimal density in Hydrogen

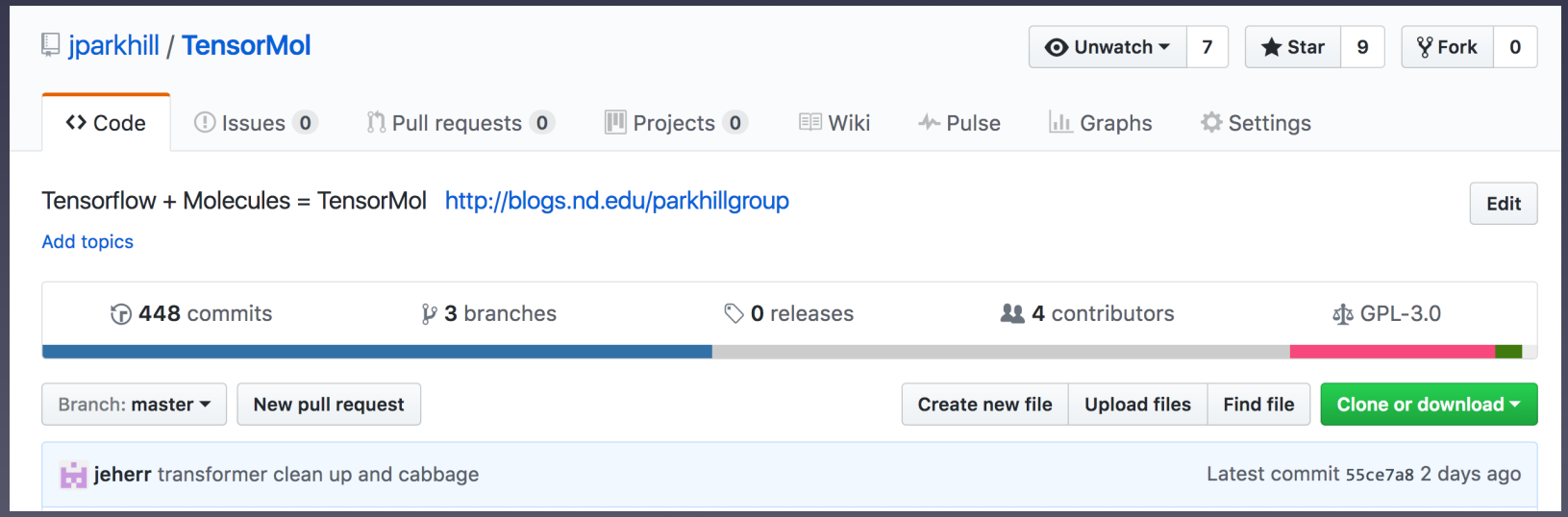


Getting Serious.

- Better Density inputs.
- Gradients (which require tight integration between electronic structure and the NN.
- Some architecture for training data

PYSCF TensorMol





What is TensorMol

A set of chemical routines
(90% python 10% C++) on top of TensorFlow ™

Capabilities:

- Behler-Parrinello, Many Body etc.
- Various network types (FC, Convolutional, 3d)
- A variety of digesters: (Coulomb, Symmetry Functions, Radial*Spherical Harmonics)
- A variety of outputs (energy, force, probability)
- Some gradients.
- Integration with PYSCF for Ab-initio energies, Coulomb integrals etc.

A Heretical Model

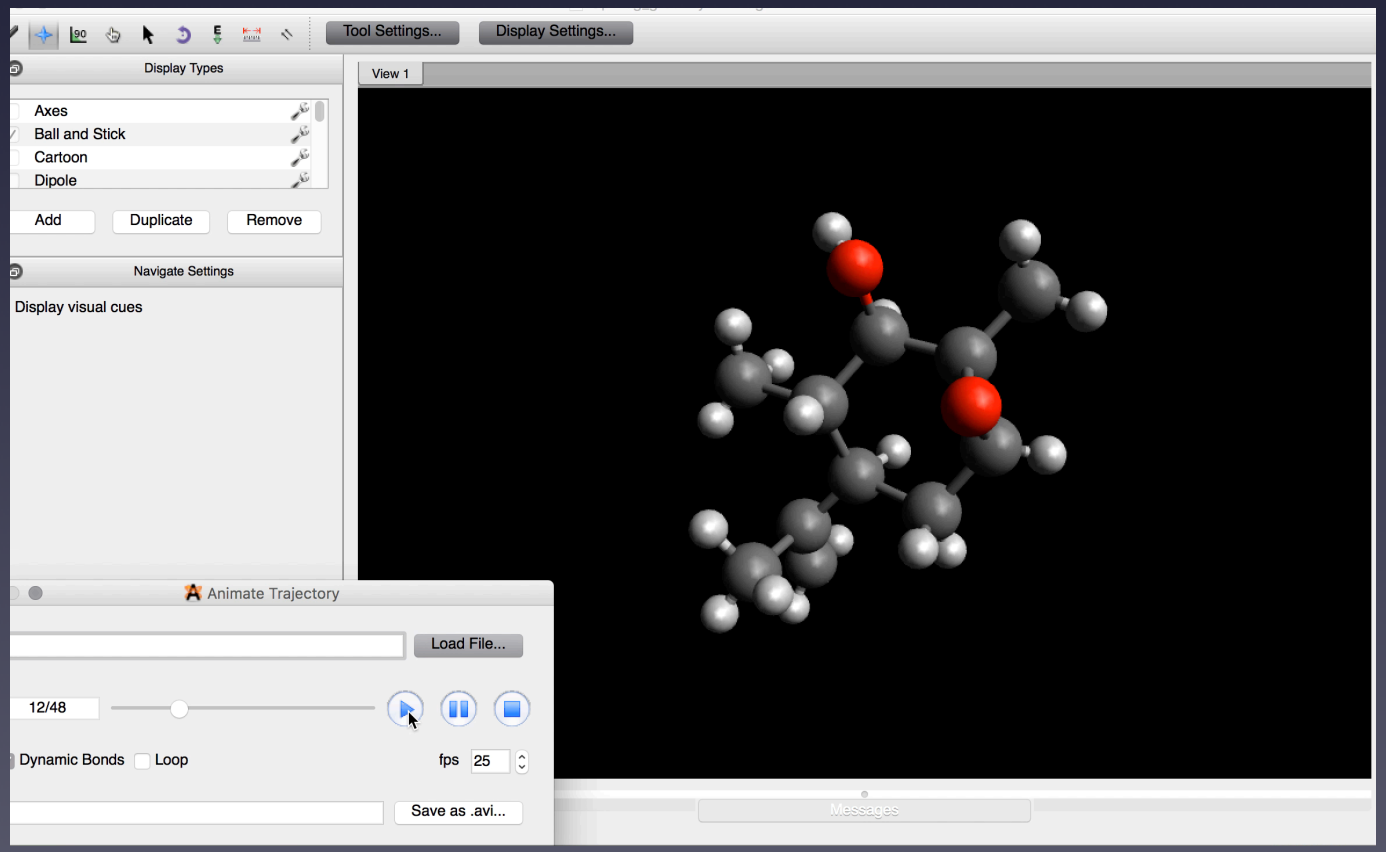
Take some crystal structures, define a Gô type potential. Sample its normal modes, and learn its force. Then optimize other molecules

```
a=MSet("OptMols")
a.ReadXYZ("OptMols")
c=a.DistortedClone(60)
b=a.DistortAlongNormals()
TreatedAtoms = b.AtomTypes()
# 2 - Choose Digester
d = Digester(TreatedAtoms, name_="GauSH",OType_ ="Force")
# 4 - Generate training set samples.
tset = TensorData(b,d)
tset.BuildTrain("OptMols_NEQ",TreatedAtoms) # generates dataset numpy arrays for each atom.
tset2 = TensorData(c,d)
tset2.BuildTrain("OptMols_NEQ",TreatedAtoms,True) # generates dataset numpy arrays for each atom.
tset = TensorData(None,None,"OptMols_NEQ_GauSH",None,6000)
manager=TFManage("",tset,True,"fc_sqdiff") # True indicates train all atoms

test_mol = a.mols[0]
print "Orig Coords", test_mol.coords
test_mol.Distort()
print test_mol.coords
print test_mol.atoms
manager=TFManage("OptMols_NEQ_GauSH_fc_sqdiff",None,False)
optimizer  = Optimizer(manager)
optimizer.Opt(test_mol)
```

A Heretical Model

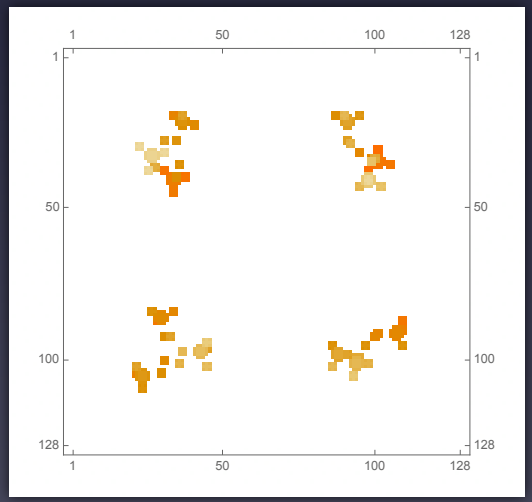
Take some crystal structures, define a Gô type potential. Sample its normal modes, and learn its force. Then optimize other molecules



Physical Inputs with Invariance

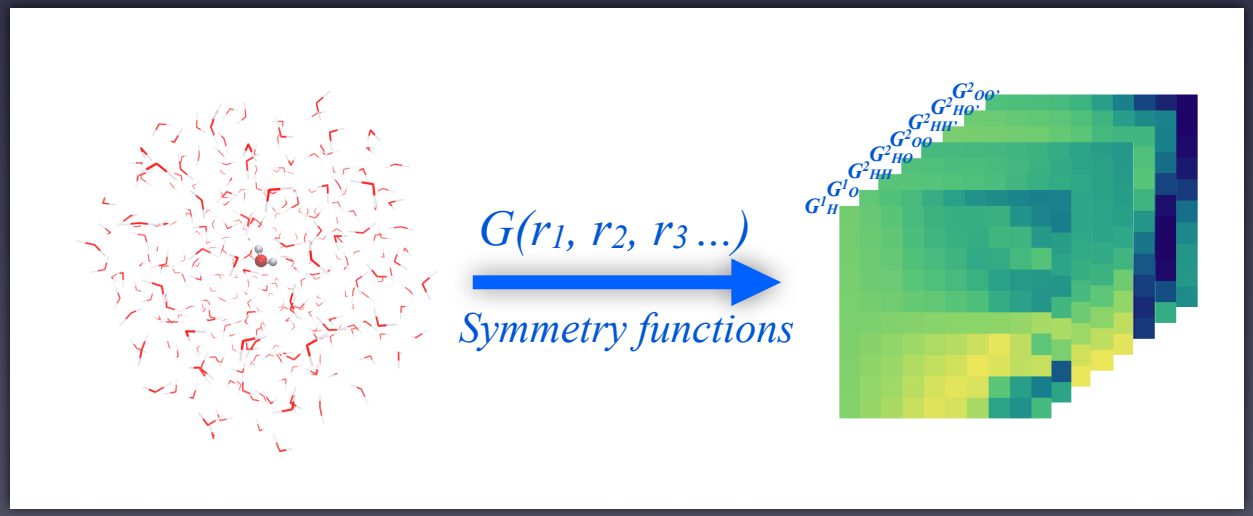
How to parameterize a molecule with invariances and retain invertibility?
How to express atomic number differences avoiding separate channels.

Depth Map



Coulomb Matrix?
Sorted by distance or atomic number?

Symmetry Functions?

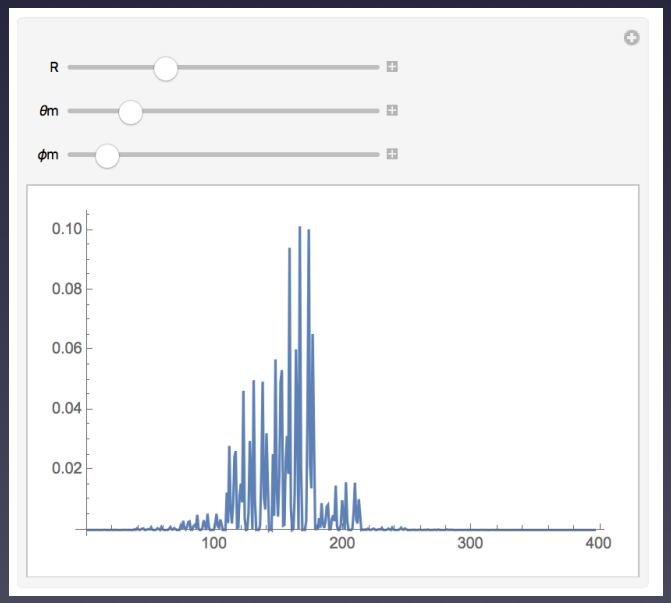


Physical Inputs with Invariance

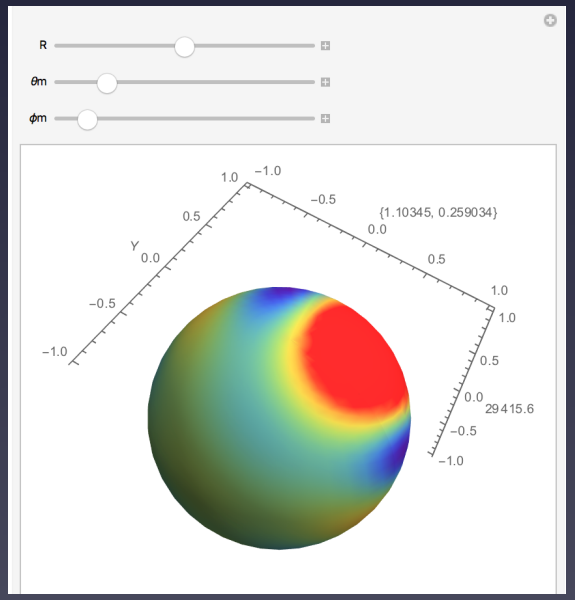
$$f_{nlm}(x, y, z) = R_n Y_{l,m}$$

$$R_n = e^{-(r-r_0)^2/(2\sigma^2)}$$

Embedded Space

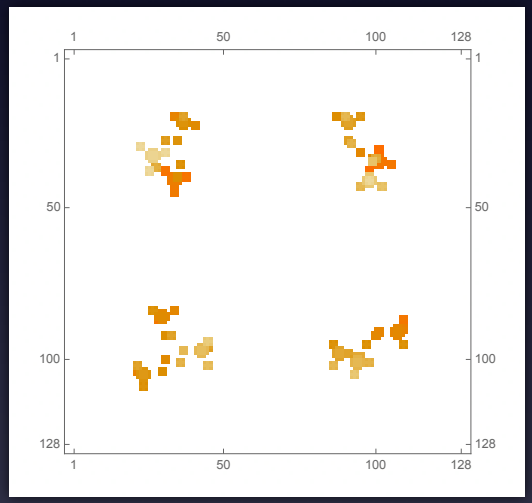


Real Space



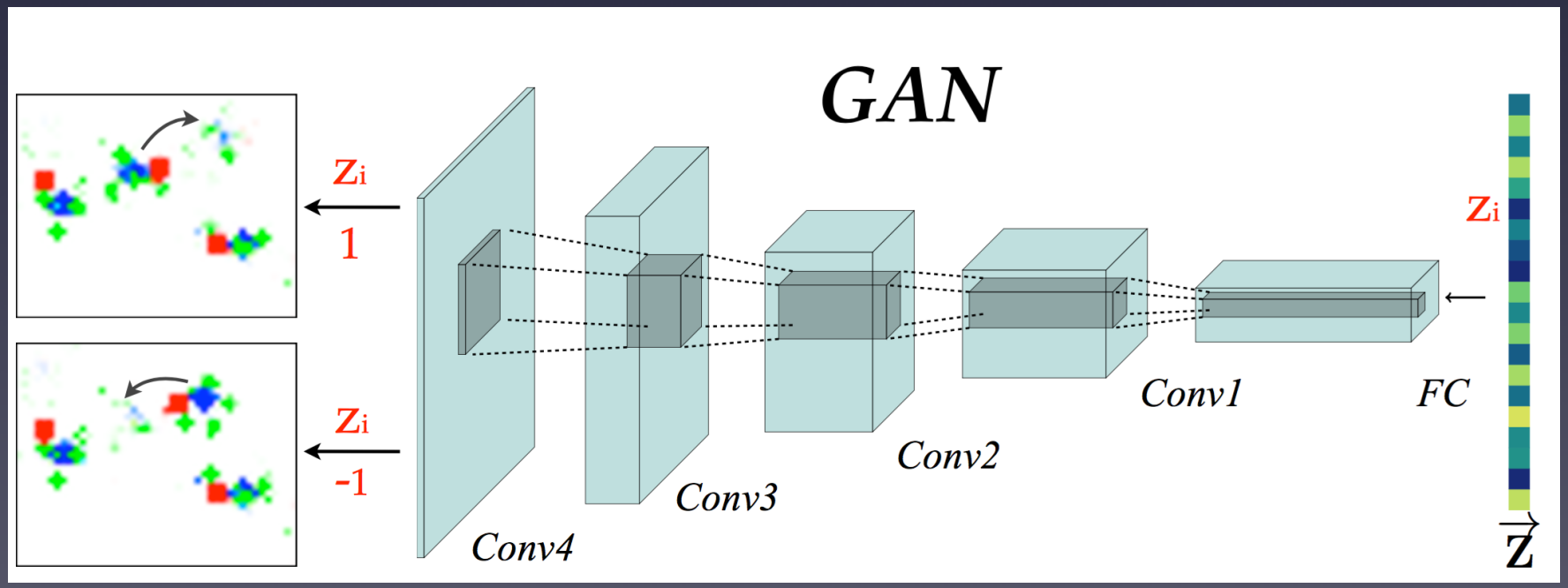
This embedding is reversible! can go between geometry and embedded geometry reversibly.

Generative Adversarial models.

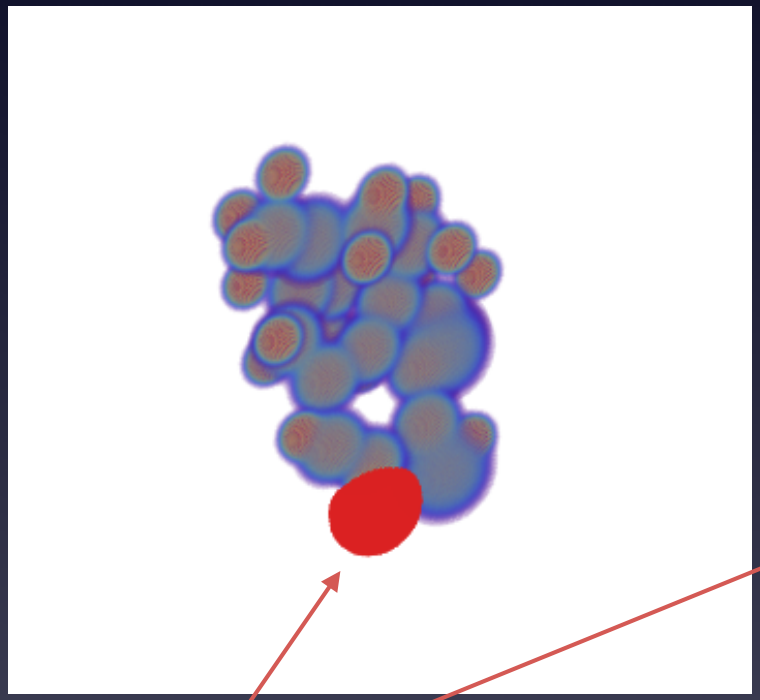


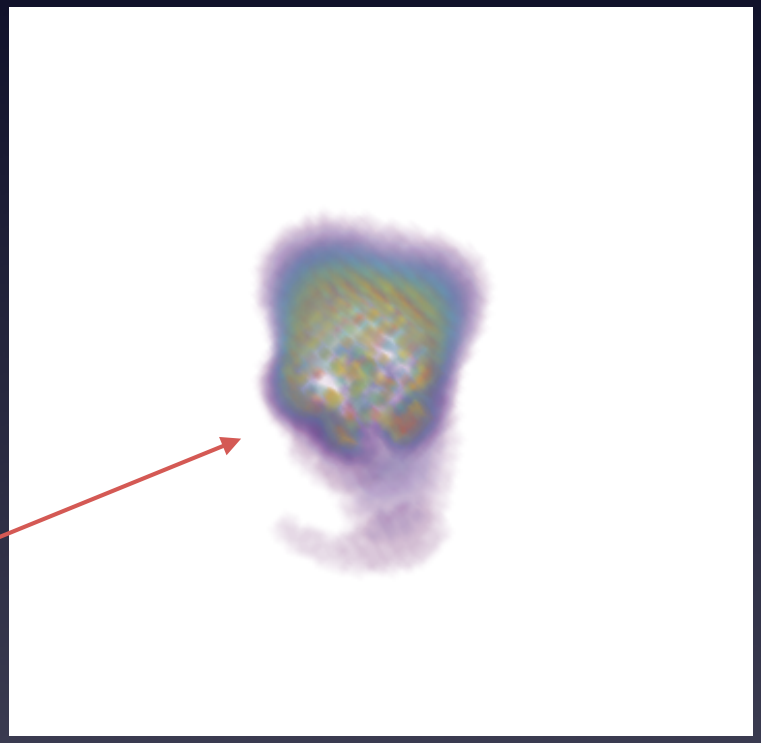
Depth of field map for an MD trajectory of 3 methanols

A way to create a set of nonlinear modes to sample chemical space.



My personal favorite





Embedding for this atom

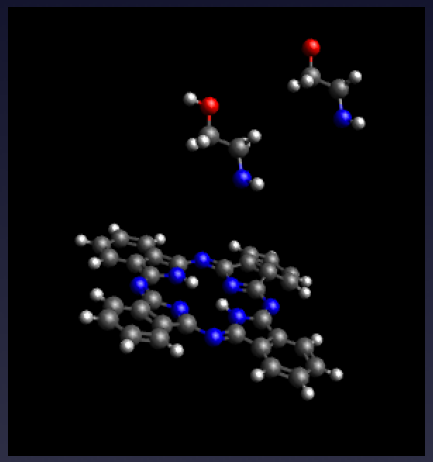
$$|rlm\rangle = \sum_{\text{atoms}} f_r(x,y,z) Y_m^l(x,y,z) * (Atm.Number)$$

Partitioning of the energy.

Behler-Parinnello

$$E = \sum_{\text{Atoms}} E_{\text{atom}}$$

Back propagates atom networks for each element with only 1 energy



Many Body Expansion

$$E = \sum_{\text{Molecules}} E_{\text{mol}} + \sum_{\text{pairs}} E_{\text{pair}} + \ldots$$

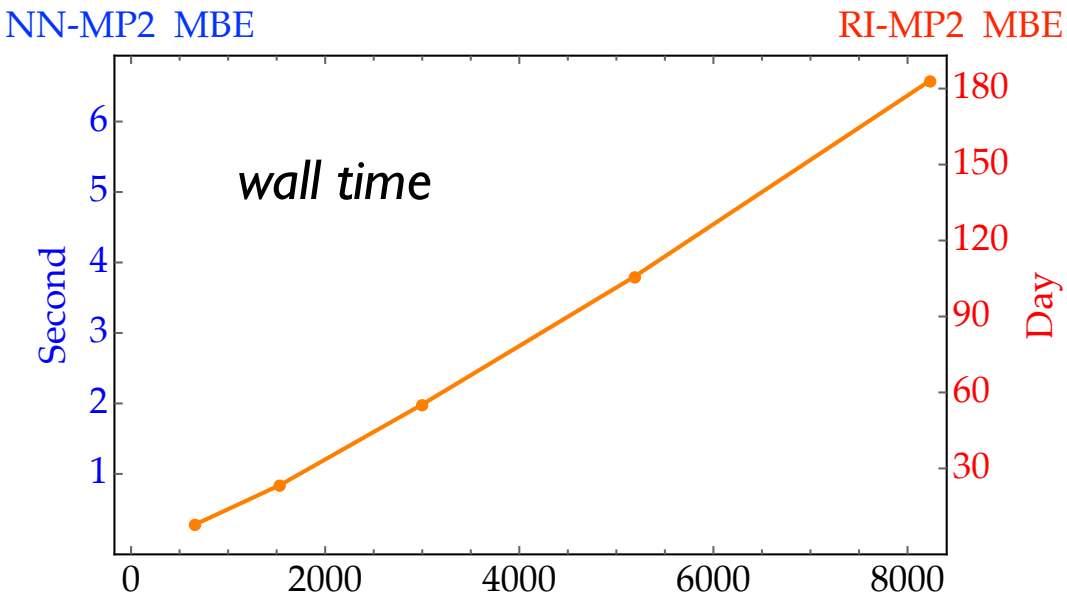
Uses separate monomer dimer etc. training data

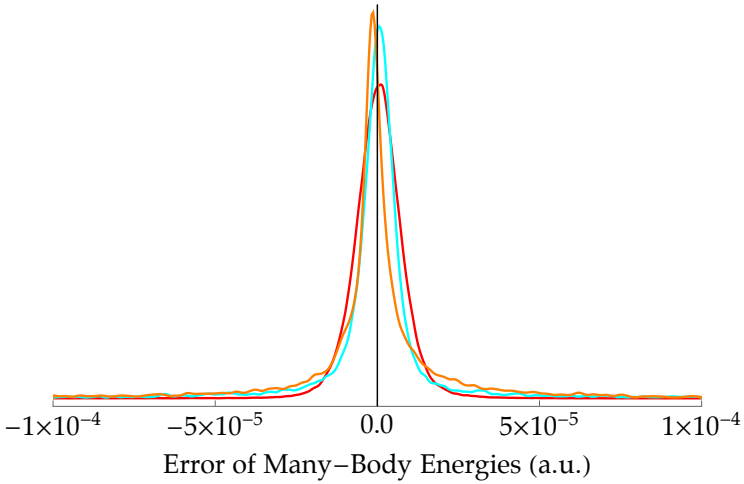
Diatomics-in-molecules NN

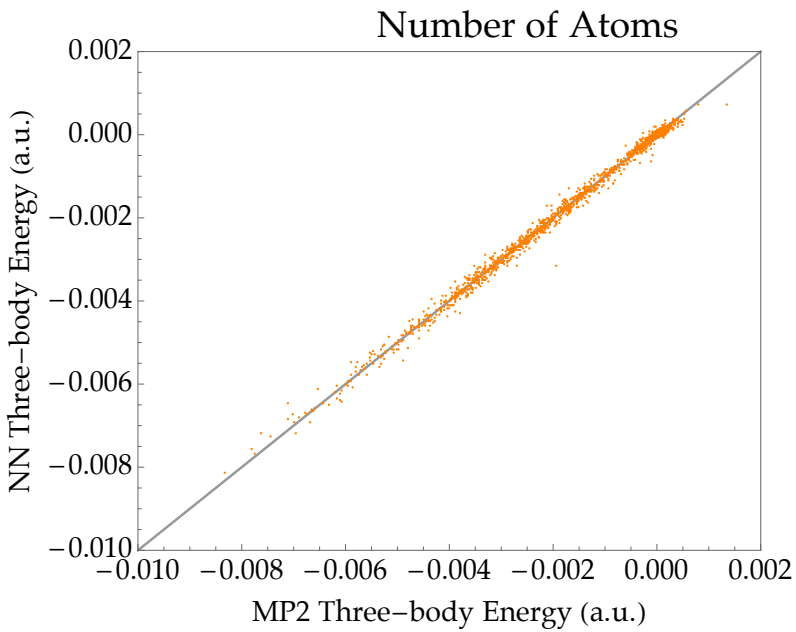
$$E = \sum_{\text{Bonds}} E_{\text{bond}}$$

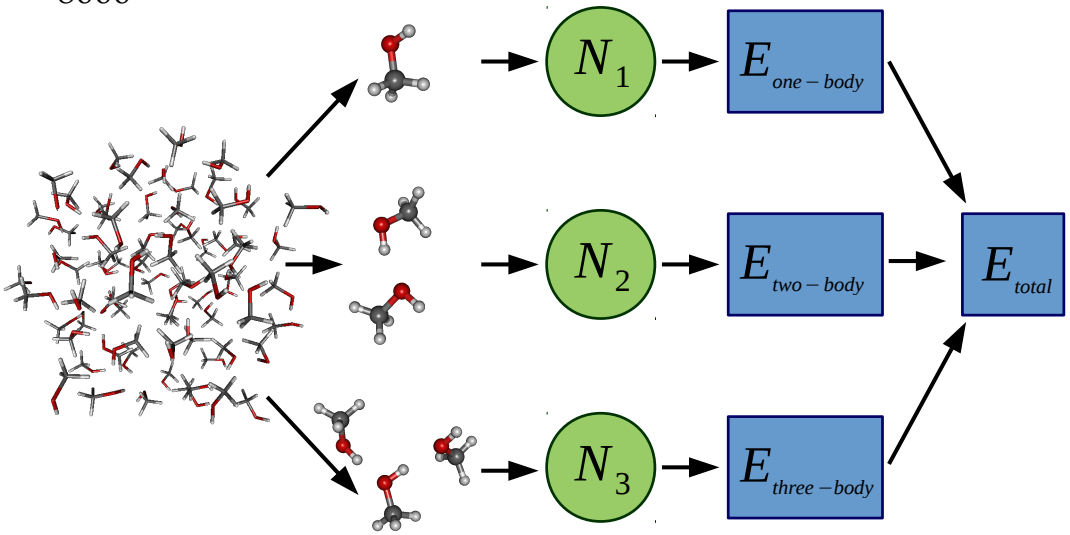
Like Behler-Parinello but bond energies vary less

Neural Network PESs

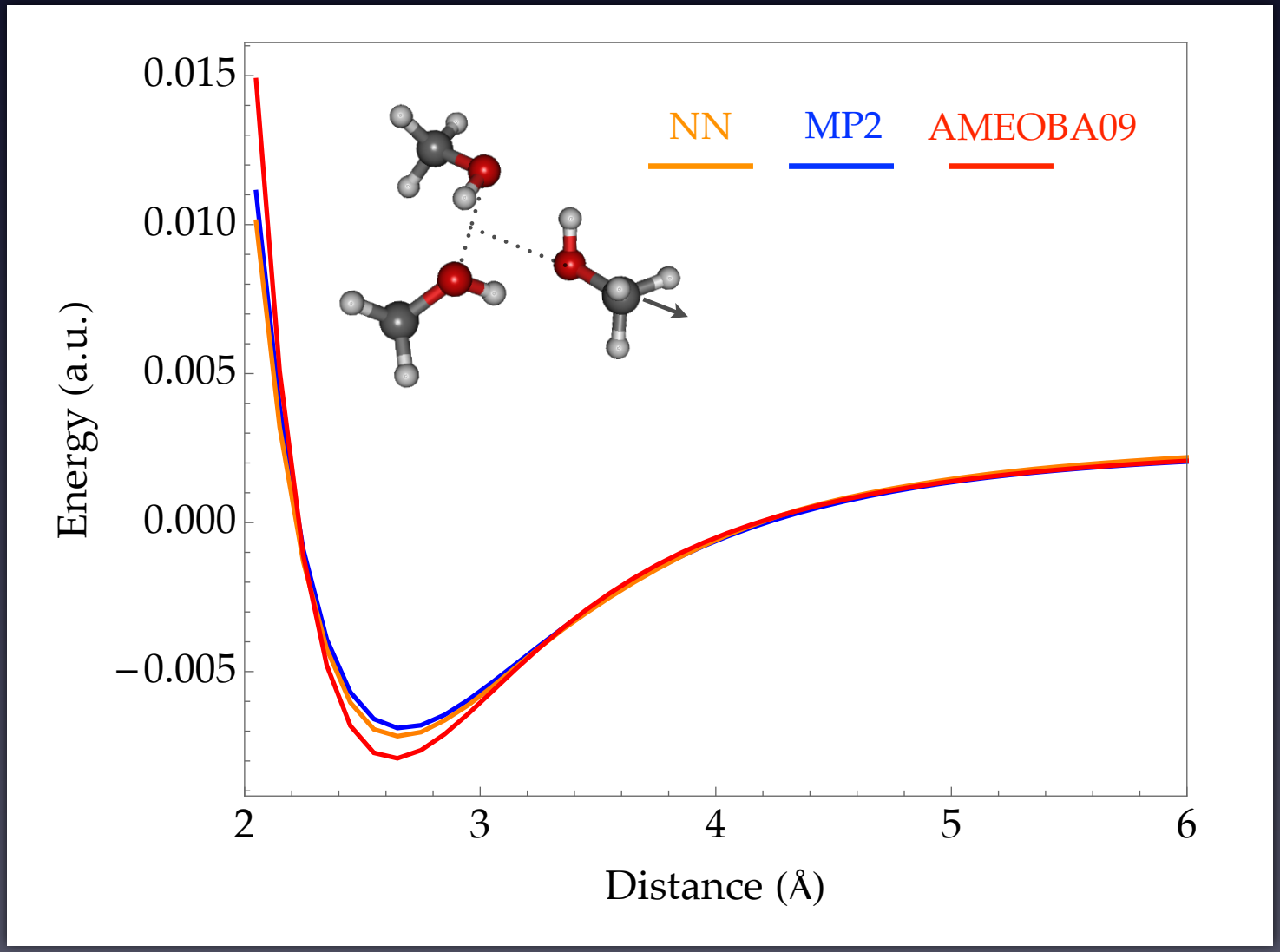




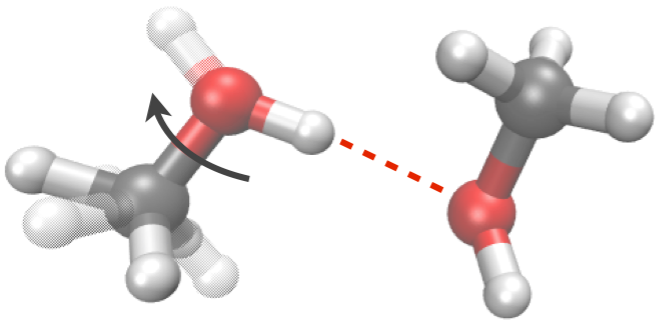




Cluster accuracy

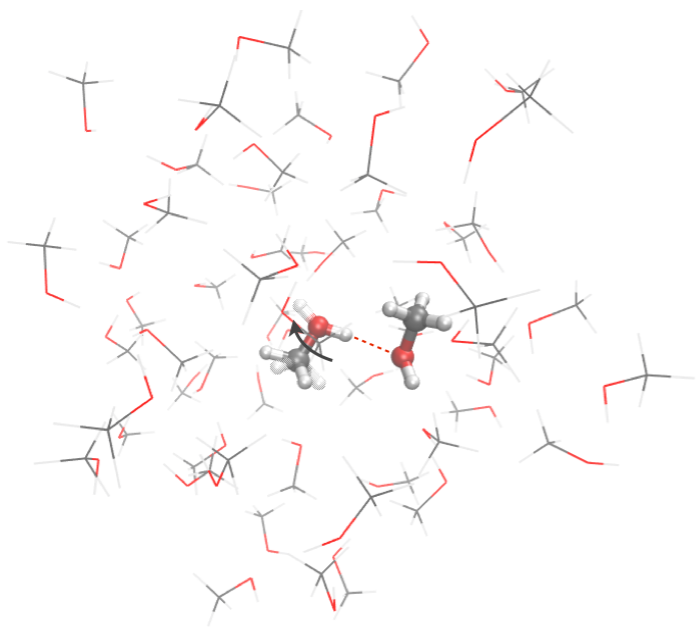


Cluster accuracy



$\Delta E_{MP2\text{-}MBE} = 5.6$

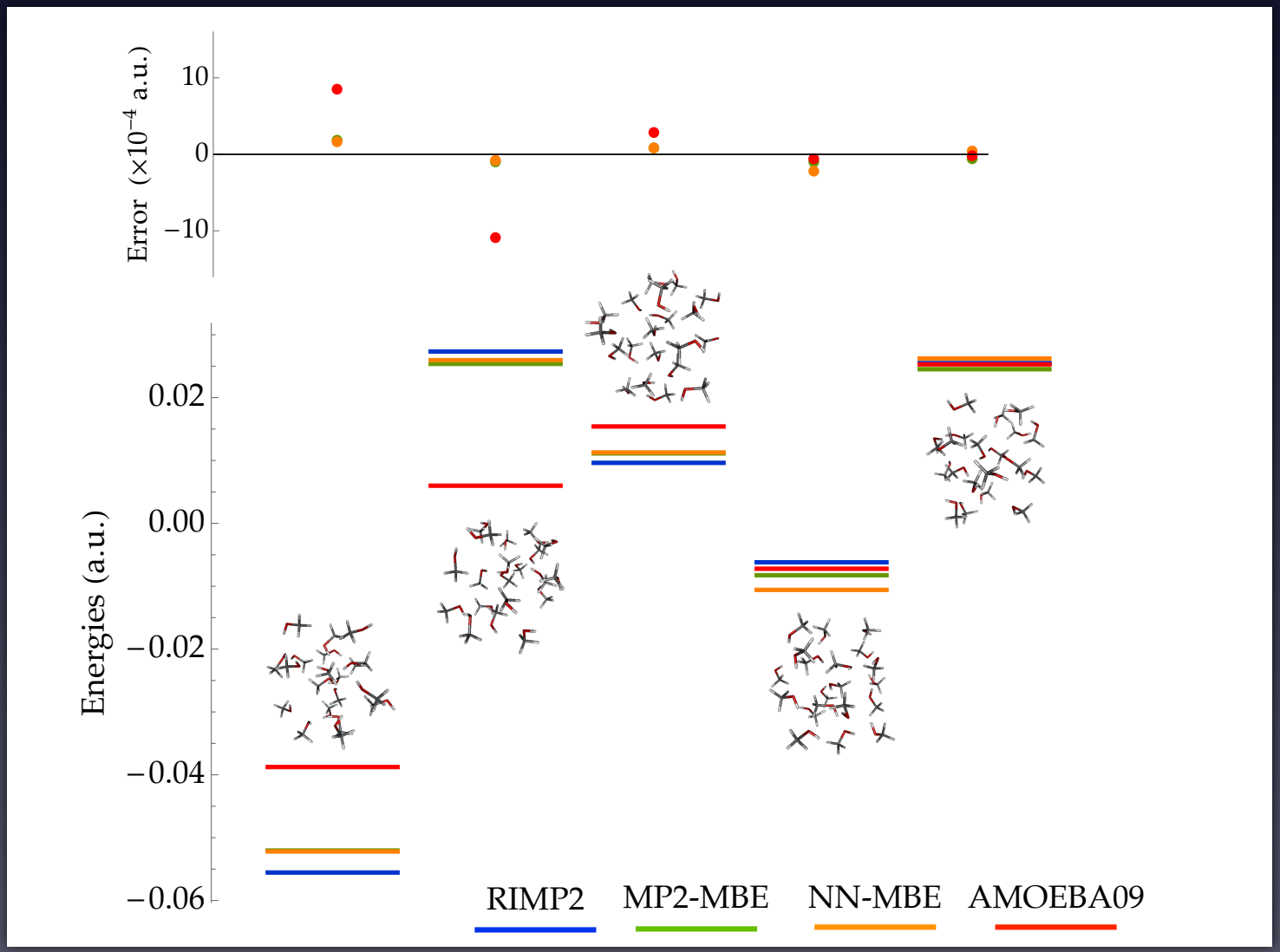
$\Delta E_{NN\text{-}MBE} = 5.8$



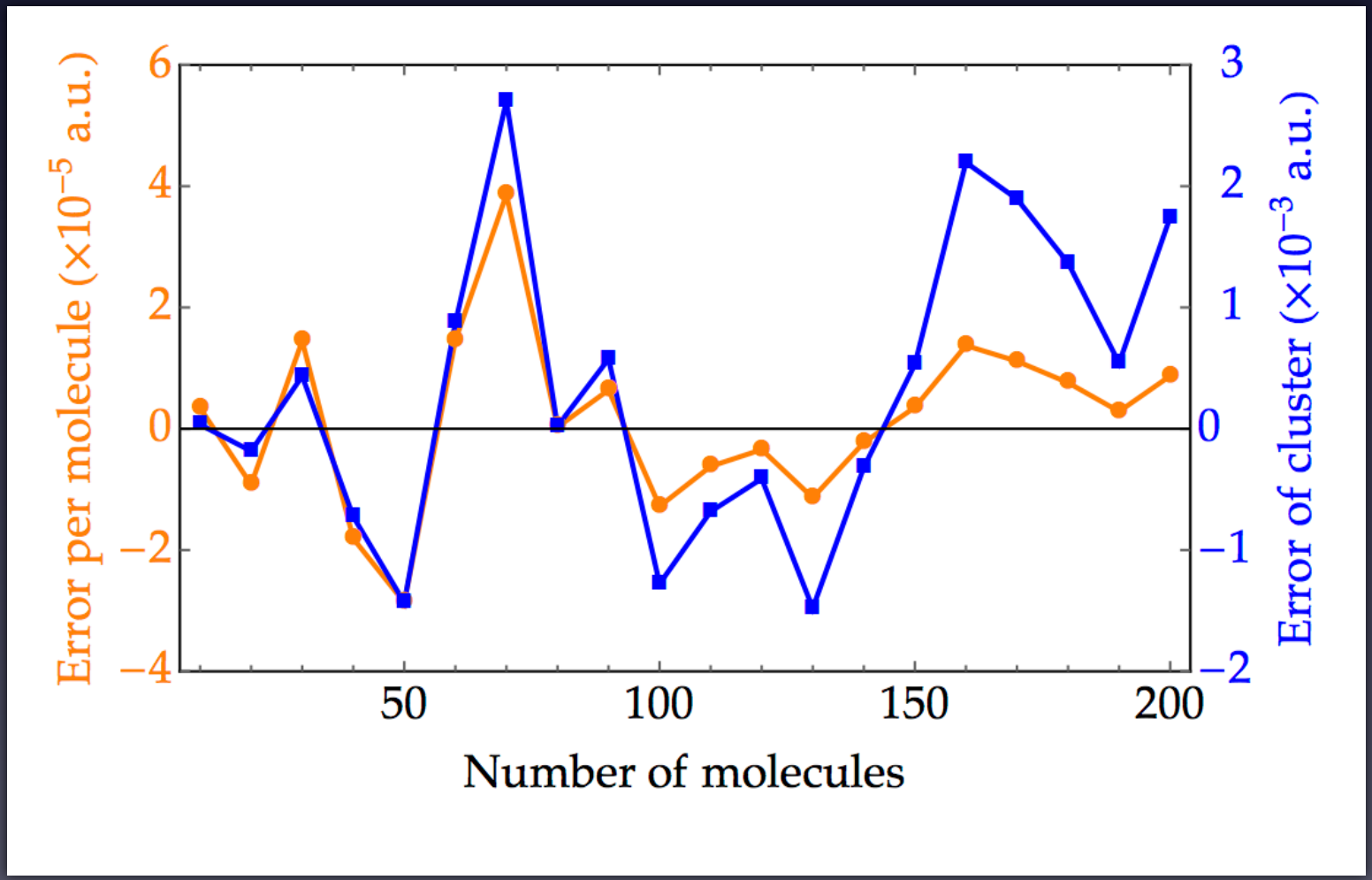
$\Delta E_{MP2\text{-}MBE} = 13.3$

$\Delta E_{NN\text{-}MBE} = 14.6$

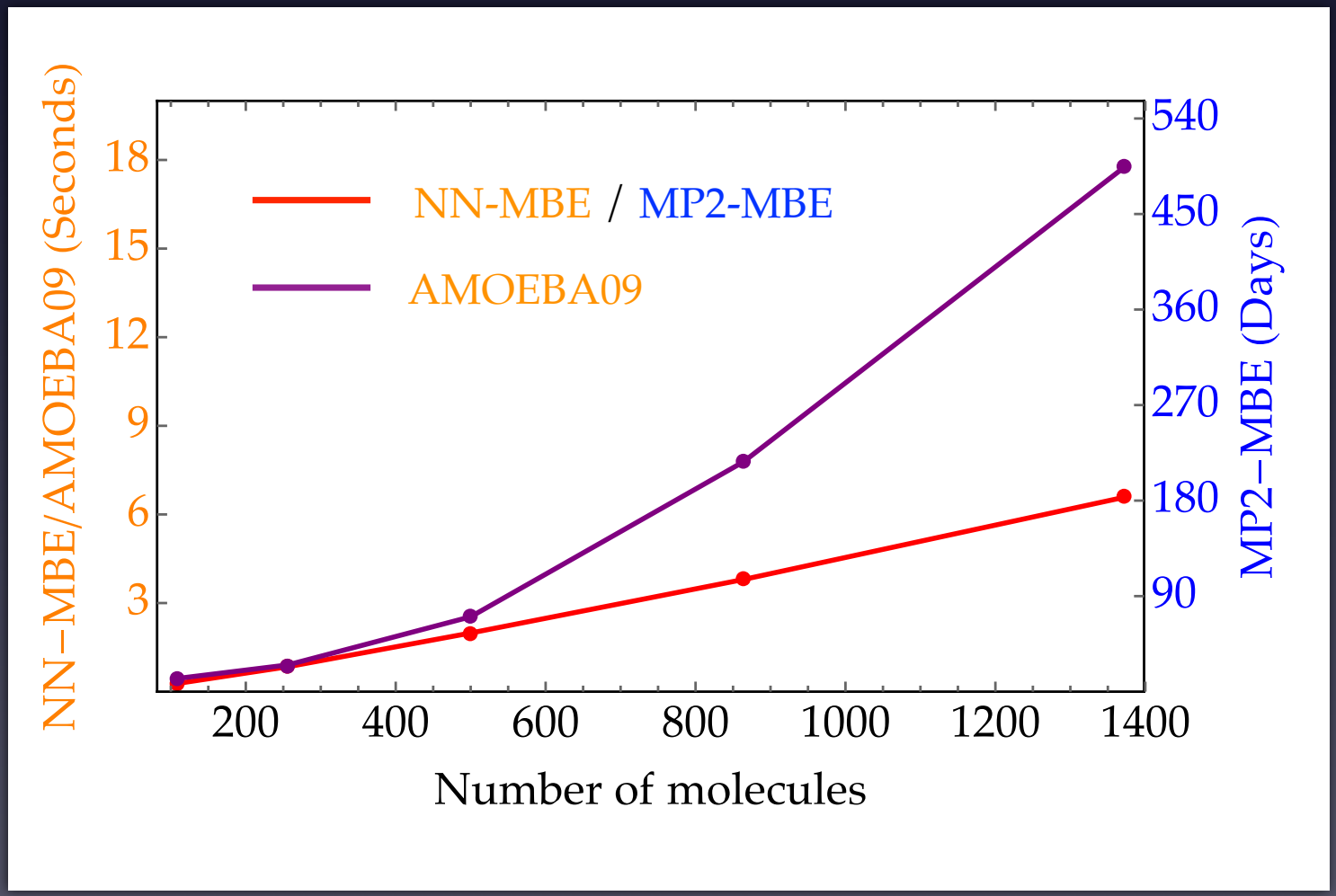
Cluster accuracy



Cancellation of errors in large clusters



Polarizable FF's on notice.

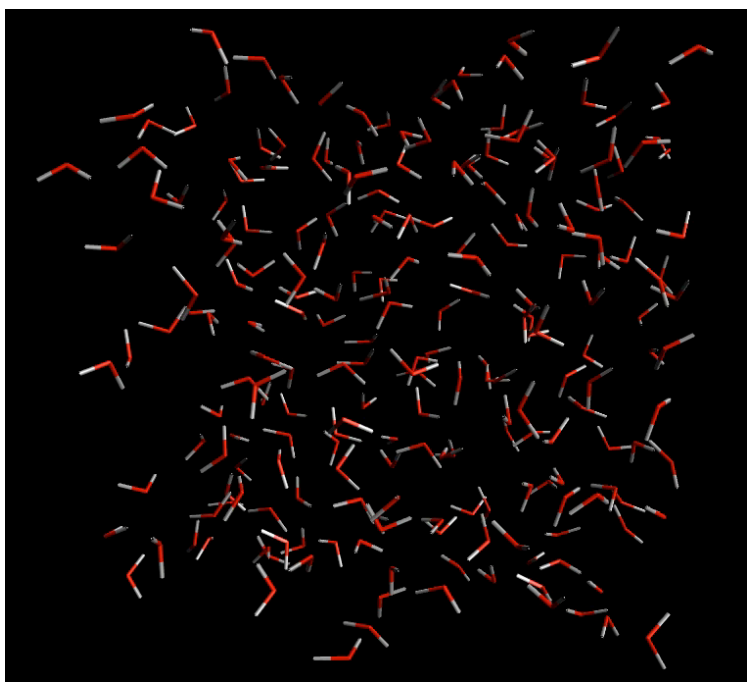


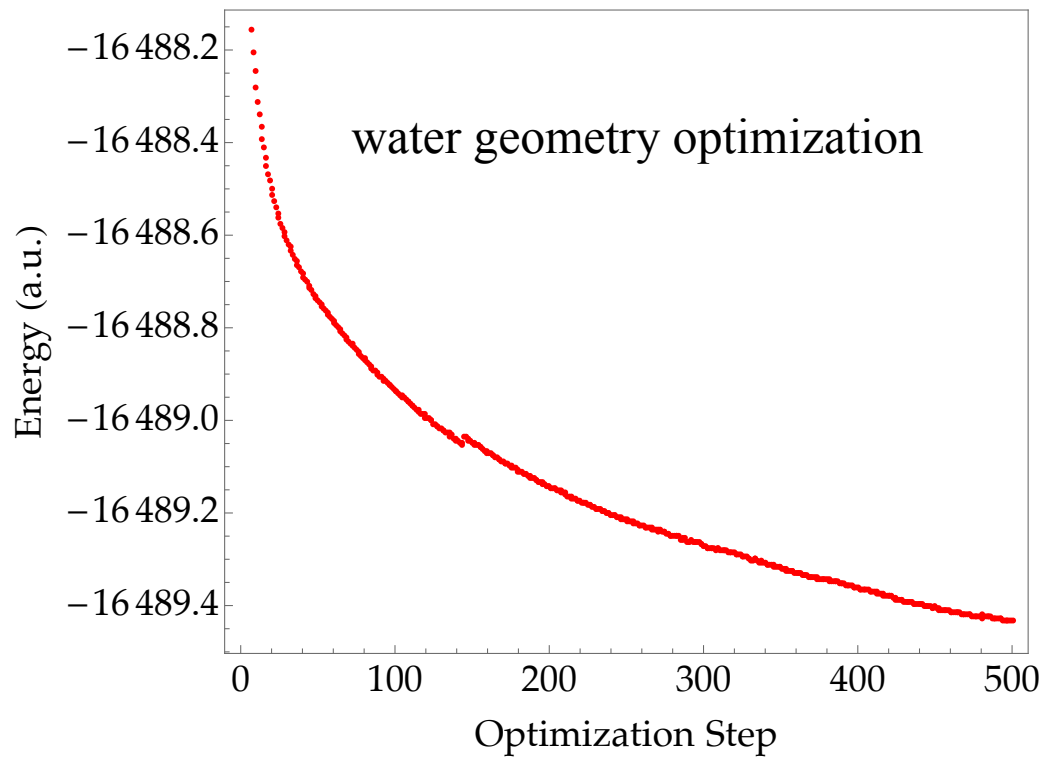
Forces

$$F(R_n) = \frac{\partial E_{total}}{\partial R_n} = \sum_{i=1}^{i_{max}} \frac{\partial E_i}{\partial R_n} = \sum_{i=1}^{i_{max}} \sum_j \frac{\partial E_i}{\partial D_j} \frac{\partial D_j}{\partial R_n}$$

Provided by TensorFlow

We Code

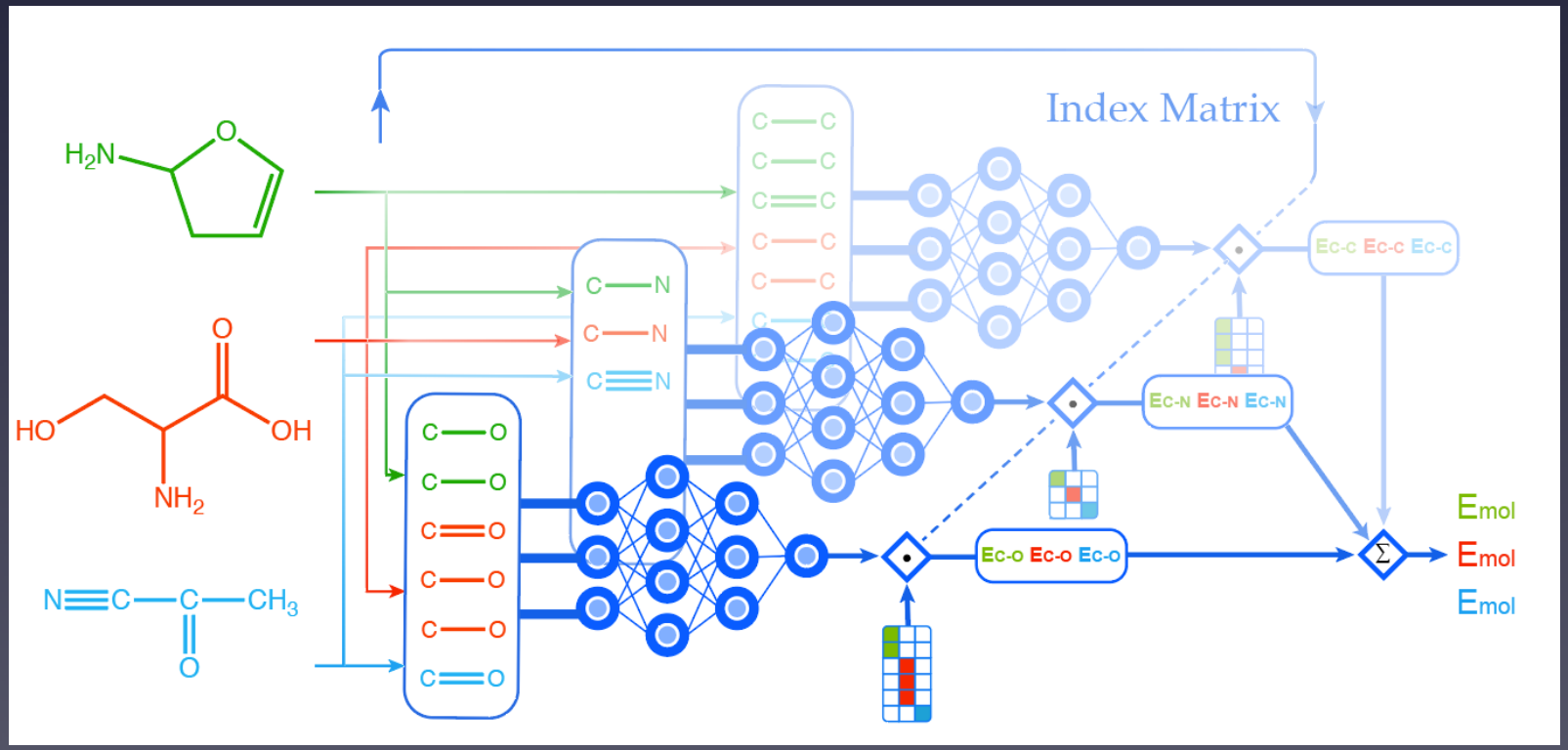




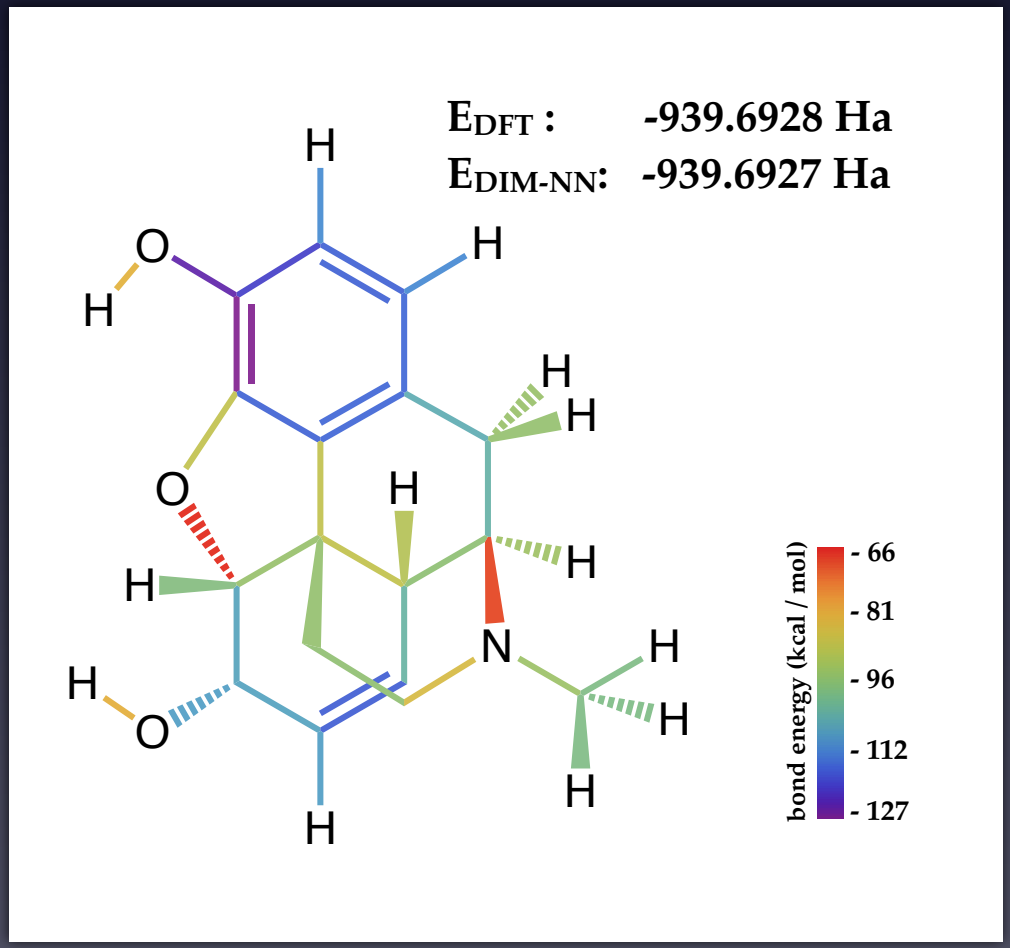
DIM-NN

Expresses the total molecular energy as a sum of bonds

-Only requires total energy training data
-Networks for each bond type

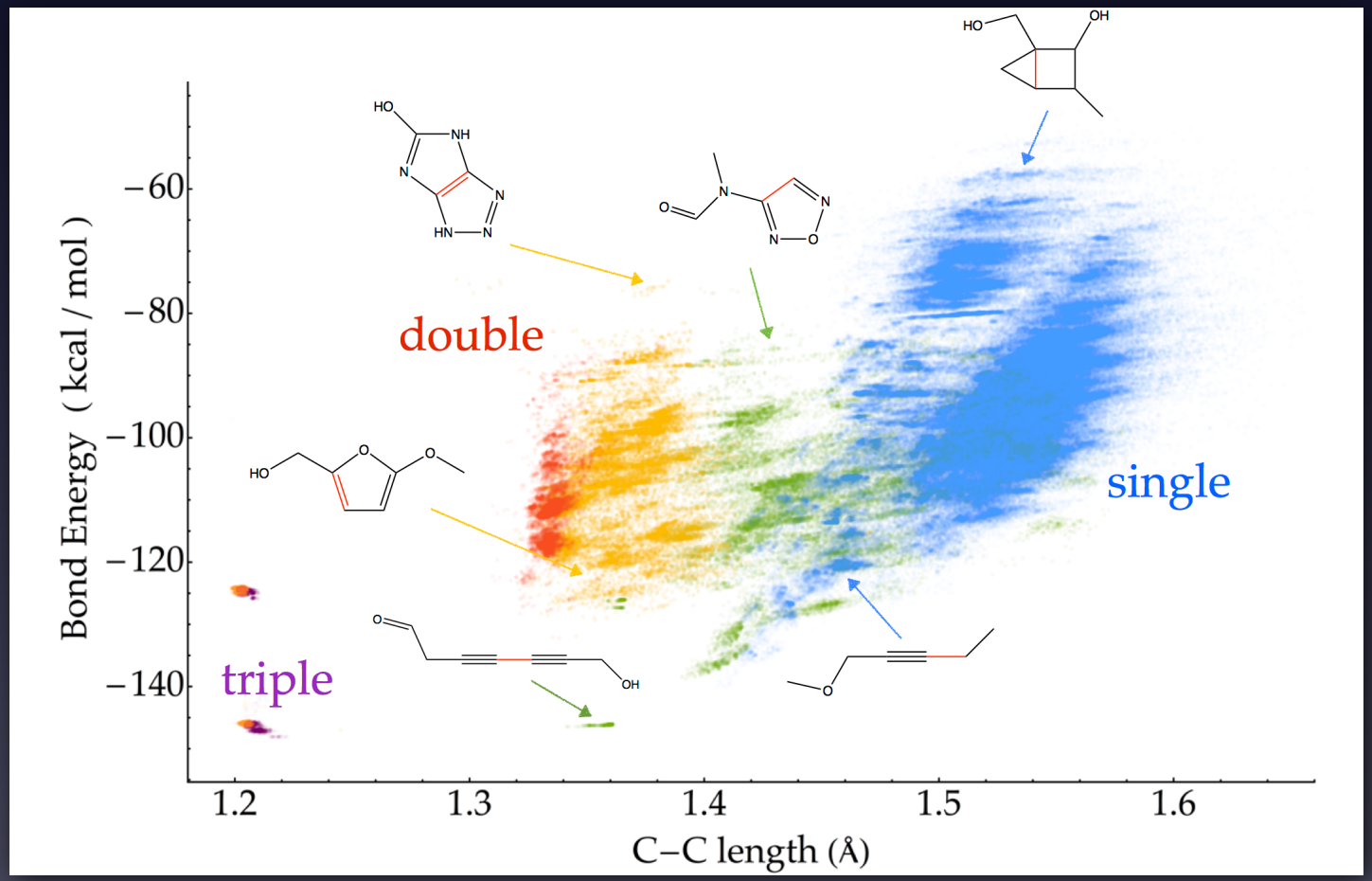


Accurate total energies.



Similar errors for vitamin B12, D3 etc…

The Space of Carbon Carbon Bonds



700,000 Carbon Carbon bond energies.

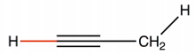
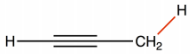
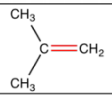
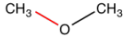
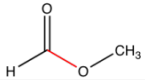
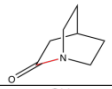
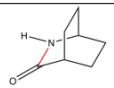
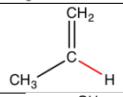
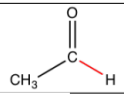
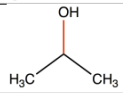
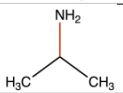
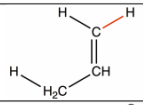
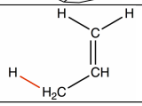
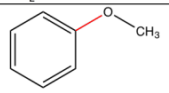
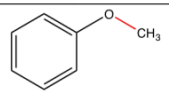
A Synthetic Chemist

Seth Brown University of Notre Dame , South Bend

Chemical Thermodynamics, Chemical Kinetics, Catalysis

35.13

TensorMOL

Case #			Chemist	Neural Network	NBO
1			>>	44.73	0.0123
2			<<	−40.50	−0.0294
3			<	−24.57	−0.0010
4			>>	23.87	0.0097
5			<<	−17.19	−0.0096
6			>>	−16.49	−0.0103
7			>	14.23	0.0020
8			>	11.88	1.6500
9			>>	11.00	−0.0044
10			>	10.68	−0.0016

Conclusions

Because of GPU dependencies and large datasets required, the most powerful ML PES methods are not in common use in chemistry
They will be soon.

Over the next year TensorMol and several other packages will appear where users can "Roll their own" ML-PES's with minimal effort.
These will compete heavily with DFT

The domain of chemical space which can be explored in a weekend is about to exponentially increase.

THANKS!