Stony Brook University

Combining DFT and Machine Learning

Towards faster and more accurate ab-initio calculations

Sebastian Dick, Department of Physics and Astronomy, Stony Brook University
Fernandez-Serra Group
Jr. Researcher Award, 08/16/2018

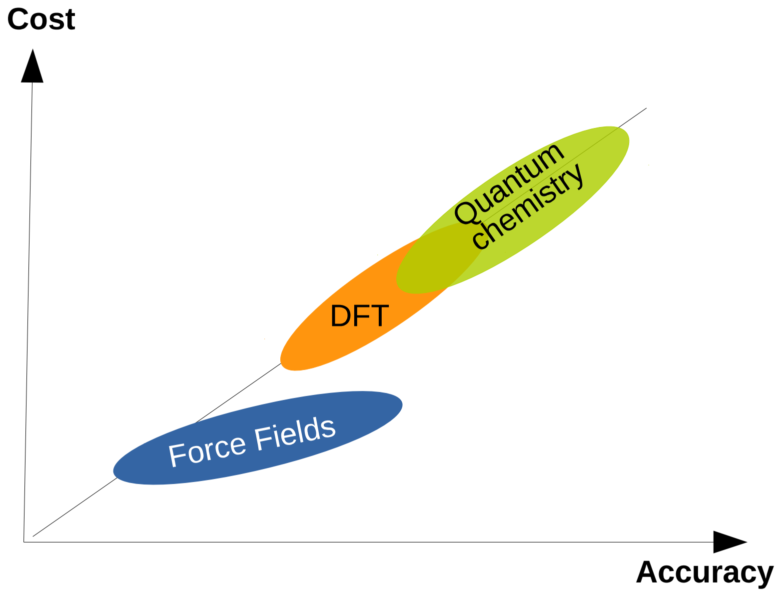
Introduction

Simulations in Molecular Sciences

Atomic coordinates

- **Force Fields**
- **Density Functional Theory (DFT)**
- **Quantum Chemistry**

Energies, **Forces**, Stress, Electron density, Spectra, ...

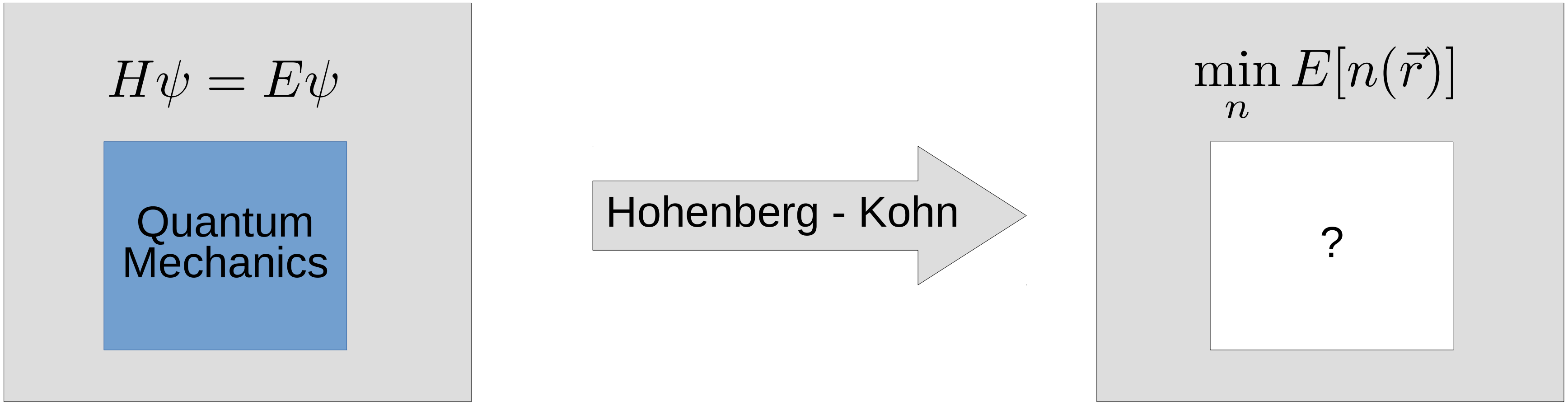


We use DFT because:

- Can scale to large systems sizes (100s to 1000s of atoms) + Periodic boundary conditions

  → Condensed systems
- Non-empirical, hence unbiased
- Fully reactive

How does DFT work ?



$$H\psi = E\psi$$

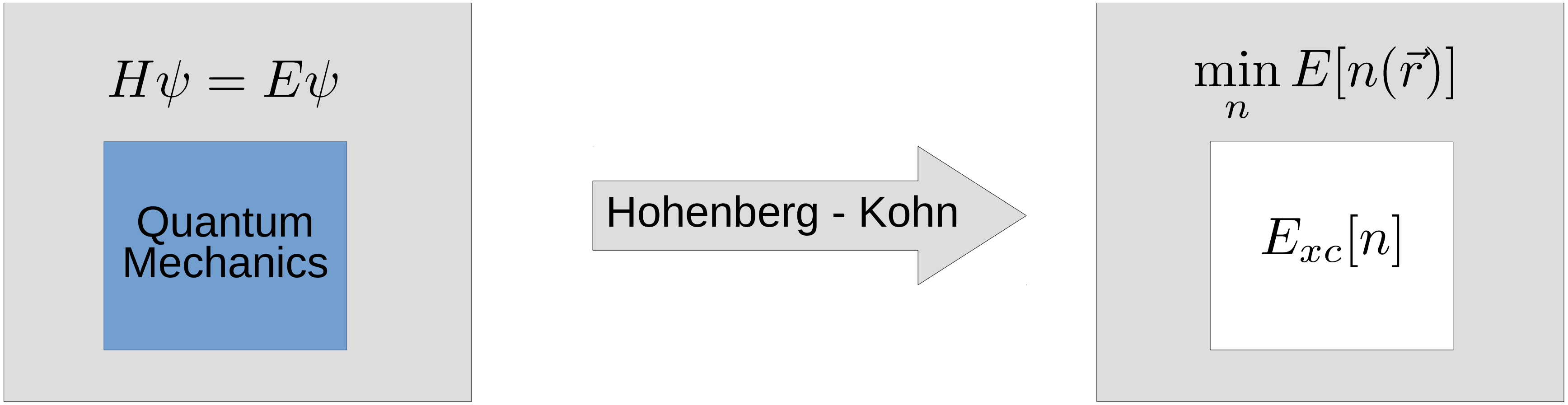
Quantum Mechanics

Hohenberg - Kohn

$$\min_{n} E[n(\vec{r})]$$

?

How does DFT work ?



$H\psi = E\psi$

Quantum Mechanics

Hohenberg - Kohn

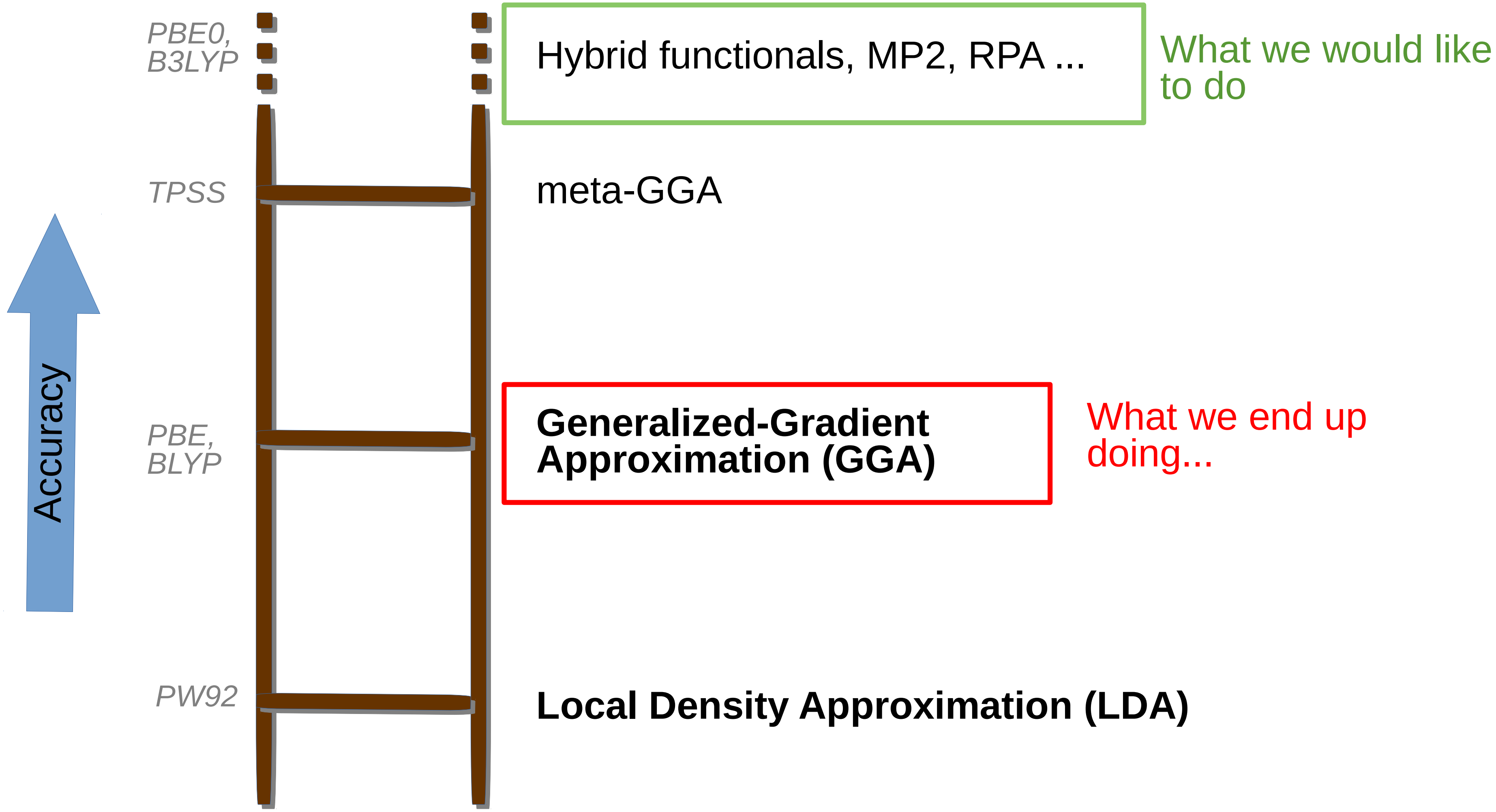
$\min_{n} E[n(\vec{r})]$

$E_{xc}[n]$

$$E[n] = T[n] + E_{xc}[n] + \int d\vec{r}\, V_{ext}(\vec{r}) n(\vec{r}) + E_{Hartree}[n] + E_{II}$$

Jacob's ladder

- A density functional approximation is uniquely defined by choosing $E_{xc}[n]$
- Accuracy, Cost ↔ Locality



PBE0,
B3LYP

Hybrid functionals, MP2, RPA ...

What we would like to do

TPSS

meta-GGA

PBE,
BLYP

Generalized-Gradient Approximation (GGA)

What we end up doing...

PW92

Local Density Approximation (LDA)

Accuracy

Machine learning in Molecular Sciences

Force Fields

Towards Exact Molecular Dynamics Simulations
with Machine-Learned Force Fields
Chmiela et al, arXiv:1802.09238 (2018)

SchNet – A deep learning architecture
for molecules and materials
JCP 148 (2018), Schutt et al

Generalized Neural-Network Representation of
High-Dimensional Potential-Energy Surfaces
PRL 98 (2007), Behler, Parrinello

Electronic Structure

By-passing the Kohn-Sham equations
with machine learning
Brockerde et al., Nature Comm. 8 (2017)

Finding density functionals with machine learning
Snyder et al, Phys. Rev. Lett. 108 (2012)

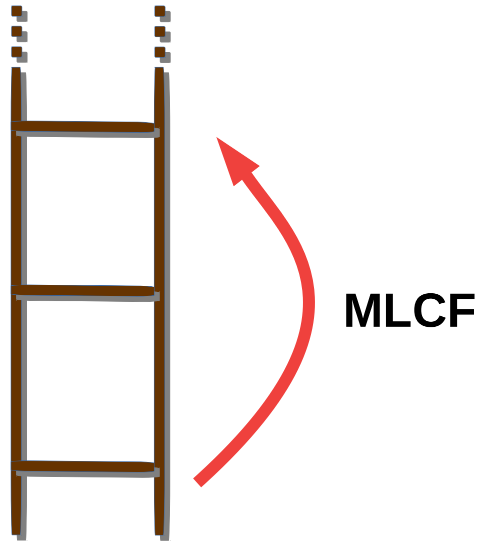
Semi-local machine-learned kinetic
energy density functional
with third-order gradients of electron density
Seino et al, JCP 148 (2018)

Machine learning in Molecular Sciences

Our idea: **Machine Learned Correcting Functionals (MLCFs)**

Train a neural network on the difference in predictions of physical observables (E, F, ...) of a lower accuracy **baseline** method (GGA) and a higher level **reference** method (Hybrid DFT, Coupled Cluster, …)

→ get a higher accuracy at the cost of the baseline method



Force Fields

Towards Exact Molecular Dynamics Simulations
with Machine-Learned Force Fields
Chmiela et al, arXiv:1802.09238 (2018)

SchNet – A deep learning architecture
for molecules and materials
JCP 148 (2018), Schutt et al

Generalized Neural-Network Representation of
High-Dimensional Potential-Energy Surfaces
PRL 98 (2007), Behler, Parrinello

Electronic Structure

By-passing the Kohn-Sham equations
with machine learning
Brockerde et al., Nature Comm. 8 (2017)

Finding density functionals with machine learning
Snyder et al, Phys. Rev. Lett. 108 (2012)

Semi-local machine-learned kinetic
energy density functional
with third-order gradients of electron density
Seino et al, JCP 148 (2018)

Stony Brook University

Machine learned correcting functionals (MLCFs)

Informed Machine Learning for Maximal Extrapolation

Rather than provide all available (raw) data in an unbiased way, **knowledge about the physical mechanisms** involved is used to pre-process and select relevant data.

Trained on a small representative dataset the model should **generalize to unseen data**. In particular, the model has to be valid for **arbitrary system sizes**.

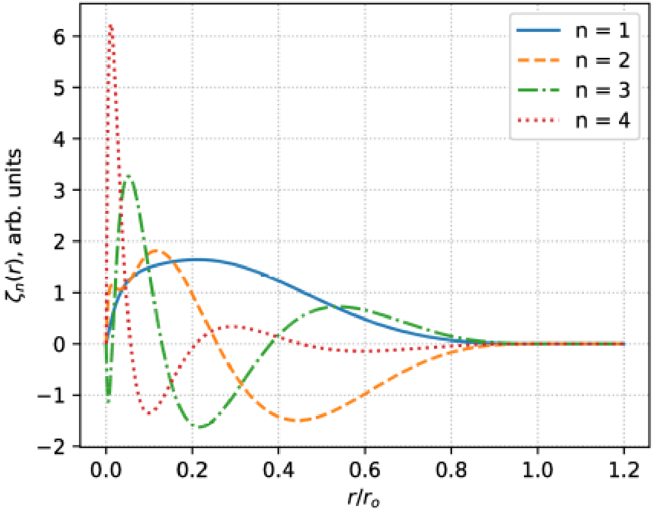
Data

- **Dataset: Water**
  - *Training:* 640 Monomers, 1600 Dimers, 1200 Trimers
  - *Testing:* 160 Monomers, 400 Dimers, 300 Trimers, 50 Tetramers, 50 Pentamers, …

- **Input:** Expansion of electron density around each atom into basis functions:

$$\psi_{nlm}(\vec{r}) = Y_l^m(\theta,\phi)\zeta_n(r)$$

$$\int_{\vec{r}} \psi^*_{nlm}(\vec{r})\psi_{n'l'm'}(\vec{r}) = \delta_{nn'}\delta_{ll'}\delta_{mm'}$$



Electronic descriptors:

$$\boxed{c^{\alpha,I}_{nlm} = \int_{\vec{r}} \rho(\vec{r} - \vec{r}_{\alpha,I})\psi^{*\alpha}_{nlm}(\vec{r})}$$

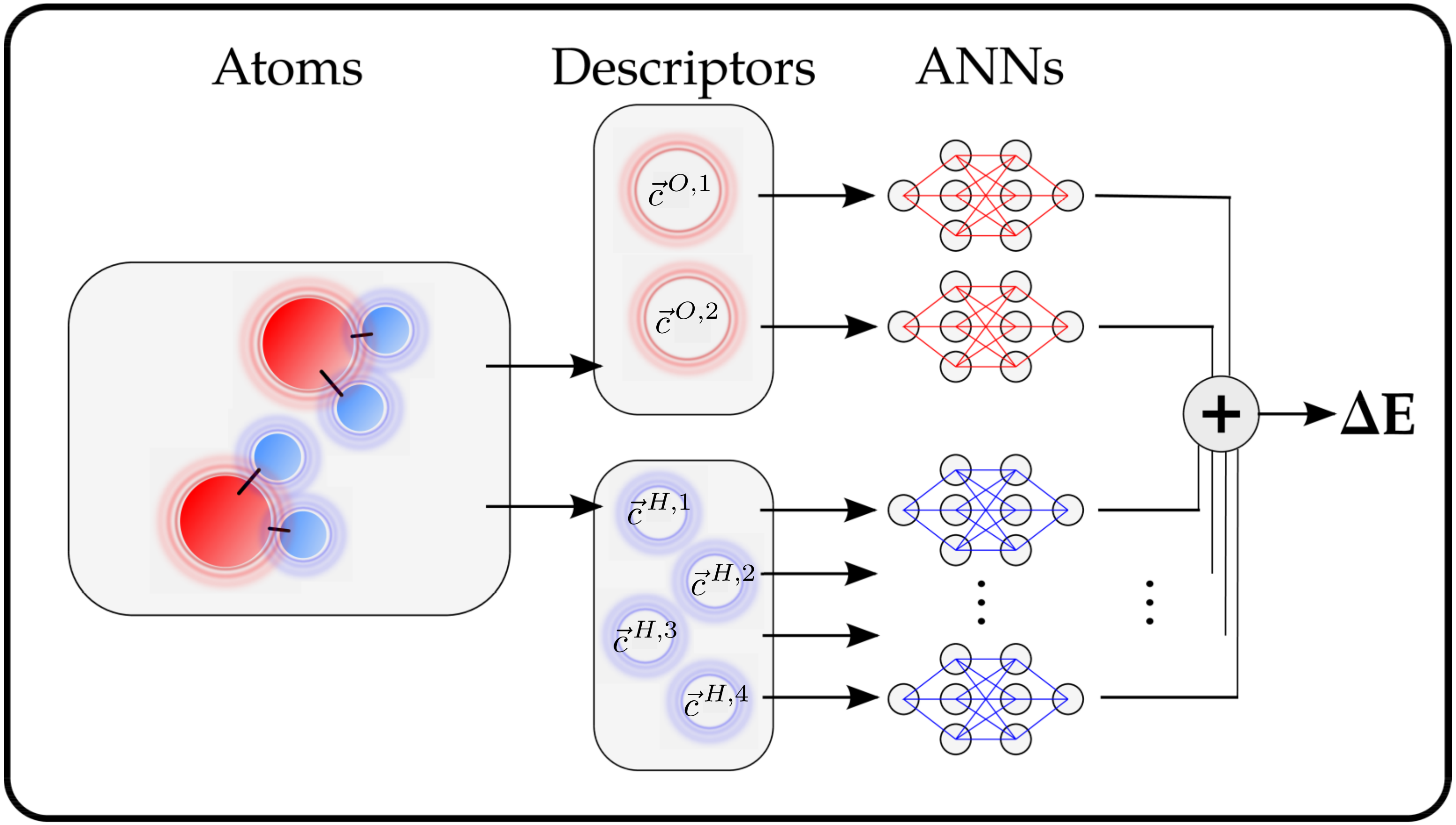
Atomic species Atom index

- **Targets:** Difference between reference (MB-pol) and baseline (GGA + vdW) energies(/forces)

$$\Delta E = E_{reference} - E_{baseline}$$

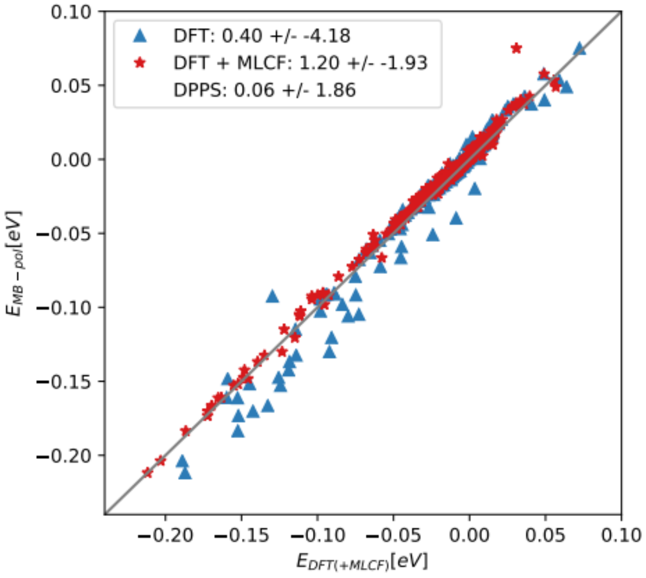
Architecture

$$n \rightarrow \Delta E_{MLCF}[n]$$



Performance on water clusters

2-body energy

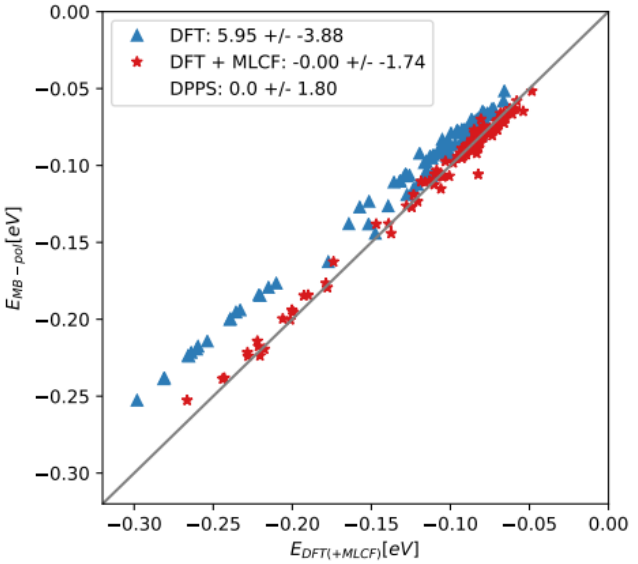


$$\delta E = E_{ref} - E_{pred}$$

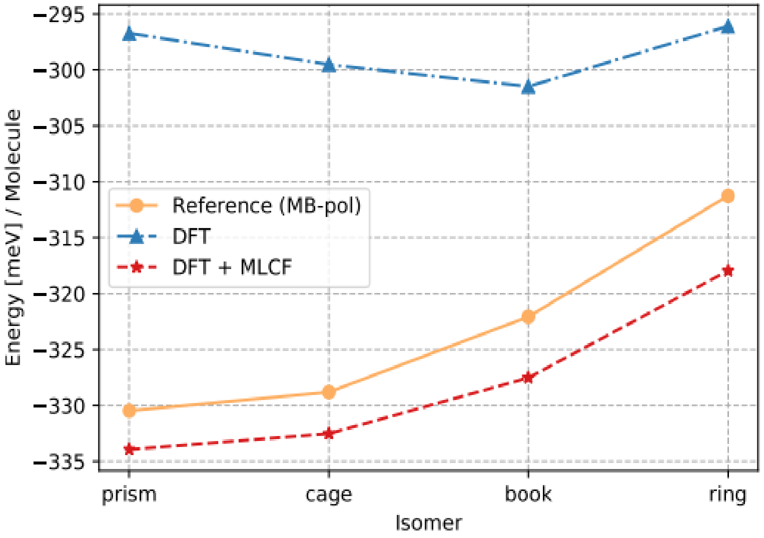
Molecules	$\langle \delta E \rangle$		$\sqrt{\langle (\delta E)^2 \rangle - \langle \delta E \rangle^2}$	
	DFT	**DFT+MLCF**	**DFT**	**DFT+MLCF**
1	-4.2	-1.4	64.3	**2.0**
2	-5.8	-1.3	42.5	**3.4**
3	-14.8	0.6	31.9	**2.3**
4	-31.2	-1.0	9.4	**2.7**
5	-31.9	0.0	12.3	**3.0**
8	-28.9	2.3	9.3	**3.1**
16	-26.1	6.6	6.2	**2.5**

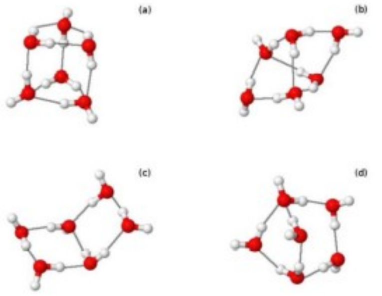
Energies in meV/molecule

3-body energy



Hexamers



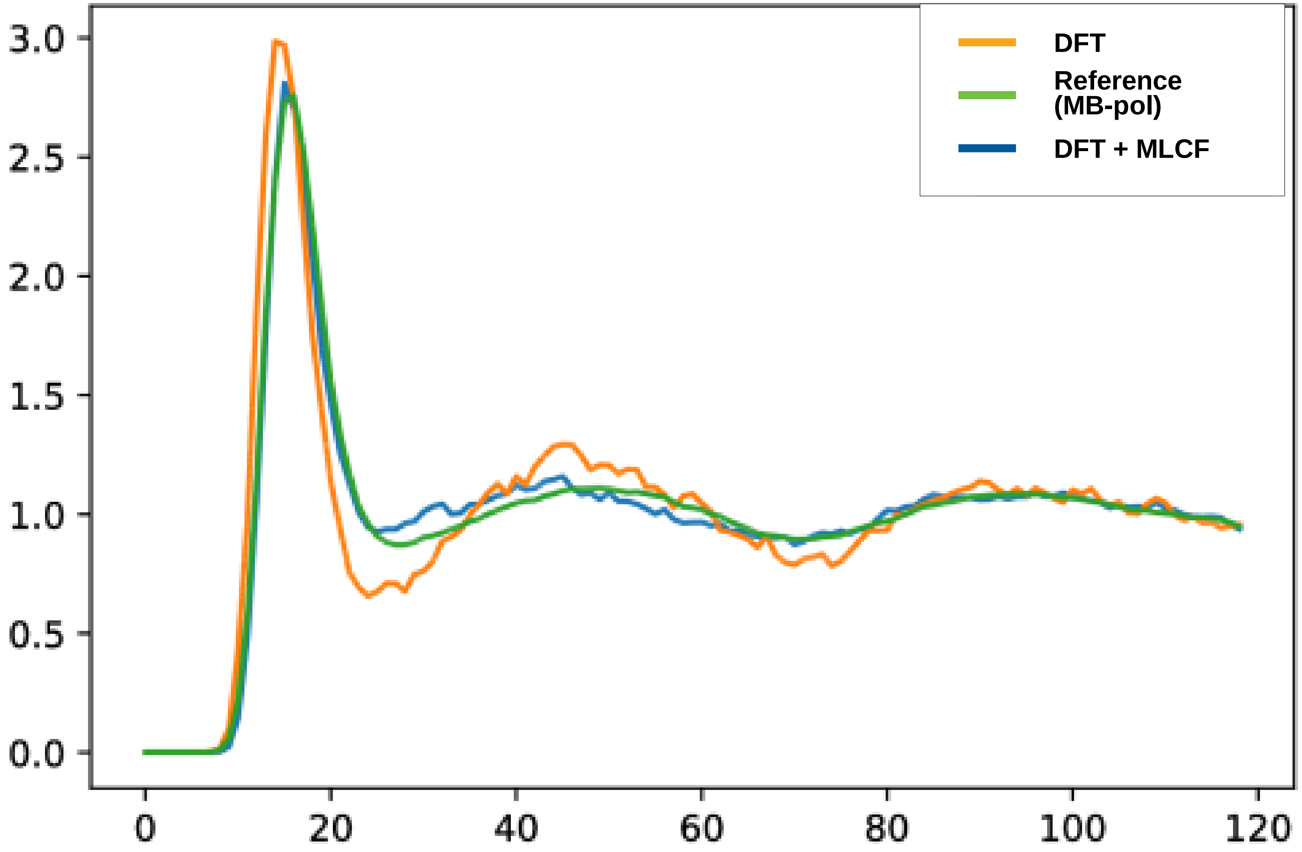


Fritz, Fernandez-Serra, Soler, J. Chem. Phys. 144, 224101 (2016), Supplementary Information

Correcting molecular dynamics simulations

- **Ab initio molecular dynamics:** Integrate the equations of motion with forces obtained from ab-intio calculations.
- GGA (DFT) is known to over-structure liquid water (peaks too high)
- Even though simulations not well converged yet (simulation time too short), MLCFs seem to correct this over-structuring

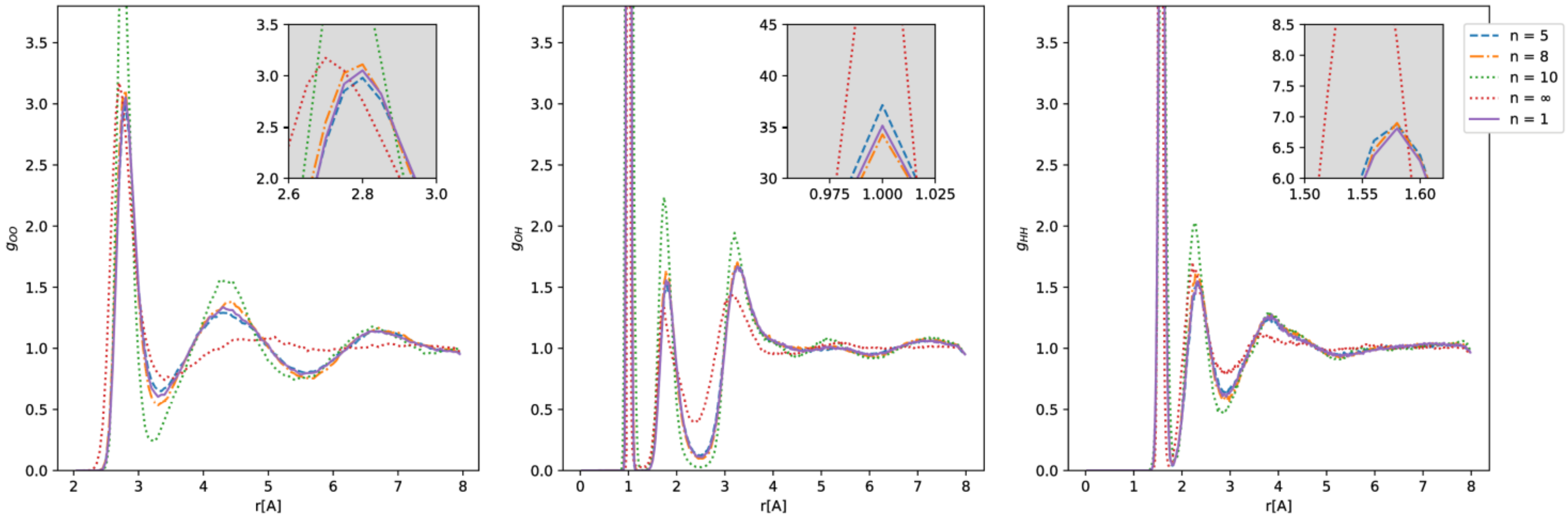
Simulation of a box with periodic boundary conditions containing 128 water molecules, with Nose-Hoover Thermostat at 300 K



Using MLCFs to speed up MD calculations

Stony Brook University

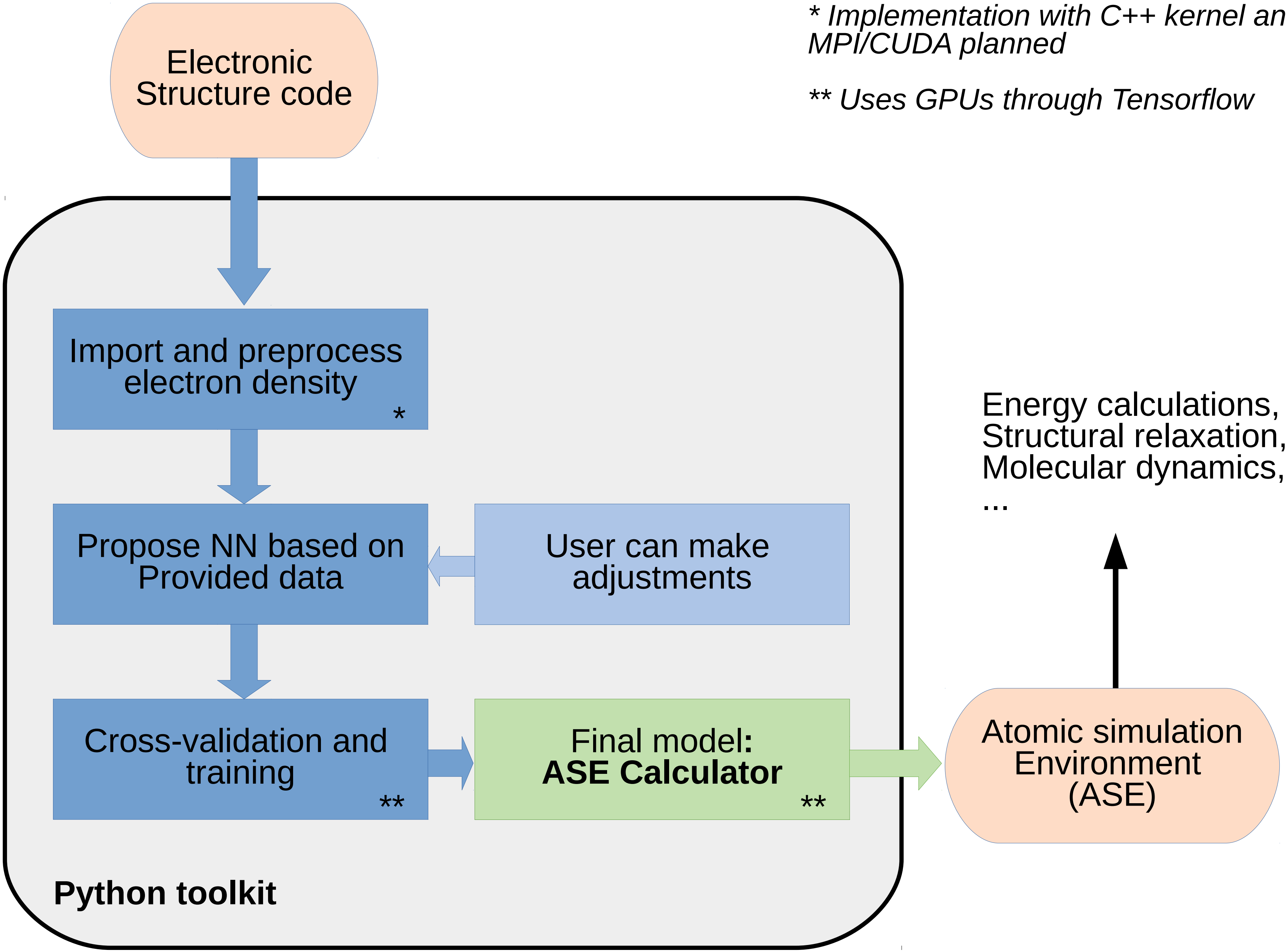
- Start from very fast DFT calculation with very low accuracy (GGA, minimal basis set, coarse grid, relaxed convergence criteria)
- Large difference between baseline and reference → only approximate correction
- Solution: Every n-th MD step use reference method to calculate correction
- **Speed-ups of up to a factor of 8 for water**
- But: possible speed-up system dependent, careful validation necessary



Outlook

Python toolkit

Stony Brook University



** Implementation with C++ kernel and MPI/CUDA planned*

*** Uses GPUs through Tensorflow*

Electronic Structure code

Import and preprocess electron density *

Propose NN based on Provided data

User can make adjustments

Cross-validation and training **

Final model: **ASE Calculator** **

Atomic simulation Environment (ASE)

Energy calculations,
Structural relaxation,
Molecular dynamics,
...

Python toolkit

Timeline

Timeline for 2018/2019:

- Sep – Dec:
  - Implementation of basic Python toolkit, v0.1 on Github
  - First publication on MLCFs
  - Using MLCFs to study the solvation of NaCl in water (together with Alec Wills)
- Jan – Apr:
  - Performance optimization (C++ and MPI/CUDA), v1.0 on Github
  - MLCF accelerated simulations of water-metal interfaces
- May – Aug:
  - MLCF accelerated simulations of water-metal interfaces
  - MLCFs as an alternative to QM/MM? Implementation of QM/QM-MLCF algorithms.

Plans for 2019/2020:

- Can ML be used to correct the self consistent electron density? (Possible collaboration with Alan Aspuru Guzik @ Toronto)
- Machine learned density functional kernels?
- Other semi-empirical methods for faster electronic sturcture calculations (Electron ‘force-field’, Collaboration with Jose Solers group @ Madrid)

Thank you!

Using MLCFs to speed up MD calculations

Stony Brook University

Replace QM/MM with QM/QM-MLCF: