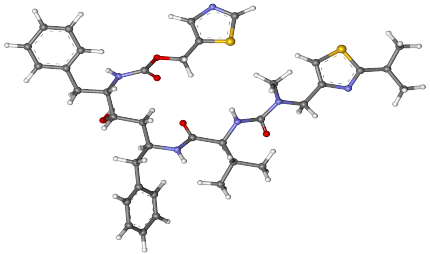
Neural networks & Crystal structure prediction

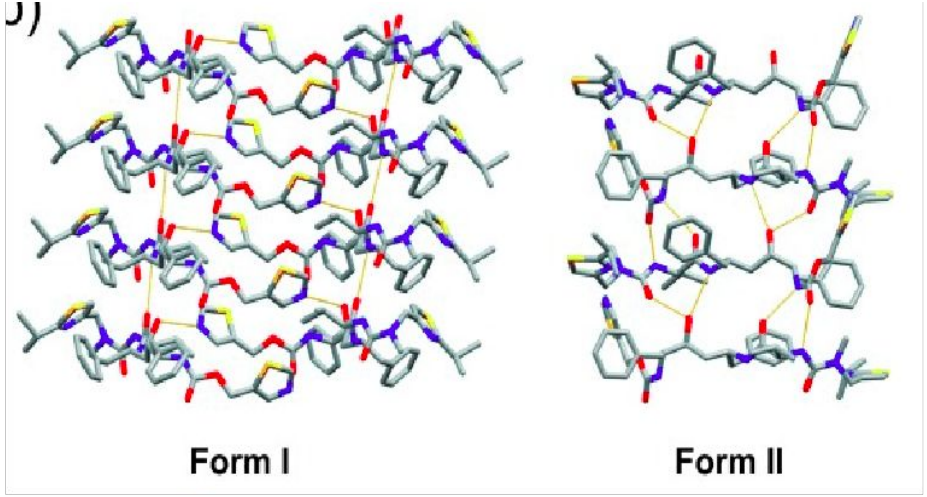
Electronic Structure Theory - SISSA
Emine Kucukbenli

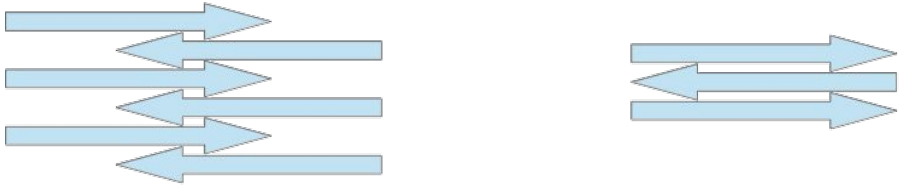
Polymorphism in real life



Famous example:
Ritonavir

- HIV/AIDS & HepC med.
- 1996 - Entered in market in "*Form I*" (non-ref. caps)
- 1998 - Removed from the market in "*Form II*"



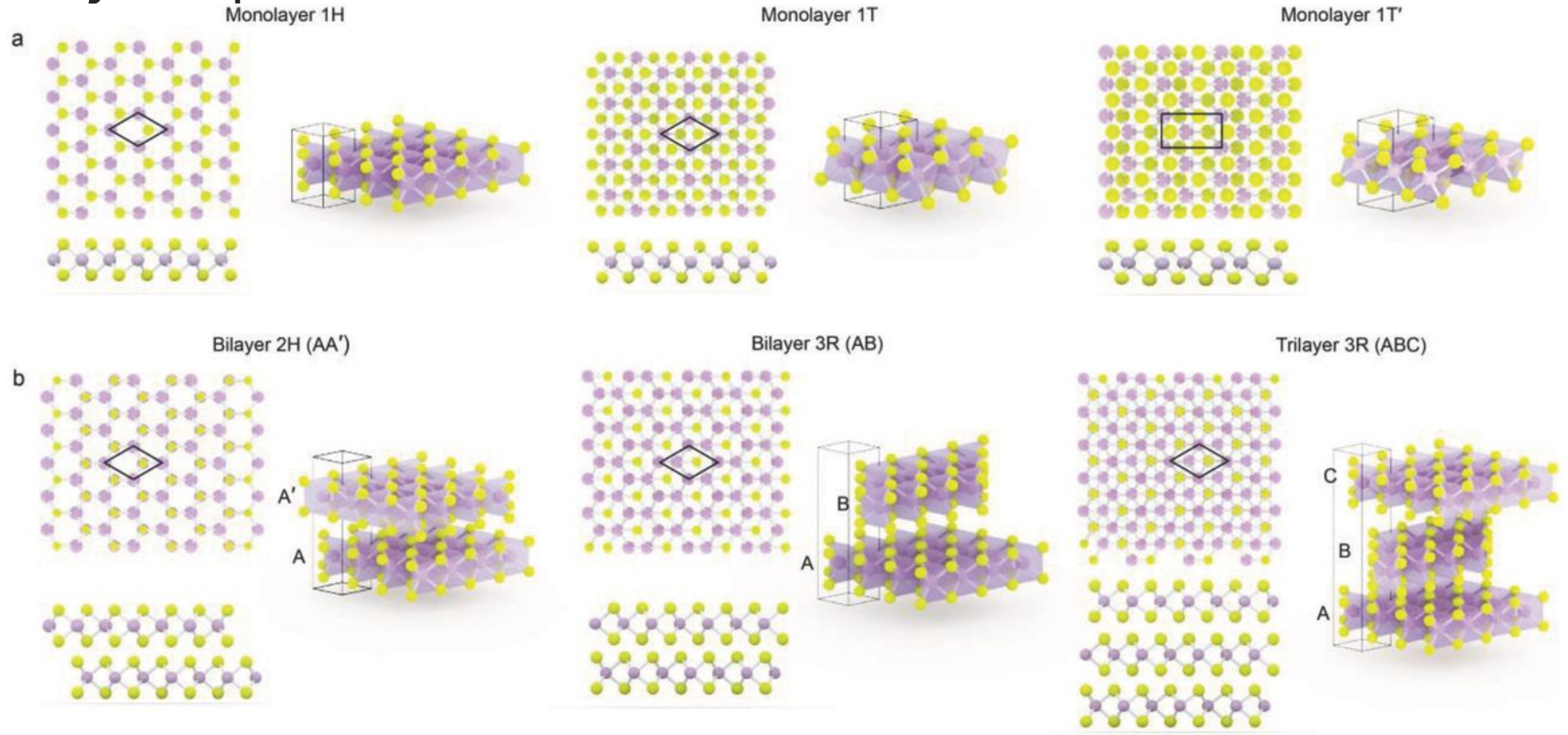


Bauer et al., Pharmaceutical Research 18 (2001)
Datta et al.,Nature Reviews Drug Disc. Rev. 3 (2004)
Bucar et. al.,Angew. Chem. 54 (2015) "disappearing polymoprhs"

Polymorphism

- Aspirin, paracetamol & others
- Al2O3 - catalysis
- Mn-cubane - catalysis
- Coronene & magnetism
- “Jumping crystals”
- 2D-heterostructures layer polymorphism and polytypes, e.g. CO2 reduction race

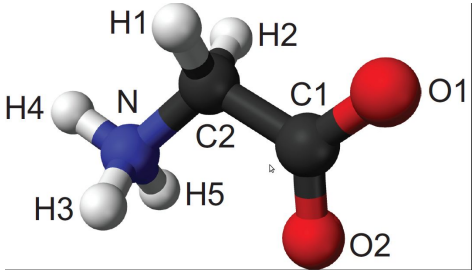
Polymorphism

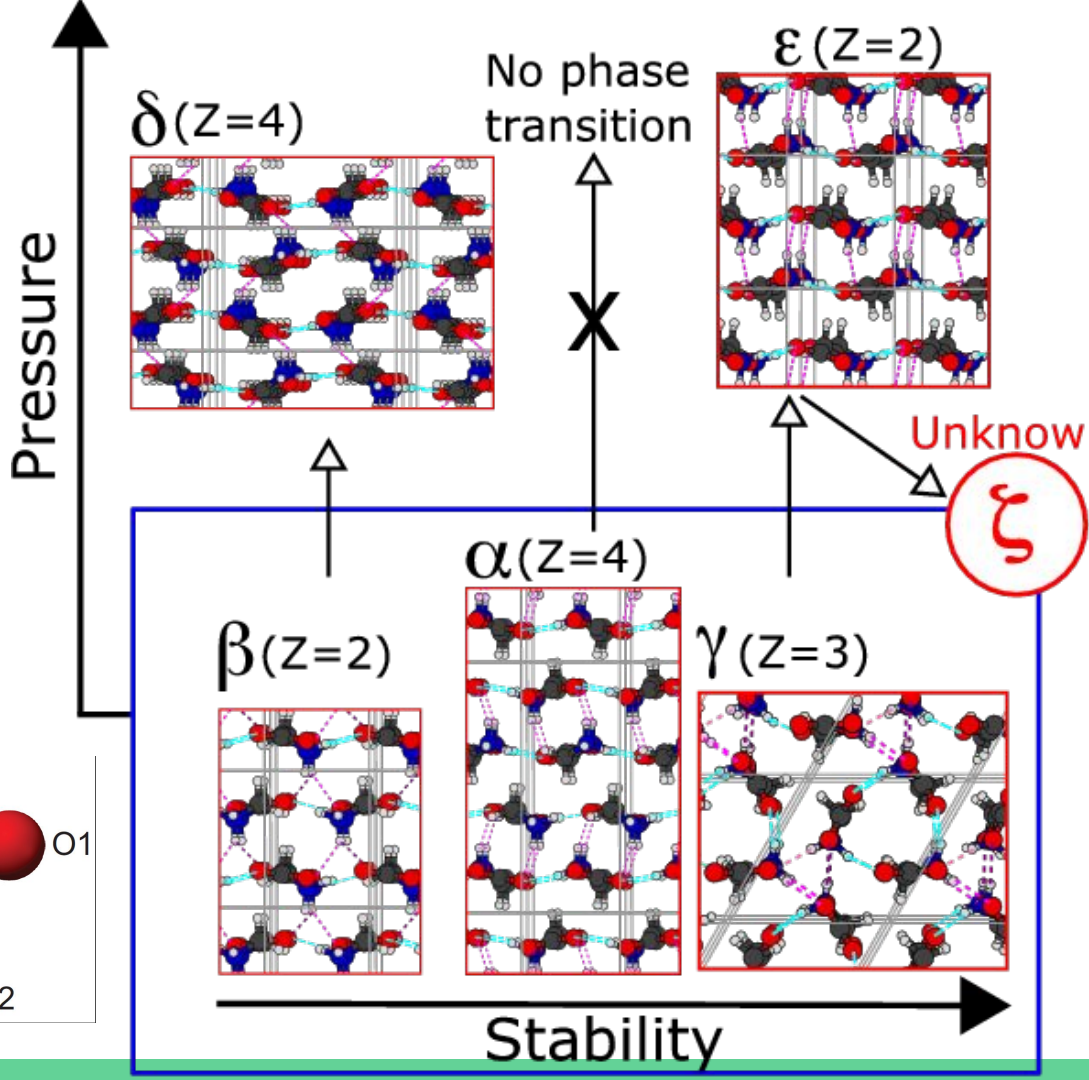


Zhao et al. Advanced Mat. 30, 1802397 (2018)

Polymorphism

- Aspirin, paracetamol & others
- Al2O3 - catalysis
- Mn-cubane - catalysis
- Coronene & magnetism
- “Jumping crystals”
- 2D-heterostructures layer polymorphism and polytypes, e.g. CO2 reduction race





Given a molecule/stoichiometry:

1. CSP questions:
- What is the crystal structure?
- How stable/soluble/active is it?
- Structure with optimum properties?

2. CSP Strategies:
- Explore the phase space
- Characterize what is found
- Build descriptors for desired properties

3. CSP challenges:
- Many possibilities to explore
- Big unit cells (Z>1, incomm. layers)
- Several energy & length scales
- Predictive power
- Structure-property - it's complicated

4.Where ML can help:
- Accelerate exploration
- Accelerate characterization
- Recognize descriptors, propose new structures with desired properties (Reinf. Learning, Gen. Models, Autoencoders etc.)

Exploring polymorphism with Genetic Algorithm

1. A population of random structures
2. Energy ordering - select champions
3. Champions reproduce: Make offsprings
4. Decide which children will survive
5. Repeat 2-4 as long as you can;
6. Repeat 1-5 for different Z

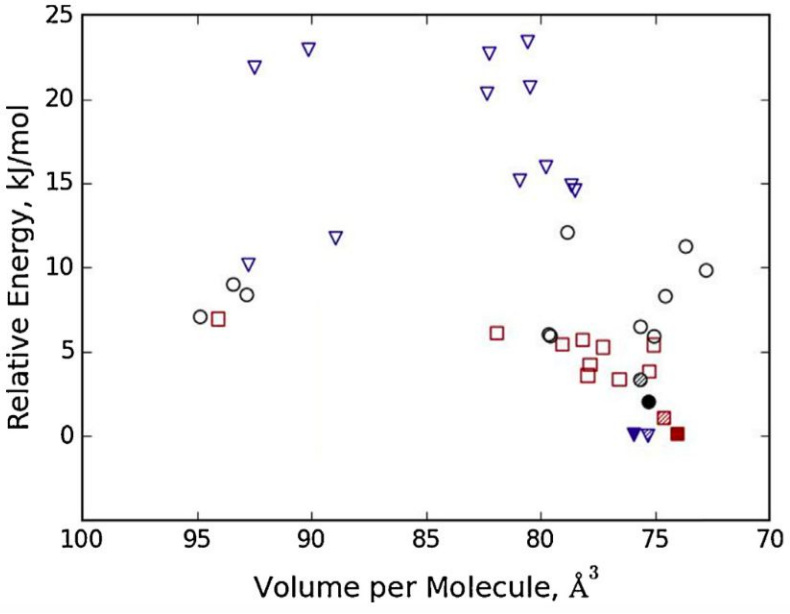
- Heredity operations
- Random members to increase diversity
- Similarity measurement via fingerprinting
- Remembering family history

USPEX J. Chem. Phys. 124, 244704 (2006)

Too big to explore with DFT:

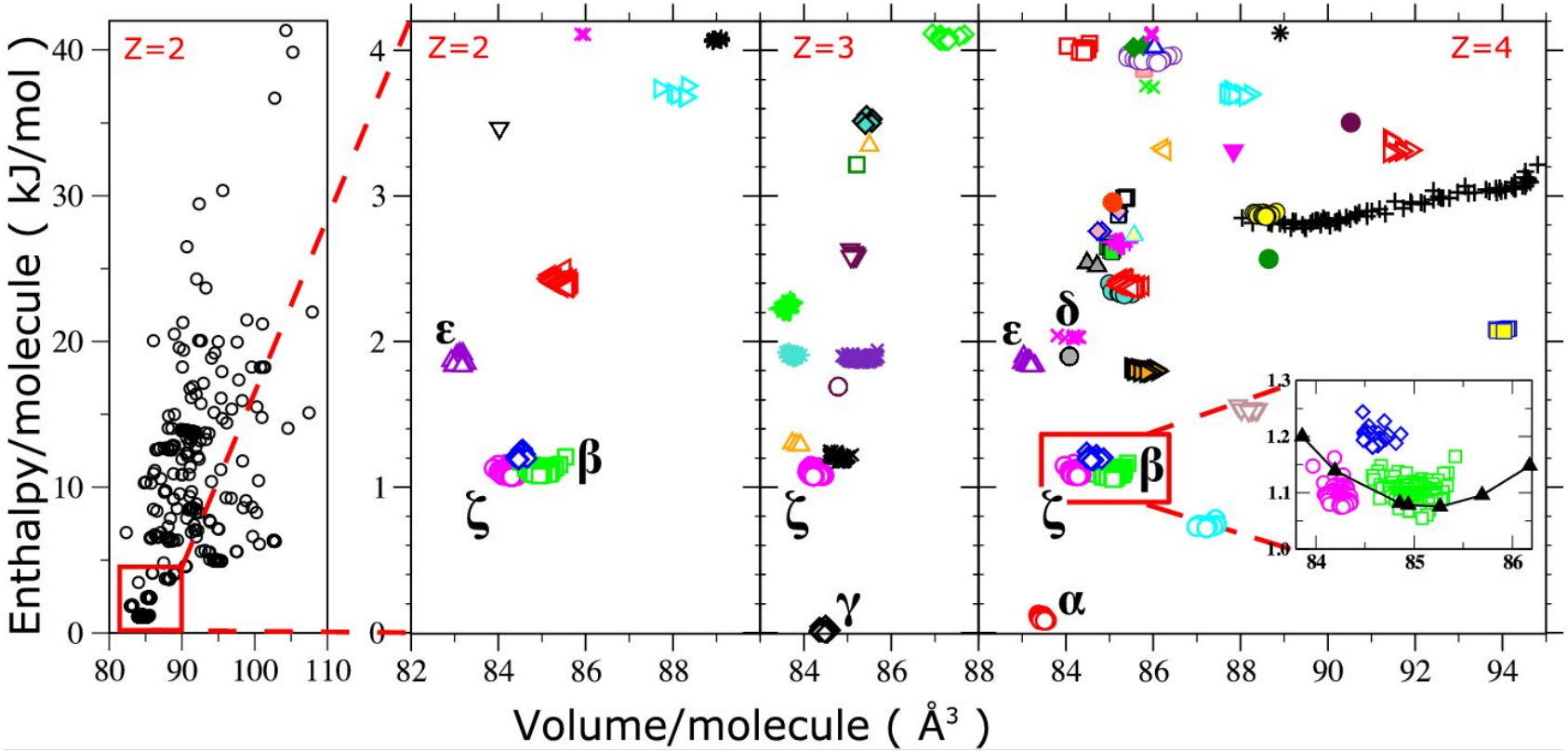
- Many more structures, shorter time than published work
- But cannot find the *alpha* phase !
- (A published work also could not find the alpha phase - visual inspection :-/)

- Found alpha by mimicking nature: “Random” musicians are not random: Symmetry group probability distribution mimics the one of nature using DB of molecular crystals. - “Data mining”



Lund et al, Chem Phys Lett 262 (2015)

GLY polymorphism



Zeta GLY: what did we do right? Accurate energy & volume

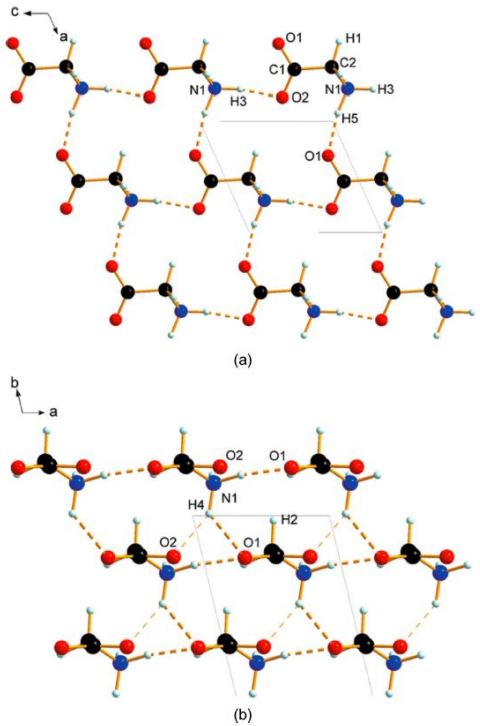
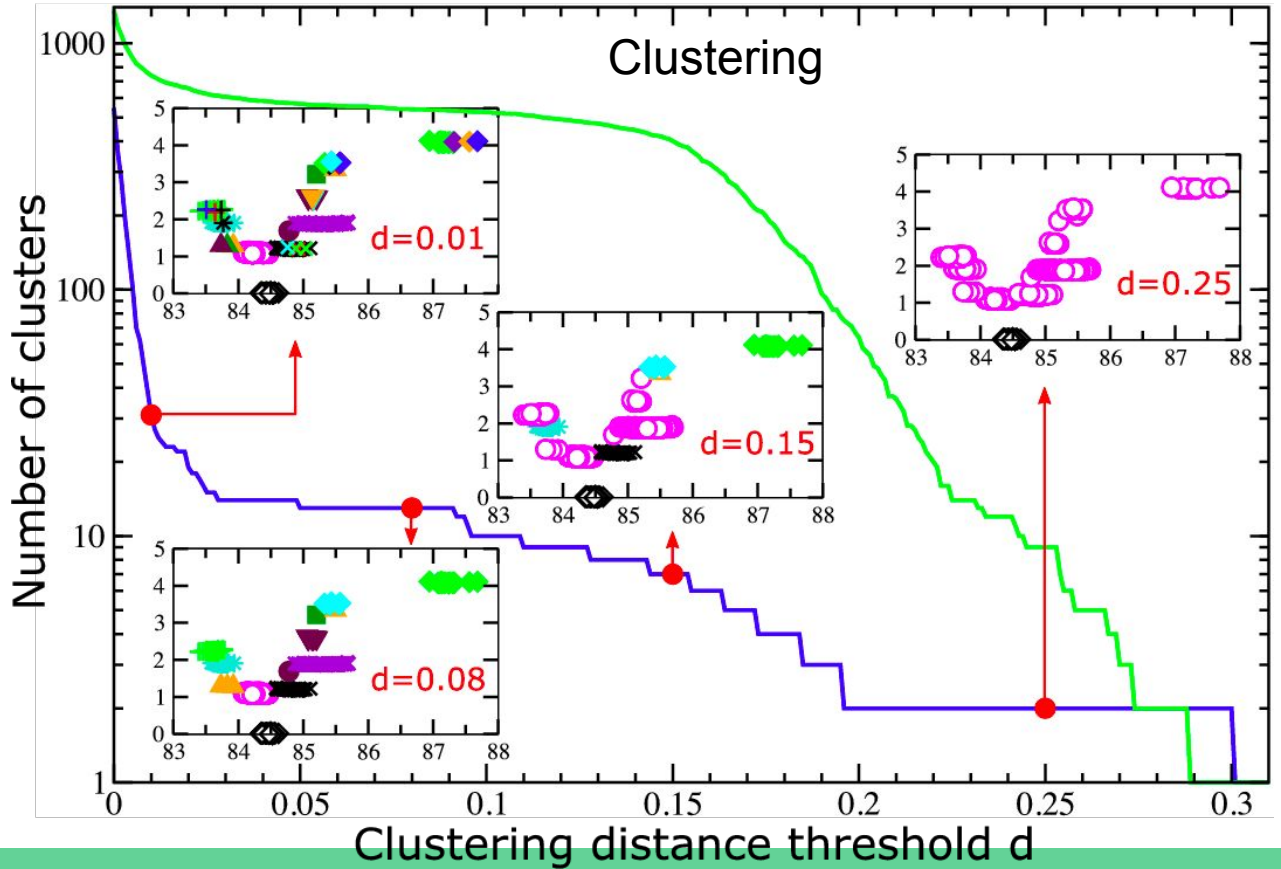


Figure 3
Intermolecular interactions in ζ-glycine. (*a*) Layers formed in the *ac* plane, viewed along **b**. (*b*) Stacking of the layers, viewed along **c**.




Oculus

Neural Networks for CSP

4.Where ML can help:

- Accelerate exploration
- Accelerate characterization
- Recognize descriptors, propose new structures with desired properties

(Just scratched the surface: Reinf. Learning, Gen. Models, Autoencoders etc.)

Learn a *molecular* forcefield for Z=1, transfer to Z>1

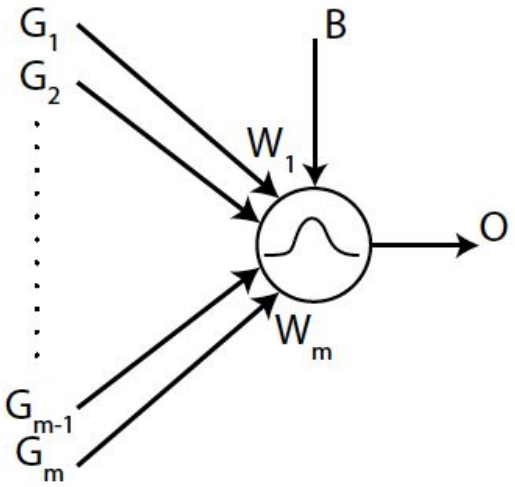
Learn *atomistic* forcefield, transfer to other molecules

Learn *relaxation moves*
For big systems with many atoms, far from the equilibrium positions

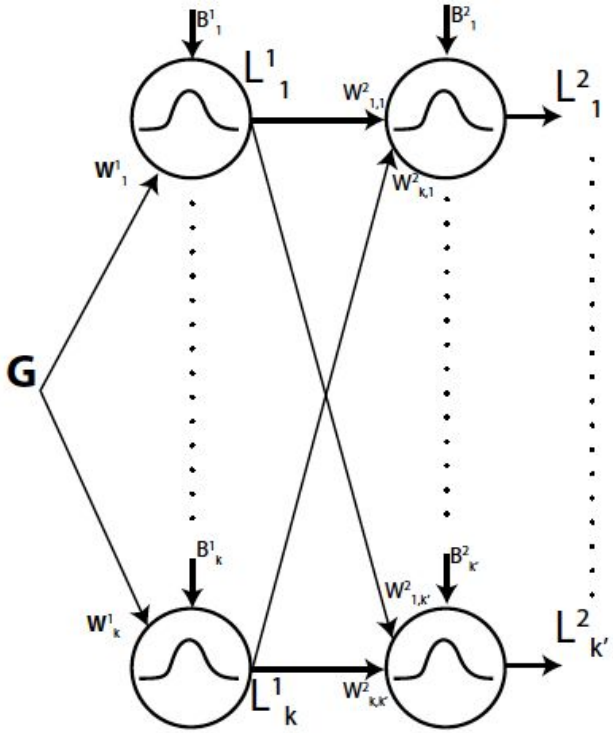
To keep in mind:

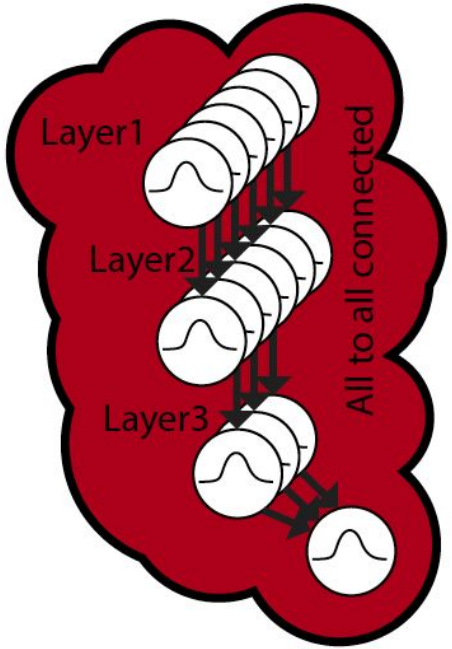
- Representation + Architecture + Target are all related
- Model training vs evaluation costs can be very different
- Transferability is good, but data keep changing

A2A, FF network with GLY

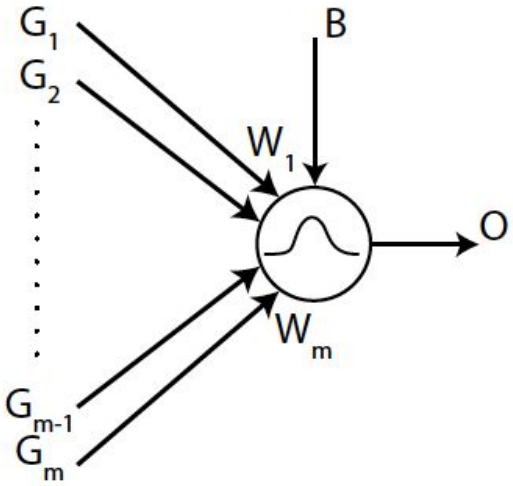


$$O = \exp\left[-\left(\sum_m G_m W_m + B\right)^2\right]$$

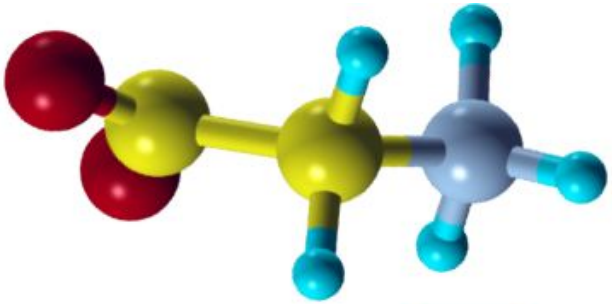


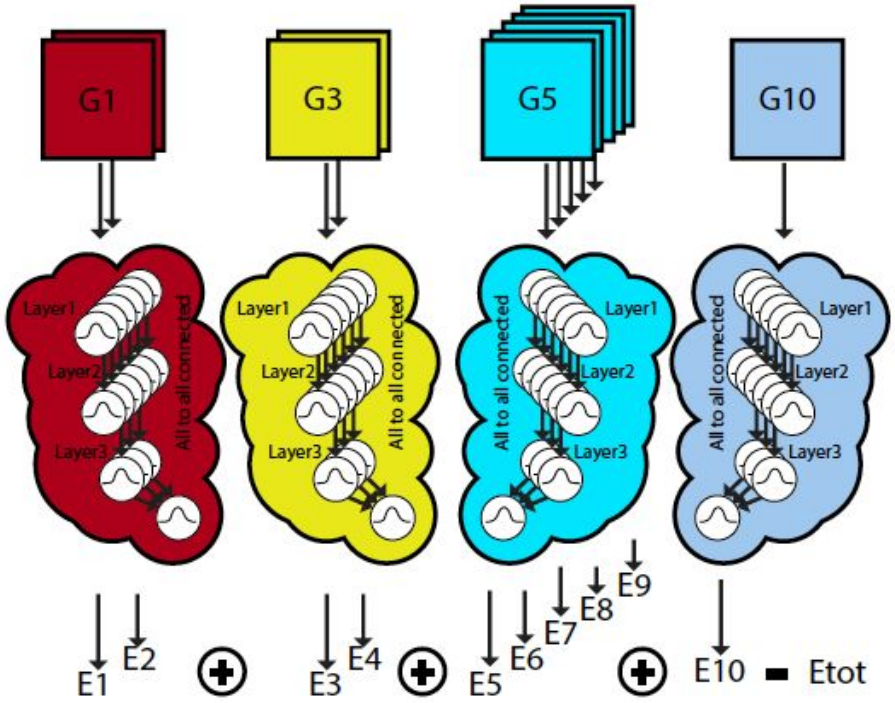


A2A, FF network with GLY



$$O = \exp\left[-\left(\sum_m G_m W_m + B\right)^2\right]$$





Representation

$$G^R_{m,s;i} = \sum_{i \neq j}^{\text{All atoms kind s}} e^{-\eta(r_{ij}-R_m)^2} f_c(r_{ij})$$

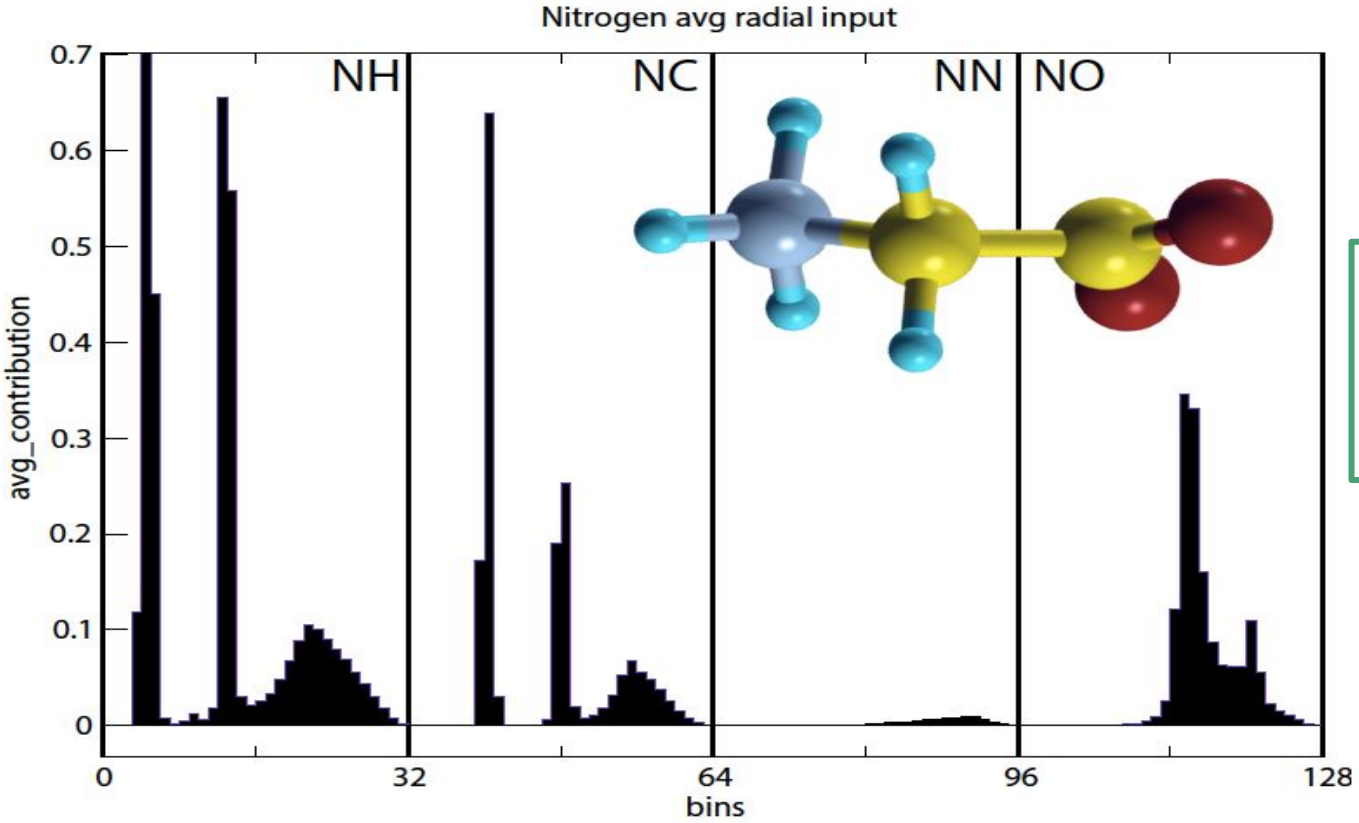
$$f_c(r_{ij}) = \begin{cases} 0.5\left[\cos\left(\frac{\pi r_{ij}}{R_c}\right)+1\right] & \text{if } r_{ij} \leq R_c \\ 0 & \text{if } r_{ij} \geq R_c \end{cases}$$

R0=0.5A , Rc= 4.6A
32 bins per pair:
32x4=128 parameters

J. Behler and M. Parrinello, PRL, 98.14 (2007).

Smith et al, Chem Sci 8 3192 (2017)
DOI: 10.1039/c6sc05720a

Average G-radial for N in GLY



R0=0.5A , Rc= 4.6A
32 bins per pair:
32x4=128 parameters

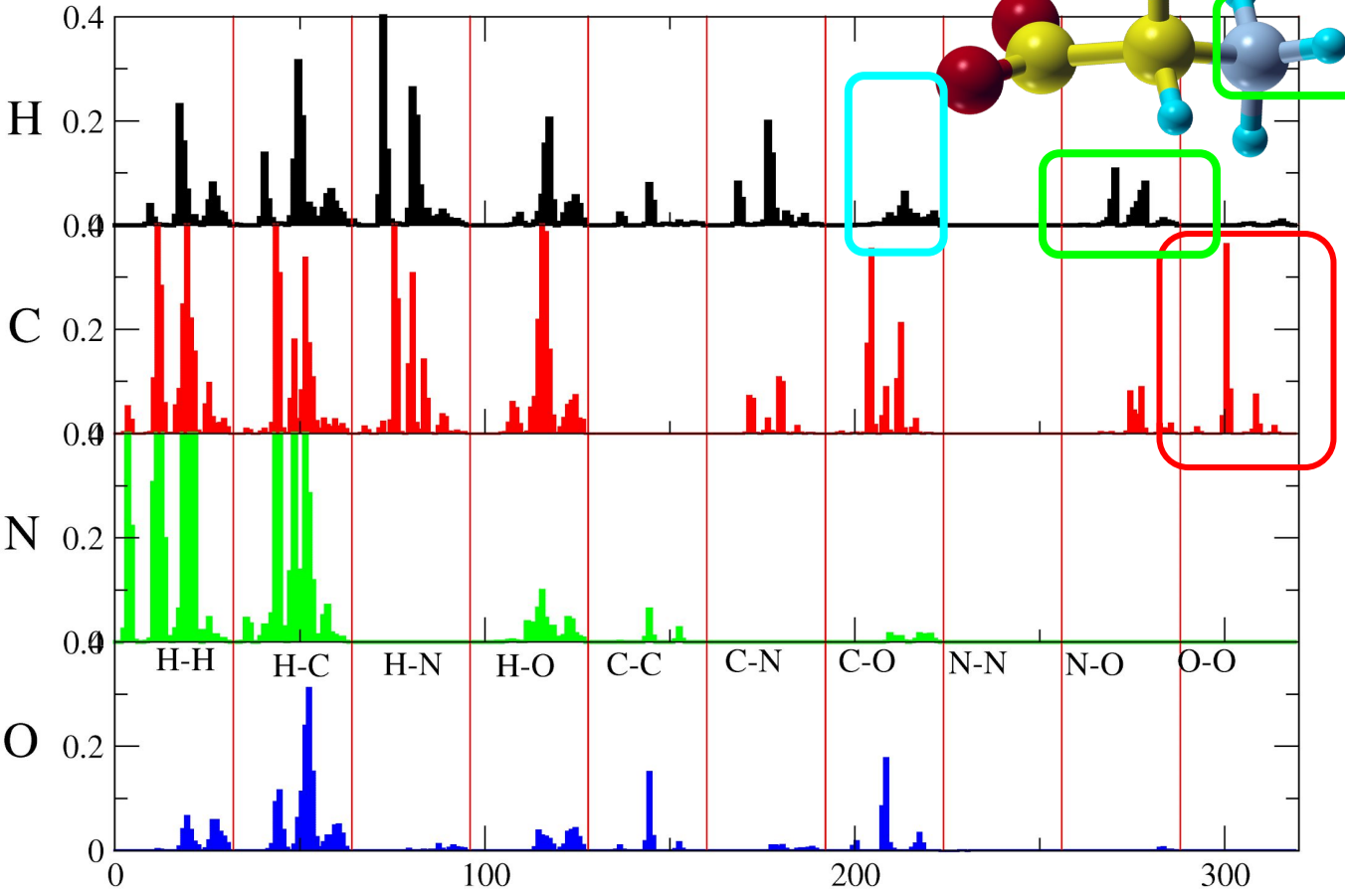
Representation

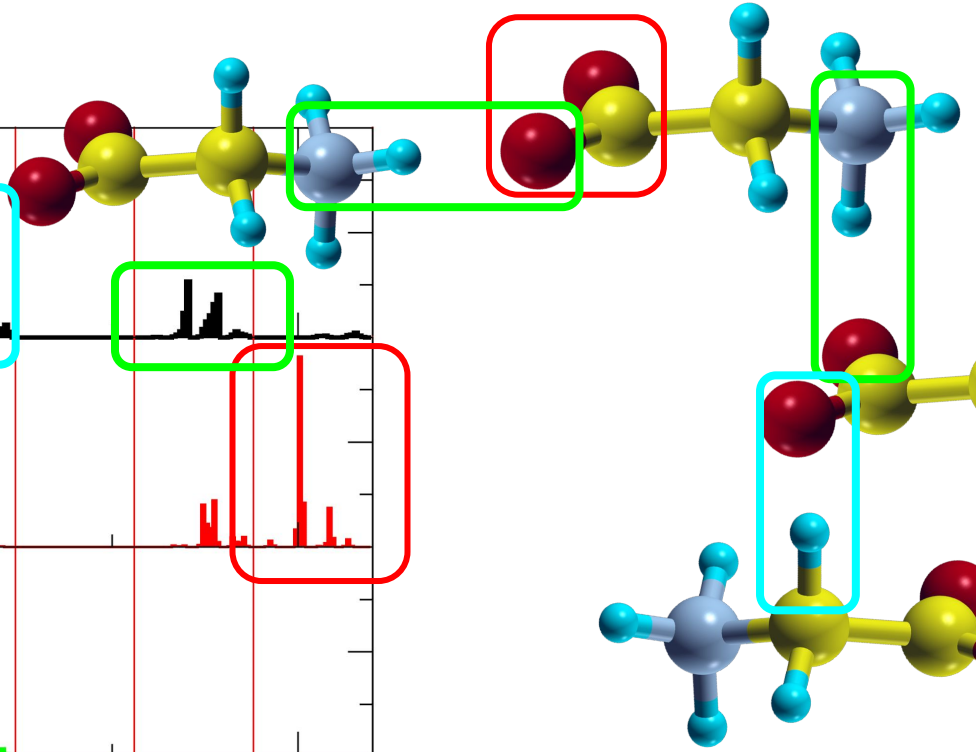
$$G^{A}_{n,m,s;i} = 2^{1-\xi} \sum_{\substack{\text{All atom of kind s} \\ j,k \neq i}} (1 + \lambda cos(\Theta_{ijk} - \Theta_n))^{\xi} \; e^{-\eta\left(\frac{r_{ij}+r_{ik}}{2} - R_m\right)^2} f_c(r_{ij}) f_c(r_{ik})$$

R0=0.5A , Rc= 3.1A
8 angular bin for each 8 radial bin
64 bins per trimer:
64x10=640 parameters

Smith et al, Chem Sci (2016)
DOI: 10.1039/c6sc05720a

Average G-angular in GLY



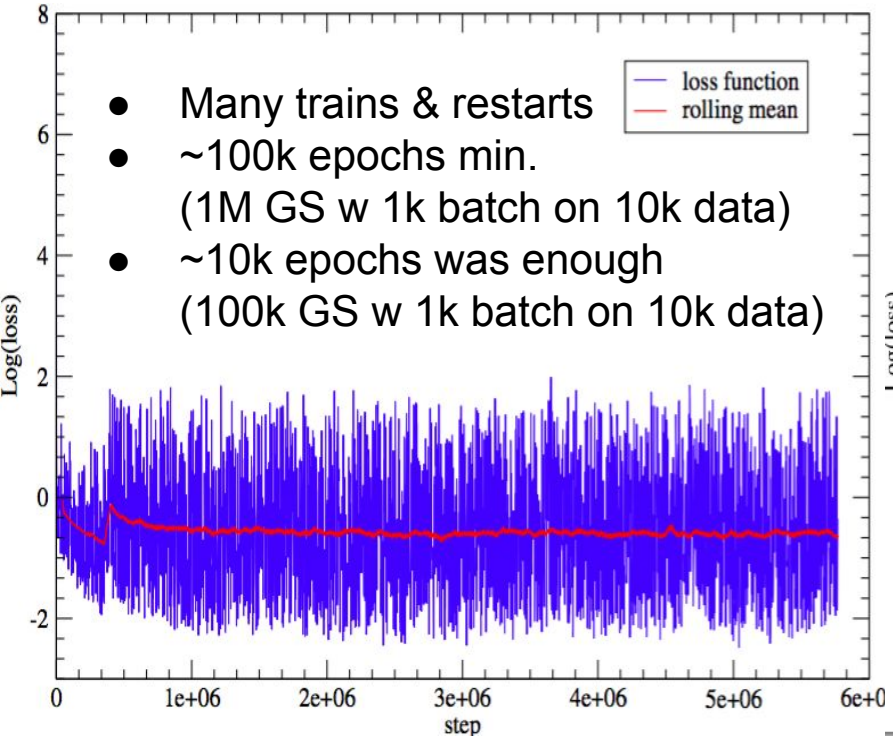


Halved version:
8 angular for 4 radial
32 per trimer
32x10=320 parameters

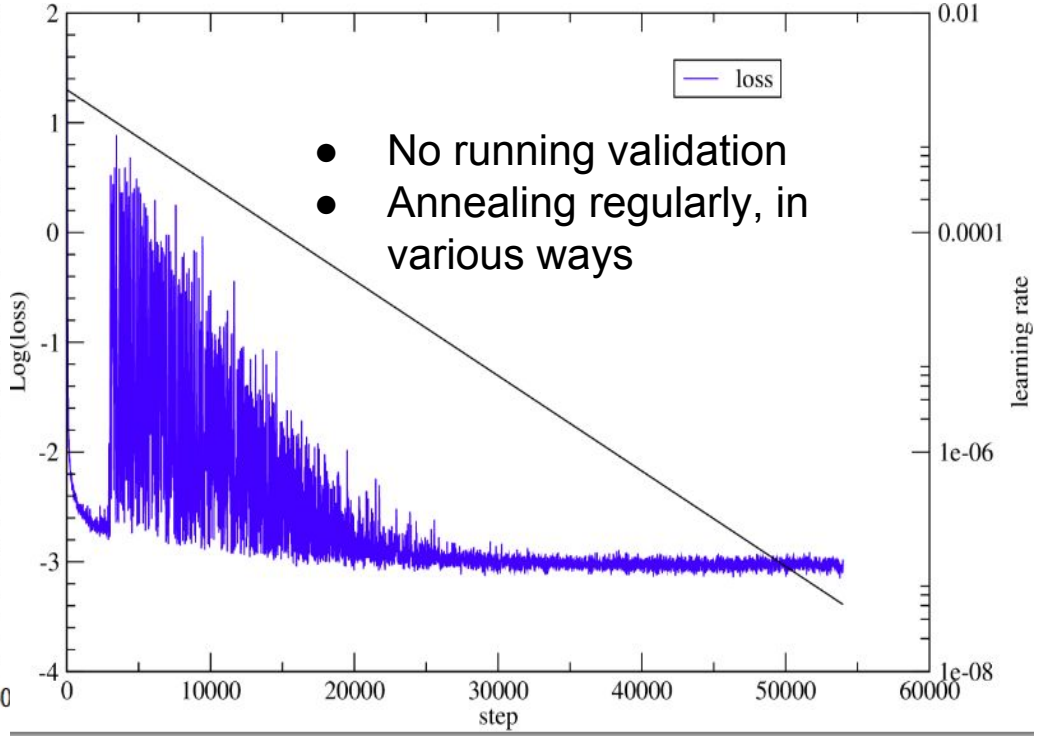
Remarks about this representation & CSP

- Representation may have *redundant* parts
- Representation may have *irrelevant* parts
- No need for atom centered representations
  - Bond centered, *molecule* centered representations
- No need for equally spaced representations
  - Allow custom gaussian centers
  - Allow gaussian centers to be learned
- 128+640 = 768-parameters per G-vector
- G-vector ➔ First layer width ≃ First layer of compression
  - 768 : 128 : 128 : 32 : 1
  - ~ 500k parameters for the whole network

Training the A2A-FF, modified BP - GLY data



- Many trains & restarts
- ~100k epochs min.
  (1M GS w 1k batch on 10k data)
- ~10k epochs was enough
  (100k GS w 1k batch on 10k data)



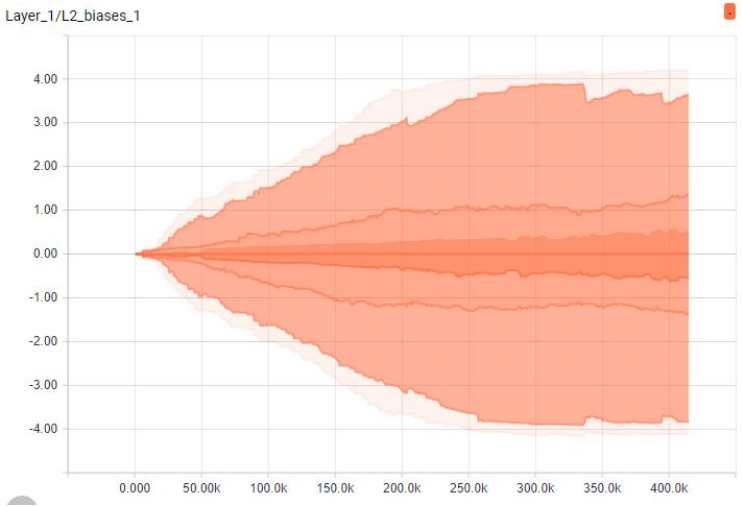
- No running validation
- Annealing regularly, in various ways

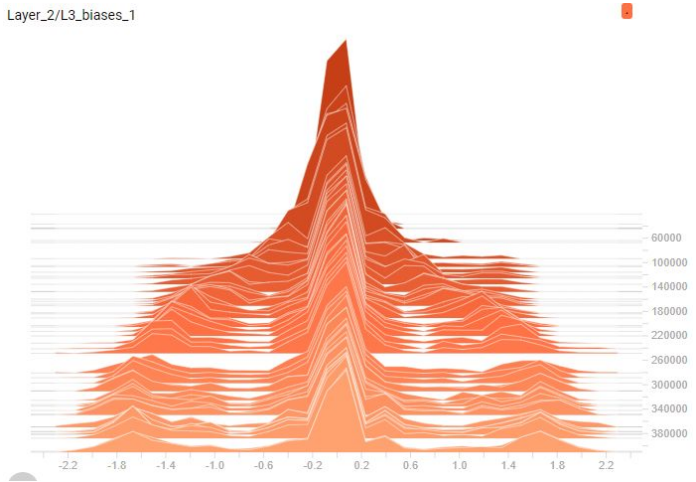
Stochastic gradient descent with Adam Opt.

T ~ learning rate / batch size

What network looks like as it trains

Layer 3 biases in a constant-LR run

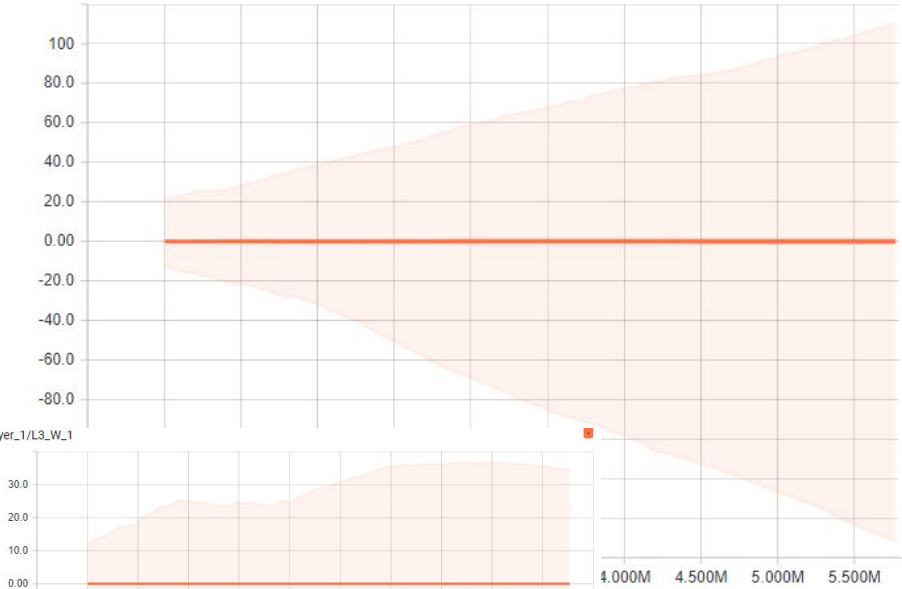




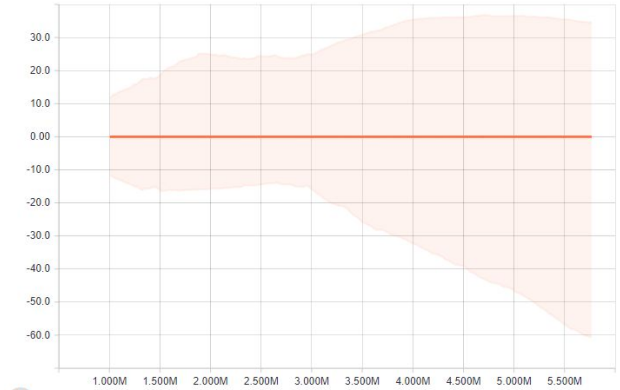
Tensorflow, Abadi et al. (2015)

Training

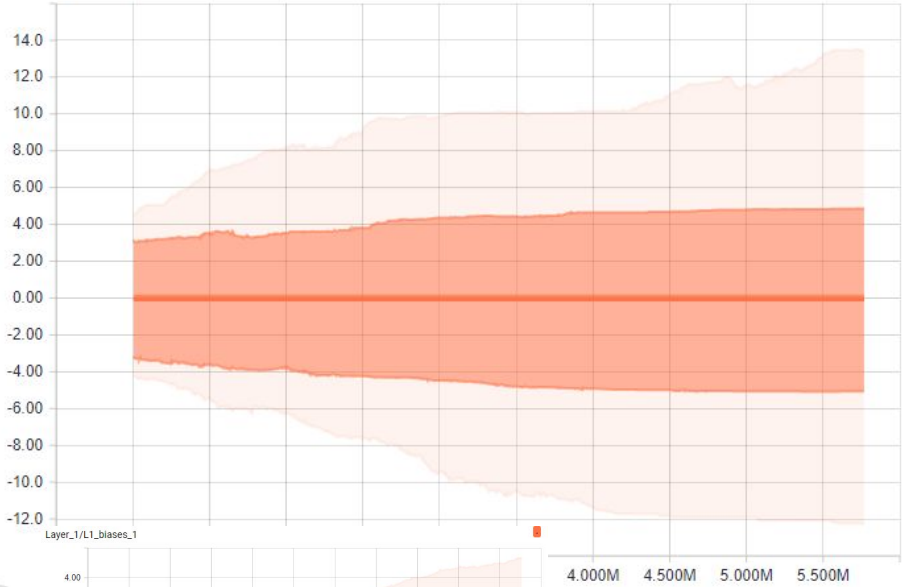
Layer_1/L2_W_1



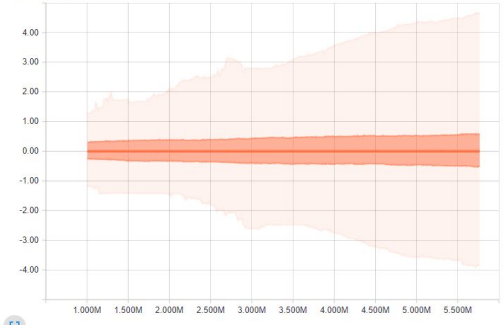
Layer_1/L3_W_1



Layer_1/L2_biases_1



Layer_1/L1_biases_1



Training

Line: +/- 2 mRy per atom.

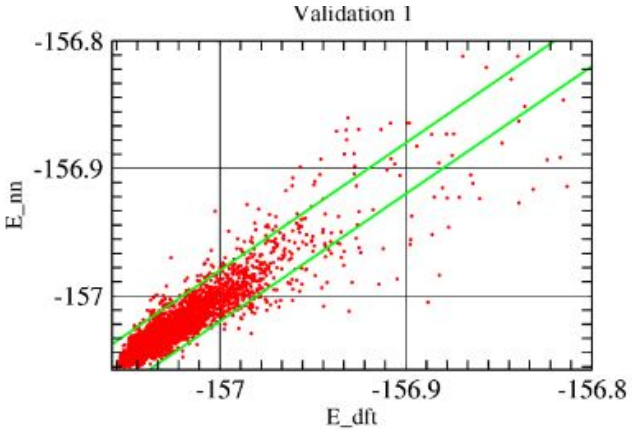
~10k data, ~500k parameters

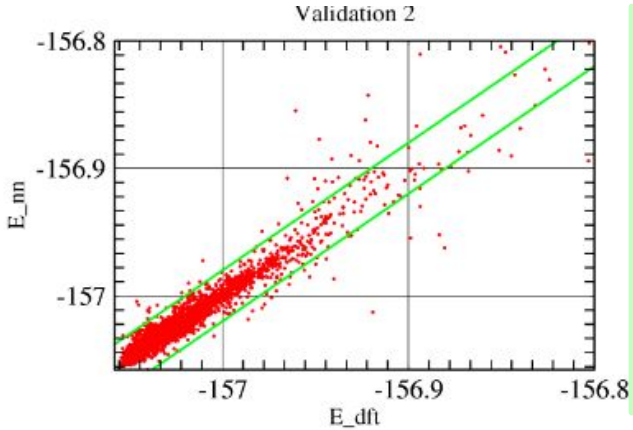
Over-fitting to the training data in an obvious way is not that easy.

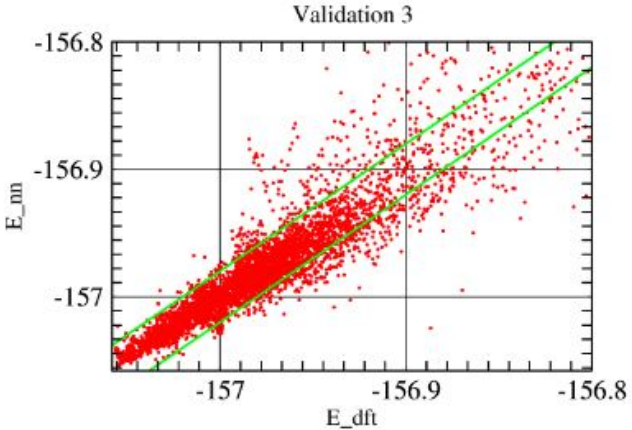
RMSE on training set:
0.33 mRy/atom @step:130k
0.41 mRy/atom @step:5M

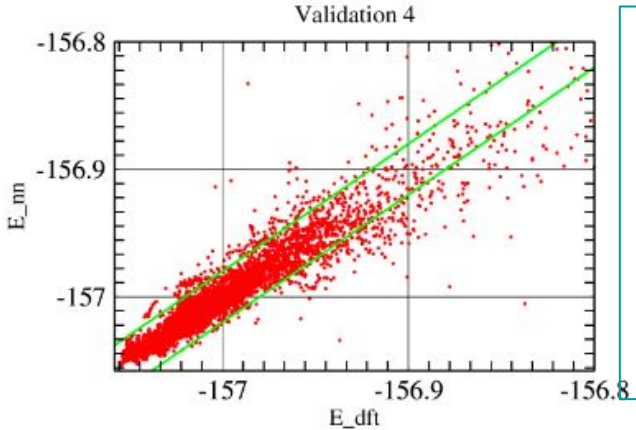
Target ***training*** accuracy
per molecule
1kcal/mol
(~43meV ~3.2mRy)
Roughly
~4meV per atom
~**0.3**mRy per atom

Training on Z=2, Validating on all Z=1,2,3,4







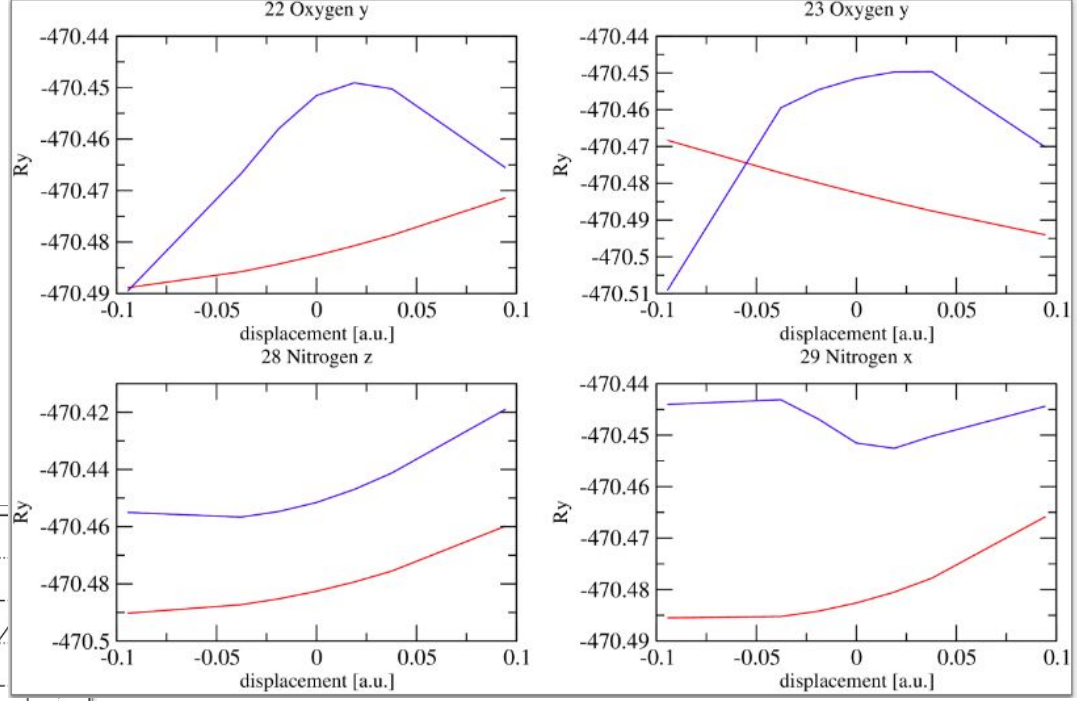


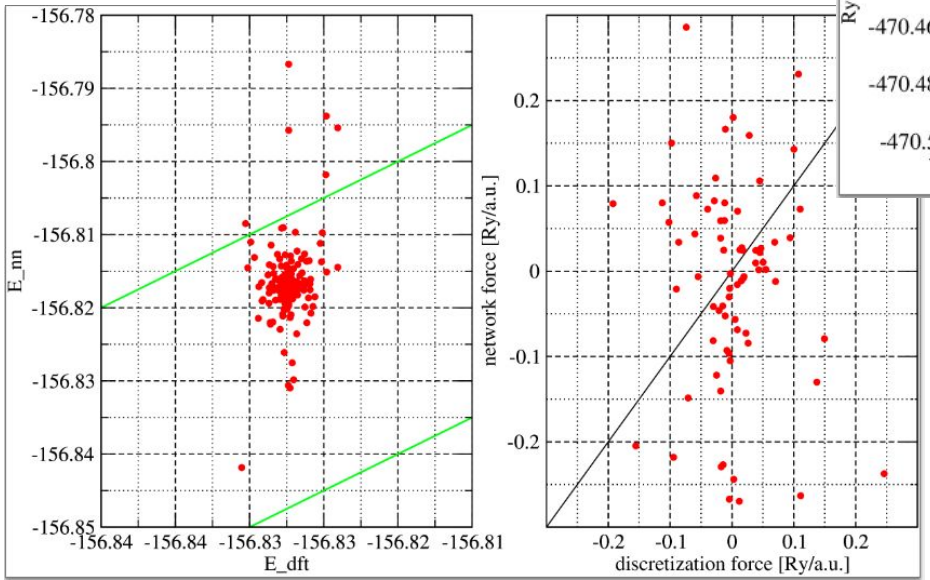
RMSE @2M steps (@6M>L1)
(in mRy/a)
1.21 (1.51 > 1.61) **Z=1**
1.04 (2.13 > 2.19) **Z=2**
4.86 (6.14 > 5.84) **Z=3**
4.06 (4.98 > 4.93) **Z=4**

Target ***extensibility***
accuracy per molecule
1.8kcal/mol
(~80meV ~5.8mRy)
~8meV per atom
~**0.6**mRy per atom

Forces

Fitting energies
Forces form finite differences
- Terrible :-(
- Unsampled atomic displacements
- Hope - at least continuous?





Bad forces also for the training set.
C or H slightly better but not always

Force RMSE~ 0.100Ry/au
(100 x acceptable threshold)

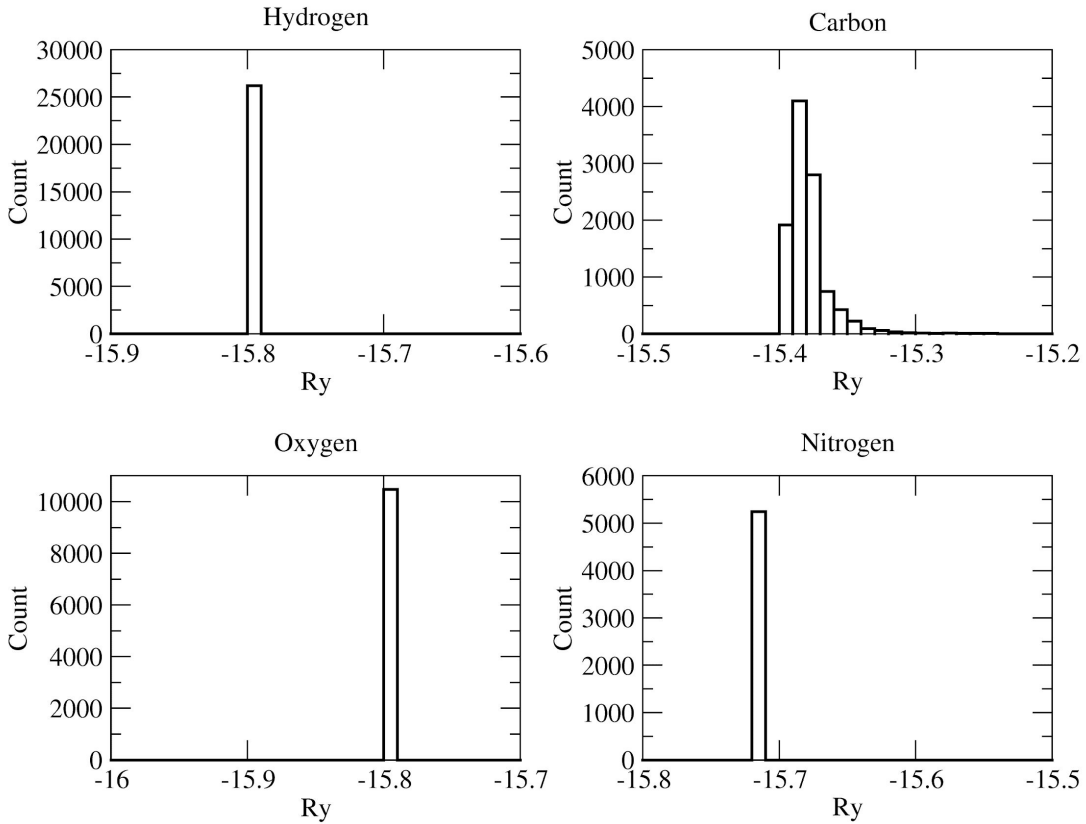
What does the network learn?

All the variation between different crystal structures is expressed via Carbon atom alone.

Why? Limited dataset (one stoichiometry) + redundancy in G-vectors

Does it matter? Bad for information transferability between systems, but not significant for single system CSP.

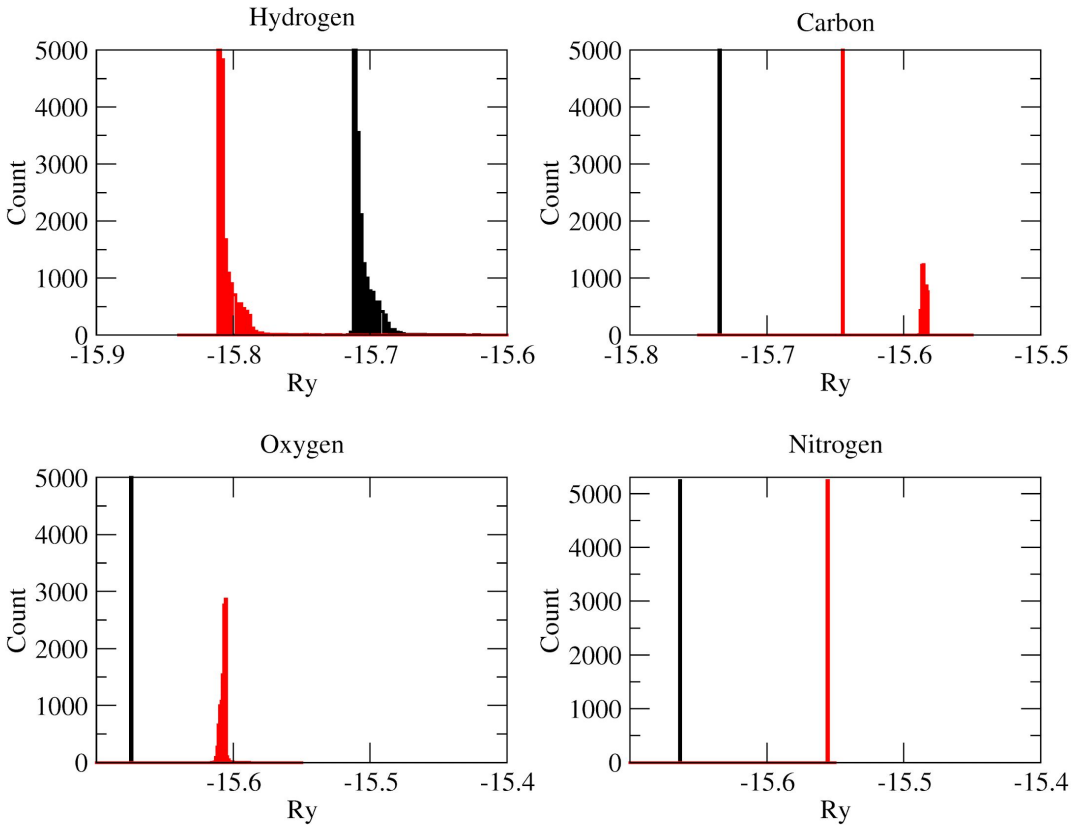
Atomic contribution to energy:



What does the network learn?

- Restart changes the “informant” atom
- Copying weights may increase variability
  (Before: black, After: red)
- Two things affect this behavior:
  - Reducing Rc
  - Shrinking network
    - Energies get better, (0.5mRy/atom)
      Forces are still 10x threshold

Atomic contribution to energy:



What we learned from A2A networks + GLY:

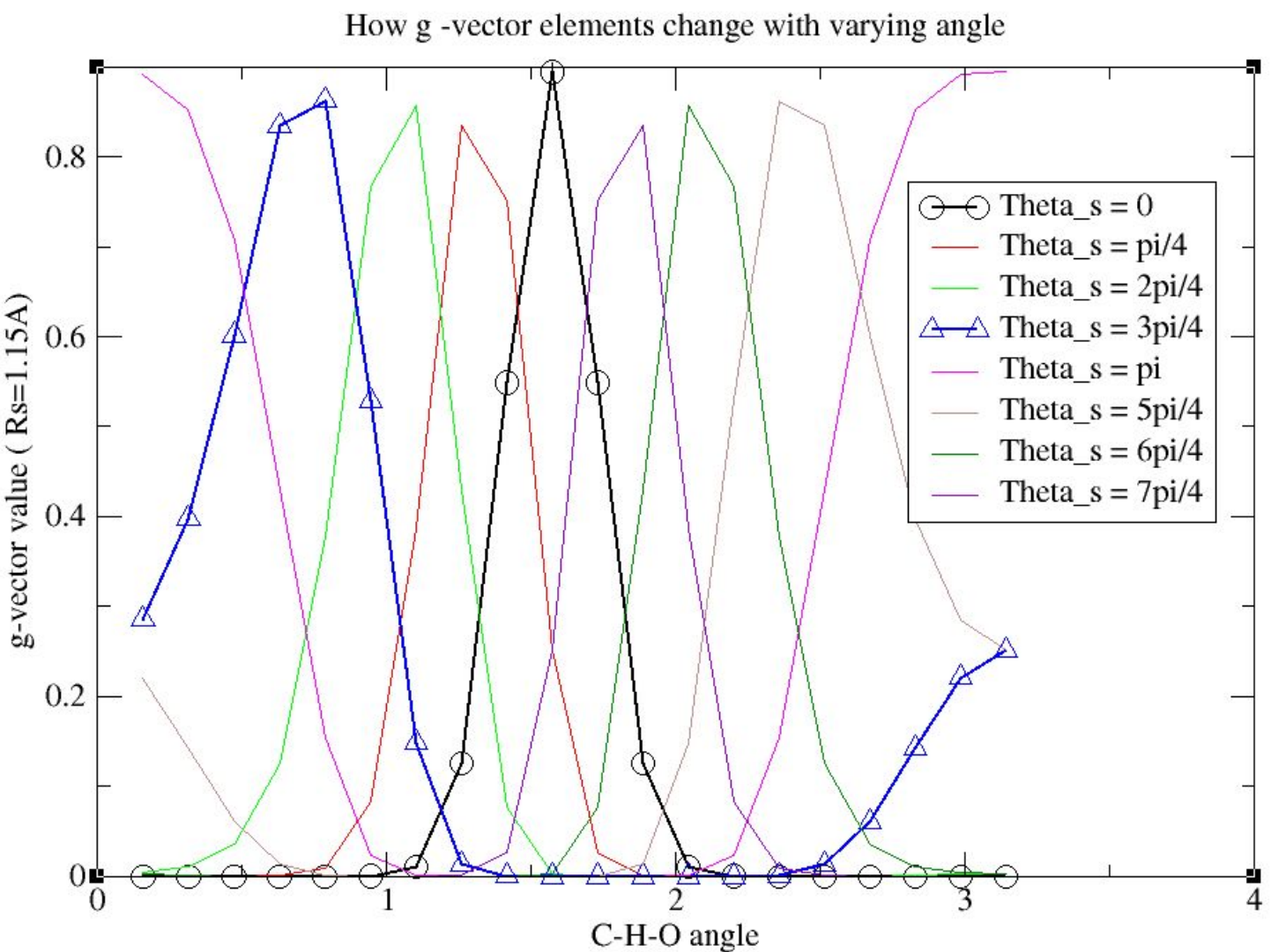
- 10k data, 500k parameters: underfitting & overfitting
- BP representation too redundant/expressive for CSP data
  + Hyperparameter optimization may be necessary for each CSP
  + Learning dynamics can be tricky
- Upon shrinking network (down to 10k parameters)
  Good enough for mild acceleration (0.5mRy/atom),
  But better forces needed for significant improvement

PANNA: TF based, cpu/gpu, allows different network per atom, freezing the dyn.

G-vectors

Code-share reveals a bug:

The reason why so many G-bins so much data so long training was needed?



Reducing error with increasing data?

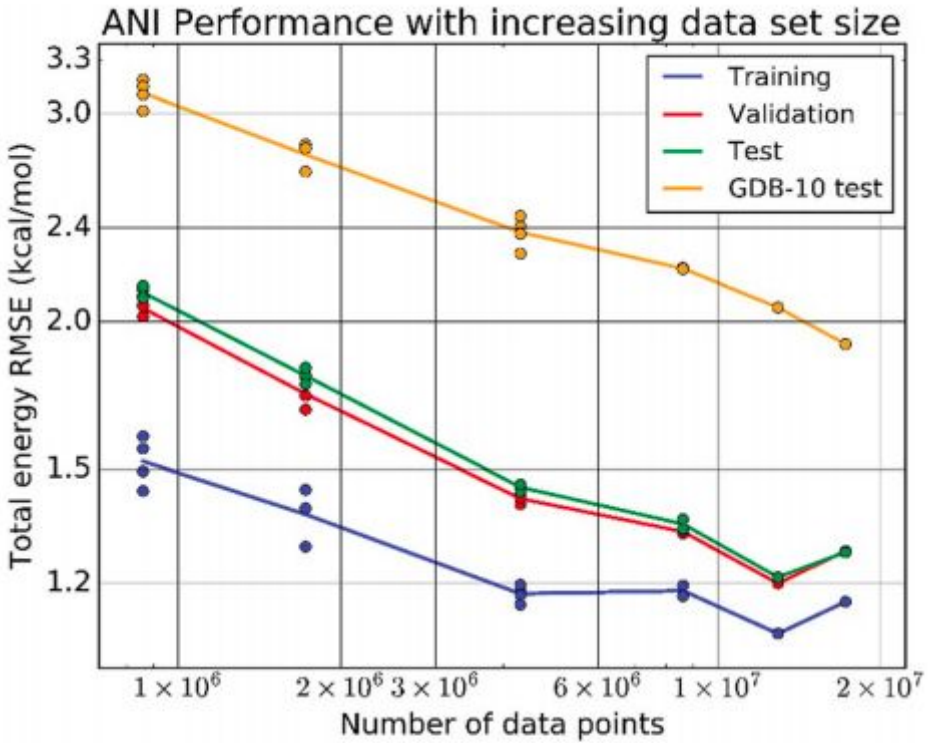


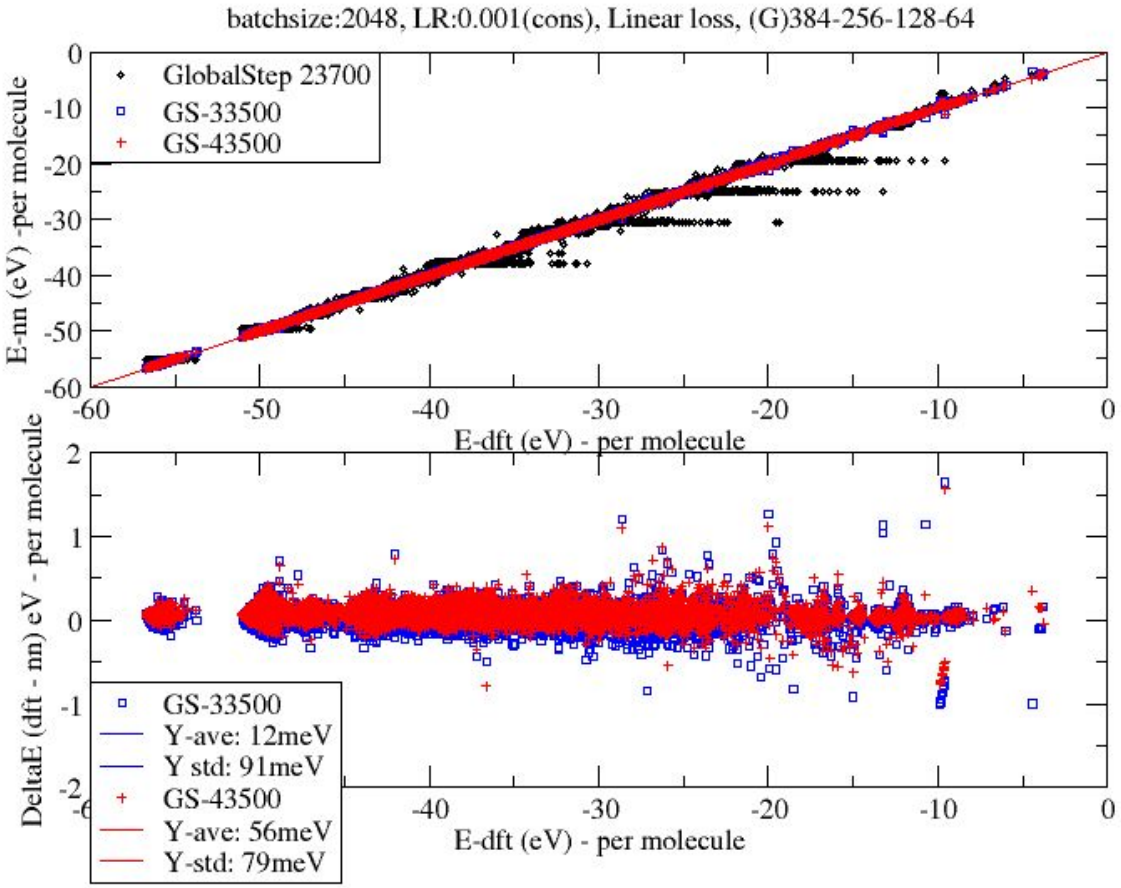
Fig. 3 Log–log plots of the training, validation, testing, and a random

Training with molecular data

Stoichiometrically varying 100k
(out of 800k ANI set)
~500k parameters
384:256:128:64:1

Target validation accuracy
per molecule
2kcal/mol
(~90meV)

OK energetics
much less data.



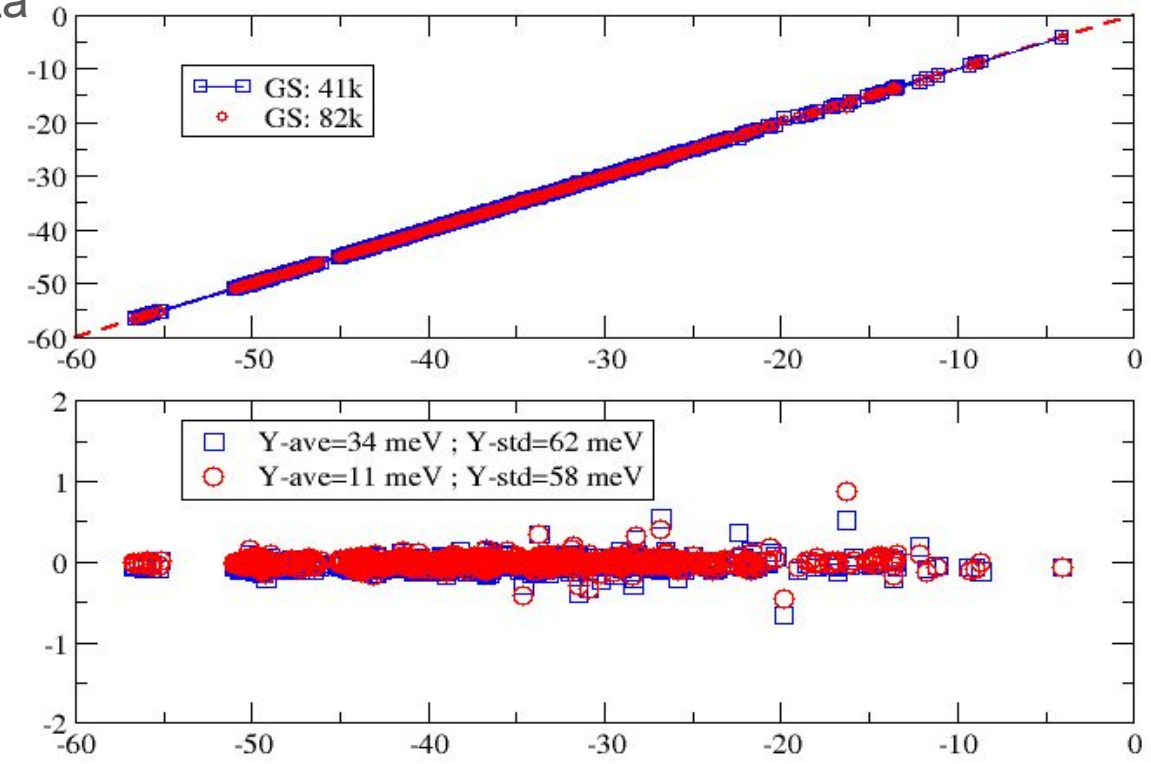
Training with molecular data

Stoichiometrically varying 10k data
(out of 800k ANI set)
~500k parameters
384:256:128:64:1

Target validation accuracy
per molecule
2kcal/mol
(~90meV)

OK energetics
much much less data.

Same network as before. Train on 8k, validate on 1k



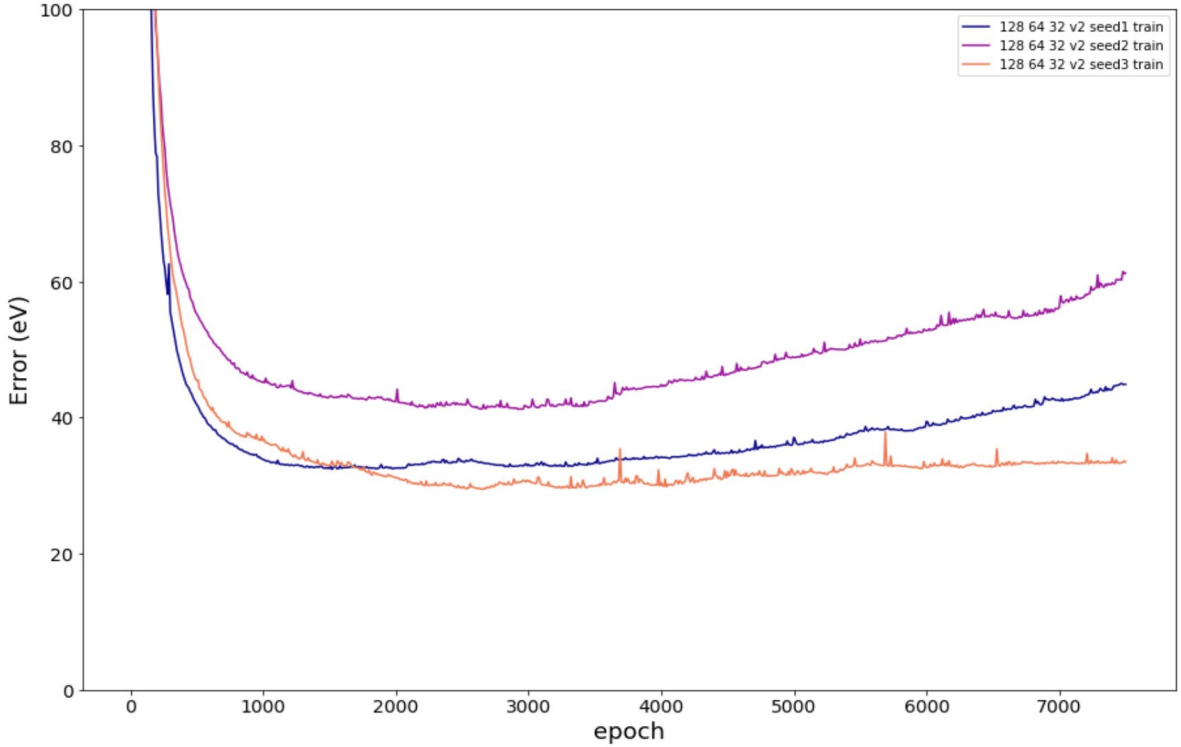
Another comparison

A C++ code, written from scratc

Tested on Silicon, Carbon

Original BP symmetry functions

No mini-batching
>whole data at each step (10k dataset)



Kolb et al. Sci.Rep. 1192 (2017)

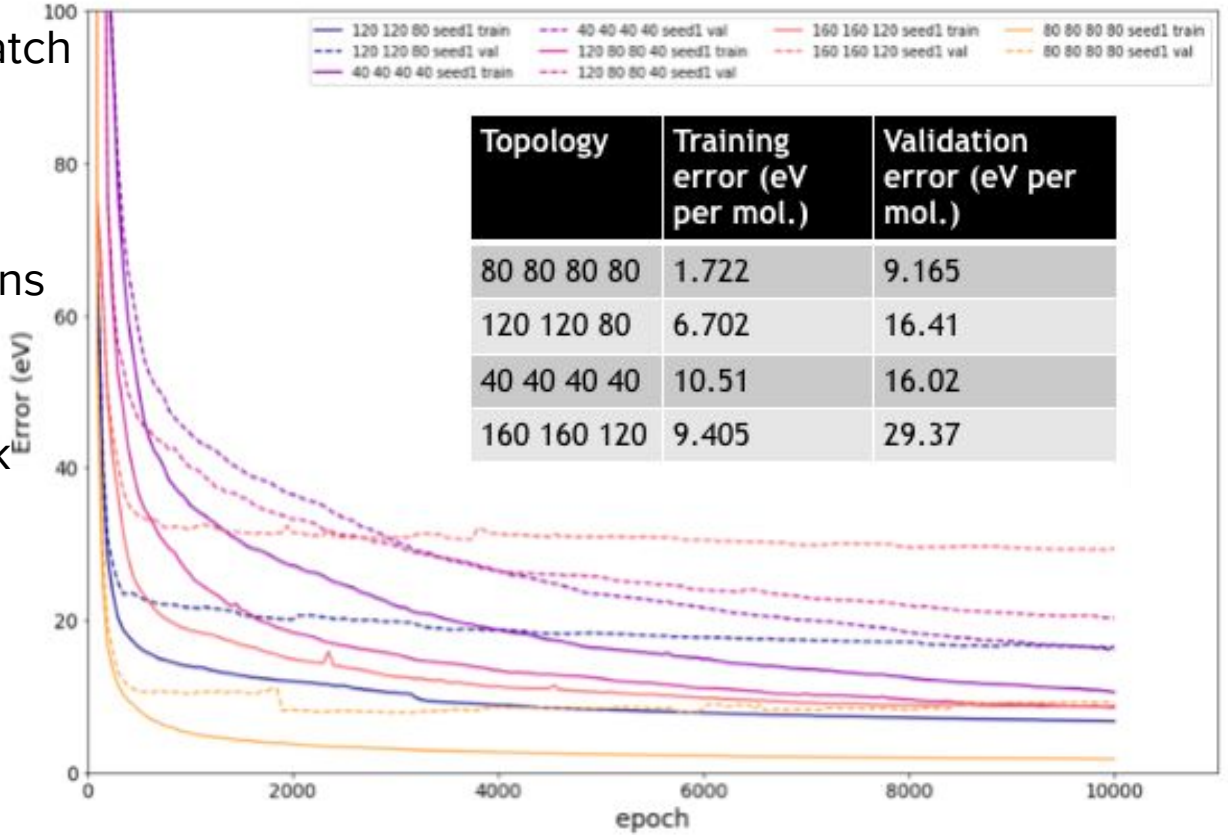
Another comparison

A C++ code, written from scratch

Tested on Silicon, Carbon

Original BP symmetry functions

No mini-batching
>whole data at each step (10k dataset)



Topology	Training error (eV per mol.)	Validation error (eV per mol.)
80 80 80 80	1.722	9.165
120 120 80	6.702	16.41
40 40 40 40	10.51	16.02
160 160 120	9.405	29.37

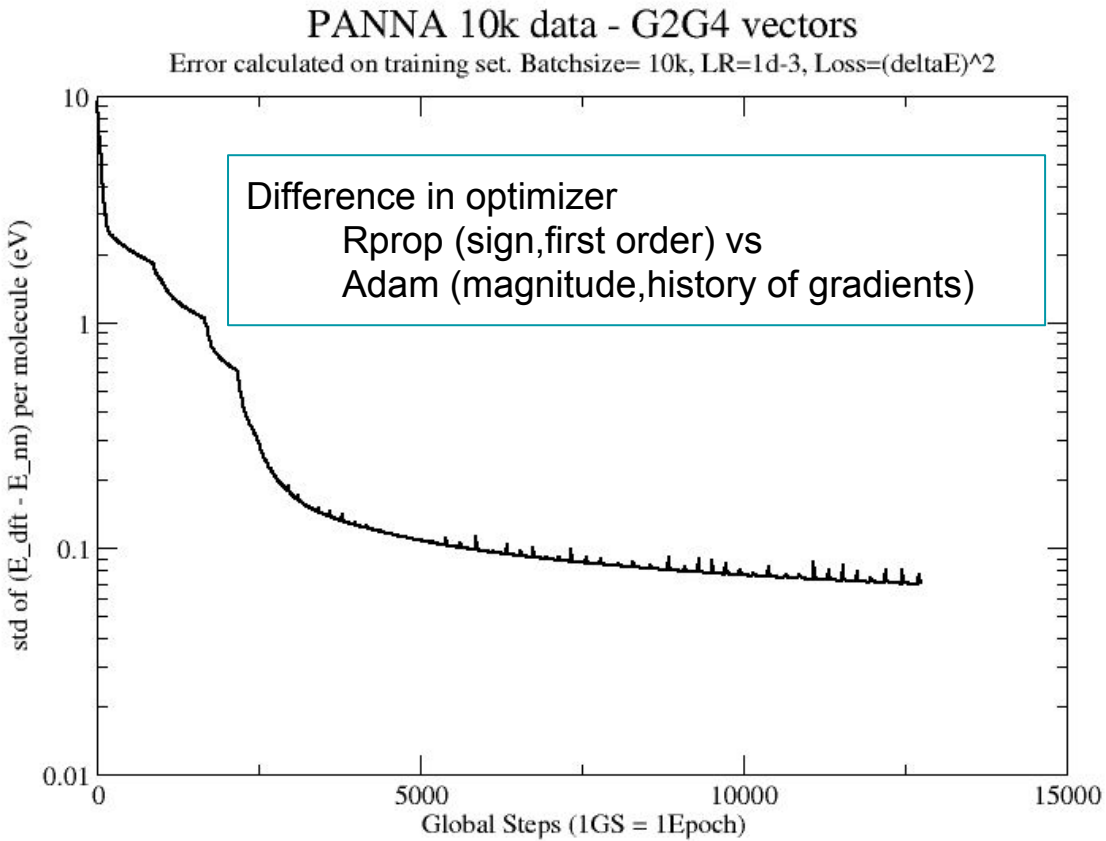
Another comparison

What is the culprit?
Modified PANNA

Original BP symmetry functions

Mini-batch size=whole data

Hits the target validation accuracy (~90meV) at 10k steps



What else we learnt from A2A networks + ANI data

NN acceleration of CSP is data-wise feasible

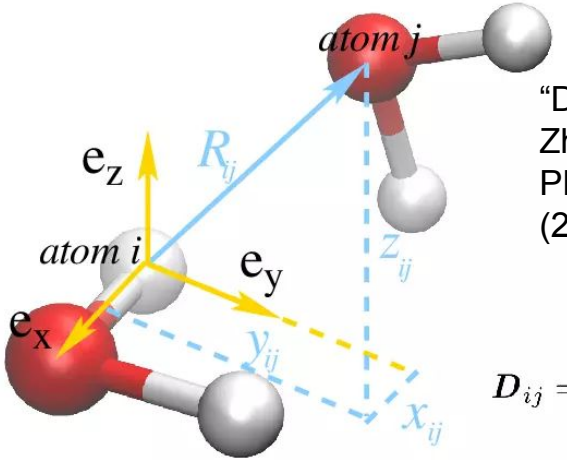
Training dynamics is not straightforward - tools to monitor & manipulate dynamics

Reproducibility

Fingerprinting

Positions? Distances, angles?
Radial distribution function?

- Ab initio, sometimes-reversible representation of the crystal
- Redundant & not every bin is equally important.
- Doesn't have built-in hierarchical comparison flexibility
- Differentiable



"DeepMD" Zhang et al. PRL 120 (2017)

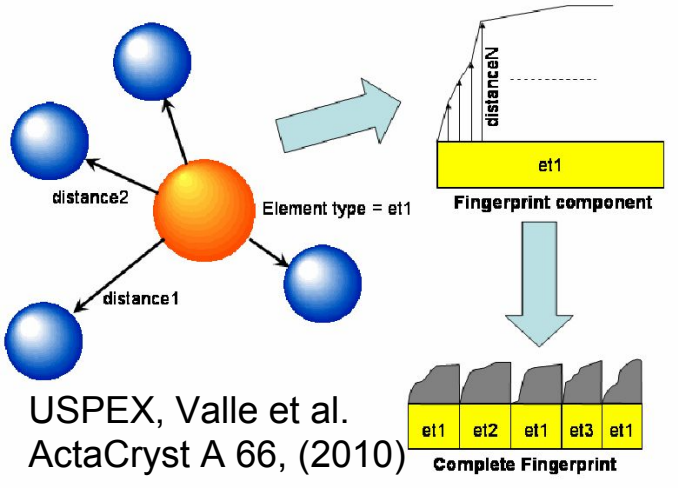
$$\boldsymbol{D}_{ij} = \{1/R_{ij}\}$$

or

$$\boldsymbol{D}_{ij} = \{1/R_{ij}, x_{ij}/R_{ij}^2, y_{ij}/R_{ij}^2, z_{ij}/R_{ij}^2\}$$

$$\tilde{k}(\mathscr{X},\mathscr{X}') = \int \mathrm{d}\hat{R}\left|\int \sum_{\alpha} \rho_{\mathscr{X}}^{\alpha}(\mathbf{r})\rho_{\mathscr{X}'}^{\alpha}(\hat{R}\mathbf{r})\mathrm{d}\mathbf{r}\right|^2$$

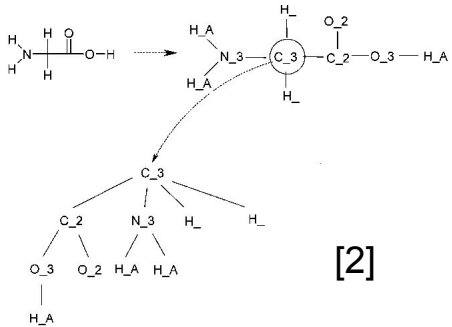
"SOAP" Bartok et al, PRB 87, 184115 (2013)



USPEX, Valle et al. ActaCryst A 66, (2010)

Fingerprinting

- Connectivity & graph based
  - Bag-of-Bonds[1]
  - Signature[2],
  - ConvNet[3],
  - SPRINT[4],
  - PIV[5]



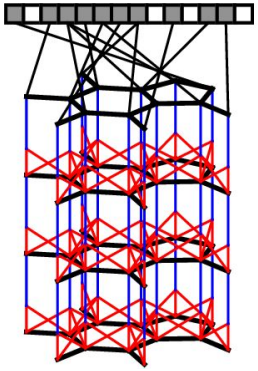
[2]

$^0\sigma$ = C_3 + C_2 + N_3 + O_3 + O_2 + 2H_ + 3H_A

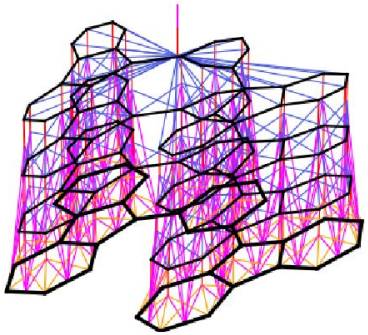
$^1\sigma$ = C_3 (C_2 N_3 H_ H_) + C_2 (C_3 O_3 O_2) + N_3 (C_3 H_A H_A) + O_3 (C_2 H_A) + O_2 (C_2) +2H_ (C_3) + 2H_A (N_3) + H_A (O_3)

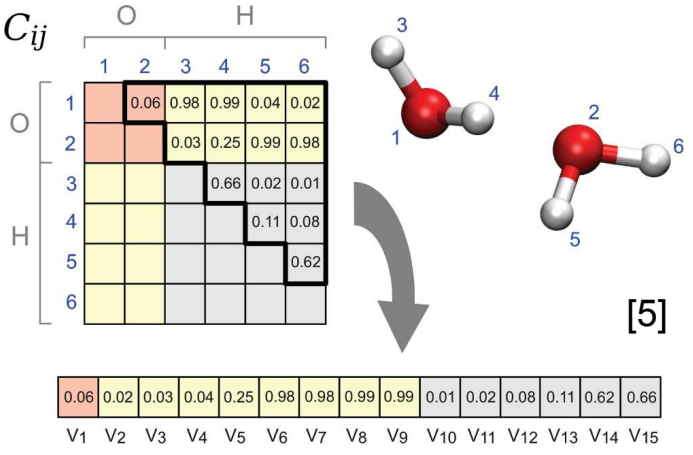
$^2\sigma$ = C_3 (C_2 (O_3 O_2) N_3 (H_A H_A) H_ H_) + C_2 (C_3 (N_3 H_ H_) O_3 (H_A) O_2) + N_3 (C_3 (C_2 H_ H_) H_A H_A) + O_3 (C_2 (C_3 O_2) H_A) + O_2 (C_2 (C_3 O_3)) + 2H_ (C_3 (C_2 N_3 H_)) + 2H_A (N_3 (C_3 H_A)) + H_A (O_3 (C_2))

$^3\sigma$ = C_3 (C_2 (O_3 (H_A) O_2) N_3 (H_A H_A) H_ H_) + C_2 (C_3 (N_3 (H_A H_A) H_ H_) O_3 (H_A) O_2) + N_3 (C_3 (C_2 (O_3 O_2) H_ H_) H_A H_A) + O_3 (C_2 (C_3 (N_3 H_ H_) O_2) H_A) + O_2 (C_2 (C_3 (N_3 H_ H_) O_3 (H_A))) + 2H_ (C_3 (C_2 (O_3 O_2) N_3 (H_A H_A) H_)) + 2H_A (N_3 (C_3 (C_2 H_ H_) H_A)) + H_A (O_3 (C_2 (C_3 O_2)))

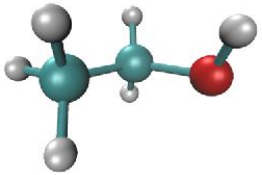


[3]

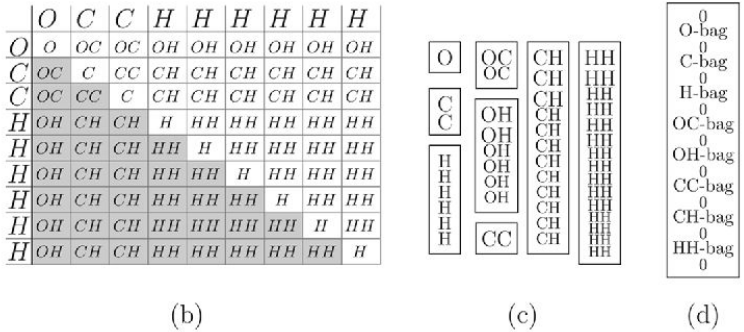




[5]



[1]



Unsupervised methods & representation

- A good representation -> good distance measure -> successful clustering
- “much of all intelligence is unsupervised clustering”
- If we could do successful clustering all the time, we could solve all classification problems
  (is it a metal? Is it a magnetic material? Is it an 2d-exfoliate-able material?)
  but also make good approximations on other properties as by expanding the property on clusters

Cluster Expansion for Li-Gr

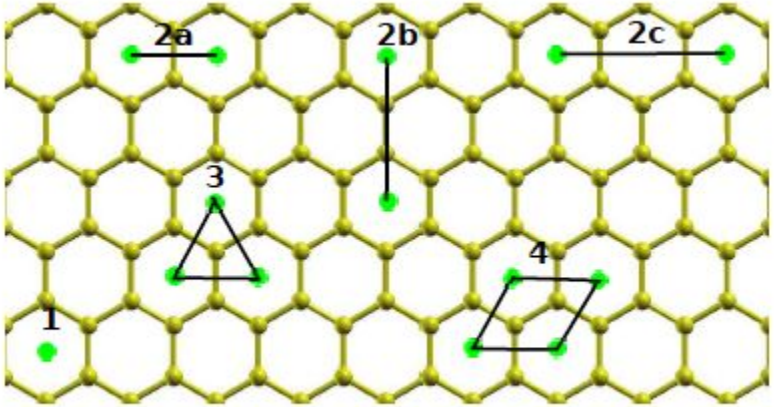


Figure 4: Figures in cluster expansion. (1) is the on-site figure (2) the 2-body figure of (2a) first neighbor (2b) second neighbor (2c) third neighbor (3) 3-body figure (4) 4-body figure

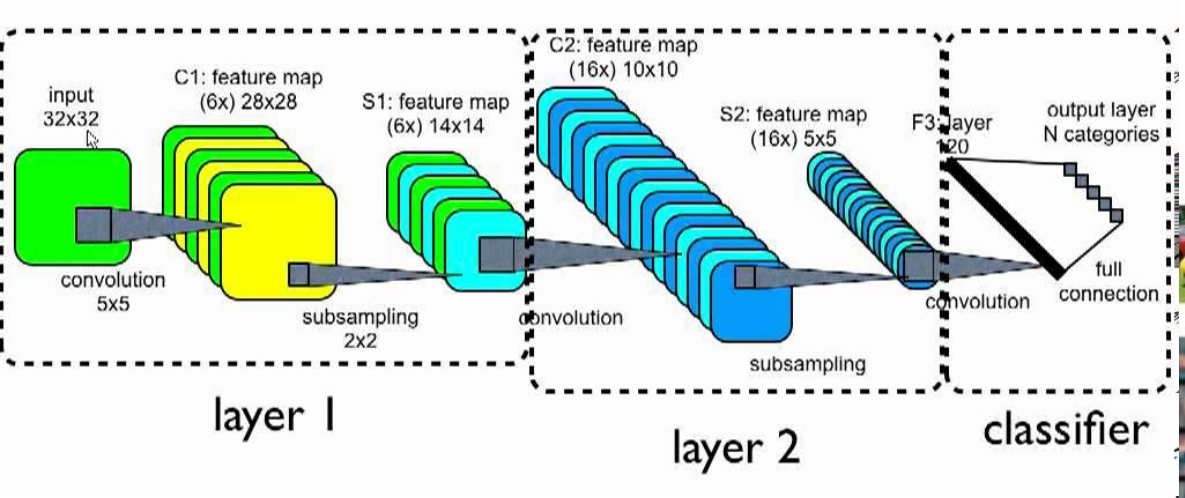
$$
\begin{aligned}
E_{CE}(C) &= J_1 \sum_{i \in f} c_i + \frac{1}{2} \sum_{p=a,b,c} \sum_{\{ij\} \in f} J_{2p} c_i c_j \\
&+ \frac{J_3}{6} \sum_{\{ijk\} \in f} c_i c_j c_k + \frac{J_4}{24} \sum_{\{ijkl\} \in f} c_i c_j c_k c_l \\
&+ \frac{J_d}{2} \sum_{ij, |\mathbf{r}_i - \mathbf{r}_j| > 2a_0} \frac{q_i q_j}{|\mathbf{r}_i - \mathbf{r}_j|^3} c_i c_j
\end{aligned}
$$

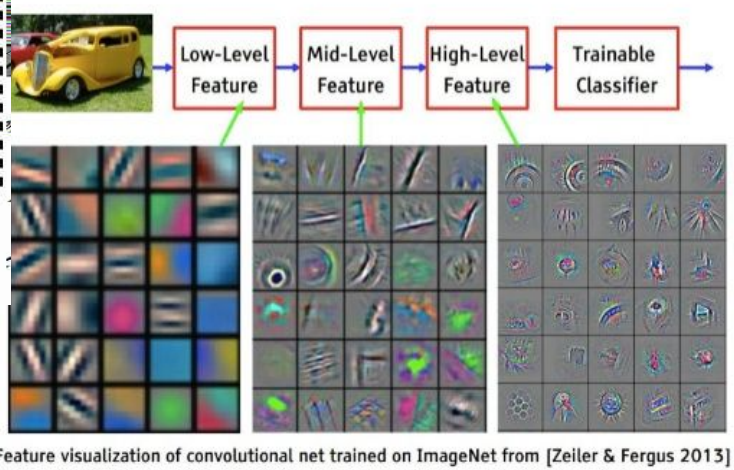
Shaidu et al, JPCC 122, 20800 (2018)

We expand the energy as a sum of “figures” (as in cluster expansion method), which is similar to dimer, trimer, and higher order filters of the convolution.

Convolutional neural networks

Between the black box learner & handcrafting (e.g. force field fitting)

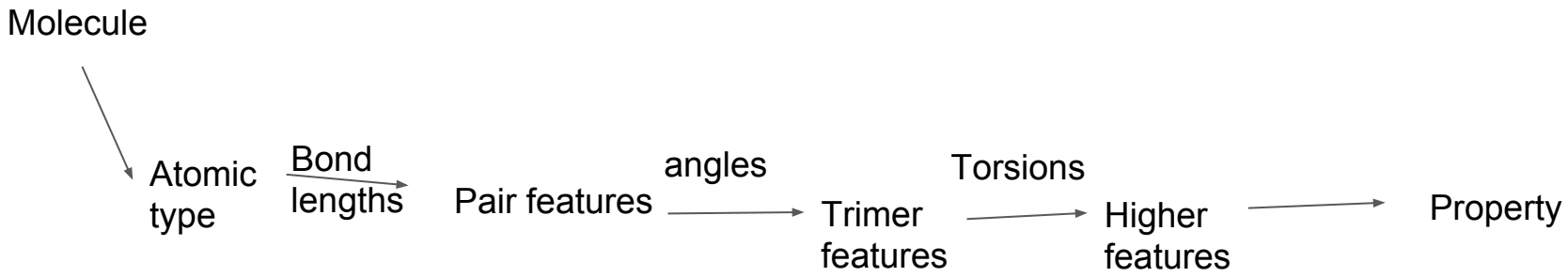




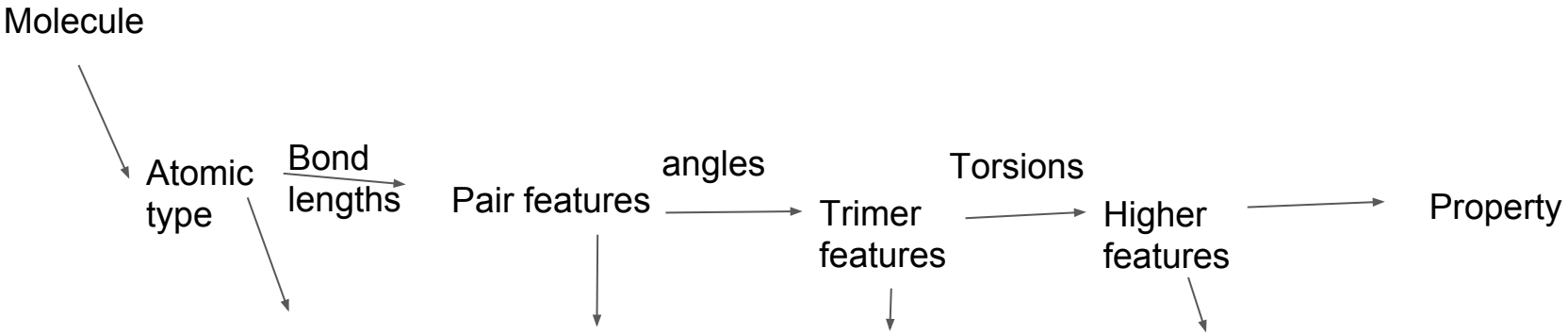
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Image: e-Lab, E. Culurciello lecture notes.

Convolutional neural networks for atomistic graphs



Convolutional neural networks for atomistic graphs



The number of features detected at each layer increase exponentially:

1- Nspec
2- Nspec * Nspec * Nbond
3- Nspec * (Nspec * Nbond) * (Nspec * Nbond) *Nangle

CNN -- in practice

Input: Atomic positions

Activations:

$t^1_{ia} = \delta_{a,\mathrm{type}_i}$

i. e. $t^1_i = [1,0,0,0] = \mathrm{H}$

+Positions

$r^1_i = [x_i, y_i, z_i]$

Layer 1: Bonds
(mask index: c)

$\tilde{r}_c$

Activations:

$t^2_{ijc} = R(w_c t^1_i t^1_j G(|r_{ij}| - \tilde{r}_c))$

+Positions

$r^2_i = [x_i, y_i, z_i]$

Layer 2: Valence-angle trimer
(mask index: d)

$\tilde{\theta}_d$

Activations:

$t^3_{ijkd} = R(w_d t^2_{ijc} t^2_{ikc'} G(\theta_{jik} - \tilde{\theta}_d))$

+Positions

$r^3_i = [x_i, y_i, z_i]$

R() is for ReLU, G() is gaussian.
All distances, angles, features can be equally weighted activators (wc & wd=1)
or separately weighted for a more expressive network (& more nonlinearity)

MolCNN

Same 10k data

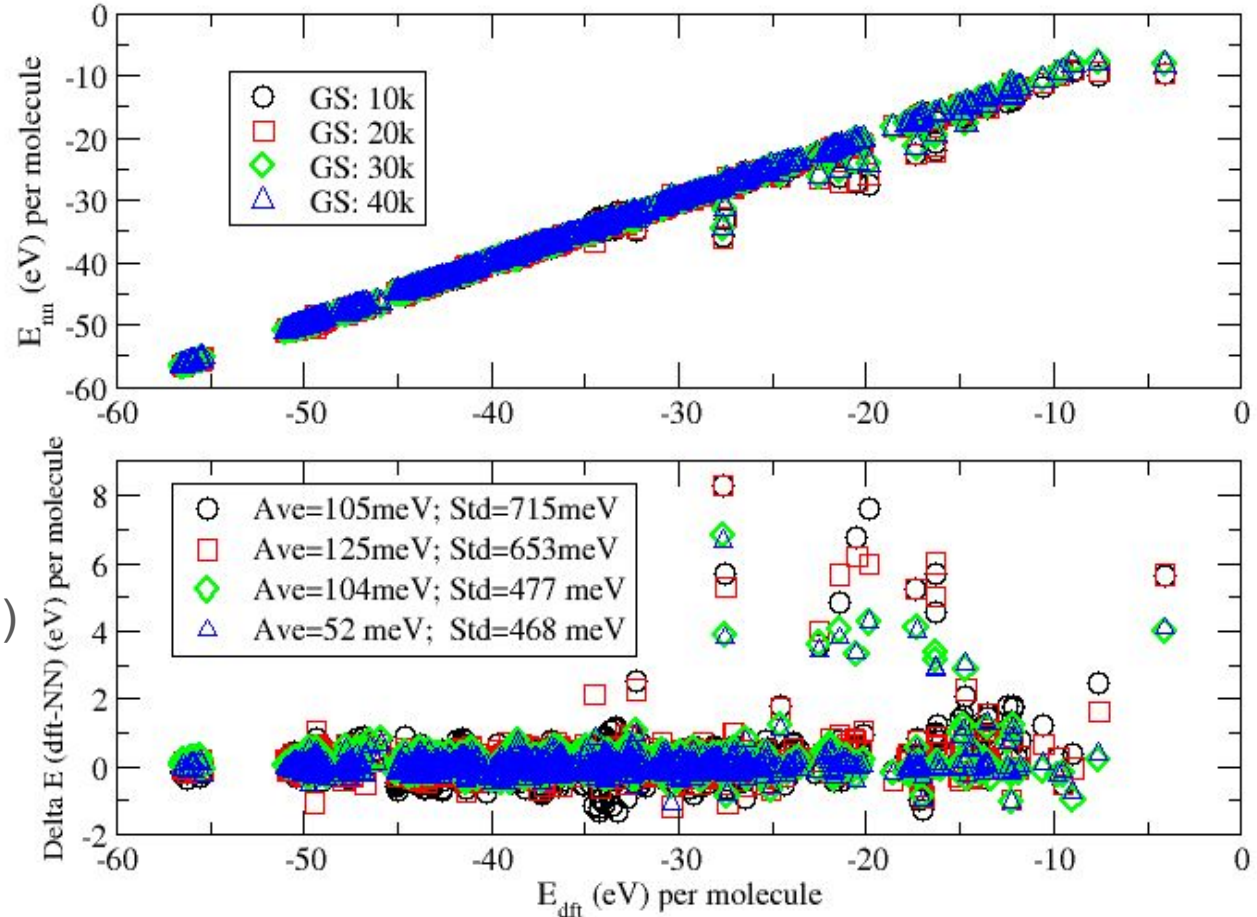
6 nearest neighbors

Nbond =4; Nangle =4

1-10k:only bond
10-20k:only angle
20-30k:both
30-40k:both but slower (1d-3)

Params
Pbond ~ 4*4*4 = 64
Pangle ~ 4*4*4*4 = 256

Batch size: 100; lr=1d-2; mask-eq-weighted network; validation



MolCNN -- Where to go?

Layer 1: Bonds
(mask index: c)

$\tilde{r}_c$

Activations:

$$t^2_{ijc} = R(w_c t^1_i t^1_j G(|r_{ij}| - \tilde{r}_c))$$

+Positions

$$r^2_i = [x_i, y_i, z_i]$$

Layer 2: Valence-angle trimer
(mask index: d)

$\tilde{\theta}_d$

Activations:

$$t^3_{ijkd} = R(w_d t^2_{ijc} t^2_{ikc'} G(\theta_{jik} - \tilde{\theta}_d))$$

+Positions

$$r^3_i = [x_i, y_i, z_i]$$

Test for network shapes/sizes

Trimers that share an edge (torsion angle layer).

Generalize trimer-trimer convolution beyond angles

Long-range correction layer

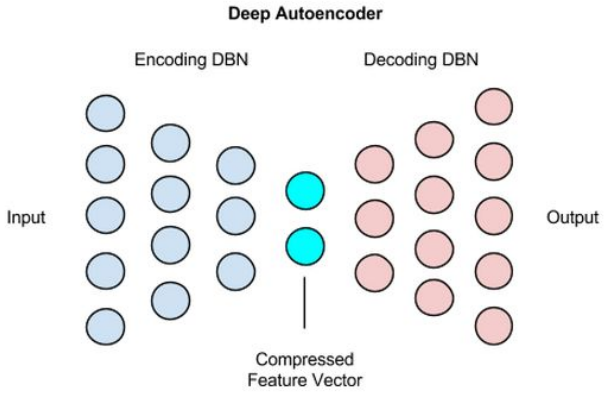
Target: Flexible force field replacement for CSP

Autoencoders:

Unsupervised data compression:
Coupled encoder+decoder able to reconstruct input passing from a much smaller representation.

May help finding compact representation, to be then used for learning, clustering or even de-noising.

Already a successful use case published - but with a Graph based representation (SMILES)
(Gomez-Bombarelli et al ASC Cent. Sci. 2018)



Reinforcement learning:

Machine learning to decide on “actions” to maximize “reward”.

Main idea: sample environment to estimate “value” of a configuration, based on future expected reward, then pick the best (Q-learning).
Value function can be learnt with a NN.

Proved effective in Atari, Go, Chess…
Useful to learn “intrinsic value” of a configuration
to guide a complex optimization procedure

Google DeepMind’s AlphaFold used to win the protein folding challenge.

Take home:

- Crystal structure prediction is a tough problem, NN can help
- Lowest hanging fruit for NN in CSP is exploration and geometry optimization
- GLYdata +A2A network :
  - Variability in data vs the representation
  - Overfitting vs Underfitting & the need to monitoring/control the dynamics
- ANIdata +A2A network:
  - Data variability > amount
  - Open source/Reproducibility
- ANIdata +MolCNN
  - WIP - Few parameters, easy to train, potentially interpretable
- Where next?

Credits:

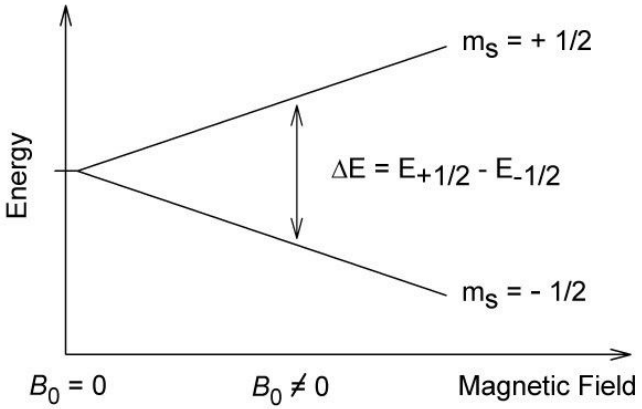
(SISSA) R Lot, F Pellegrini, Y Shaidu, S de Gironcoli

(MIT) S Wyant, A Kolpak

The End.

Notes

ab initio NMR



$$\Delta E = \hbar \gamma B_{\text{tot}} \Delta m_s = \hbar\omega$$

Induced field due to electron shielding

$$\mathbf{B}_{\text{in}}^{(1)}(\mathbf{r}) = -\vec{\sigma}(\mathbf{r})\mathbf{B}_{\text{ext}}$$

Density Functional Perturbation Theory in Linear Response:

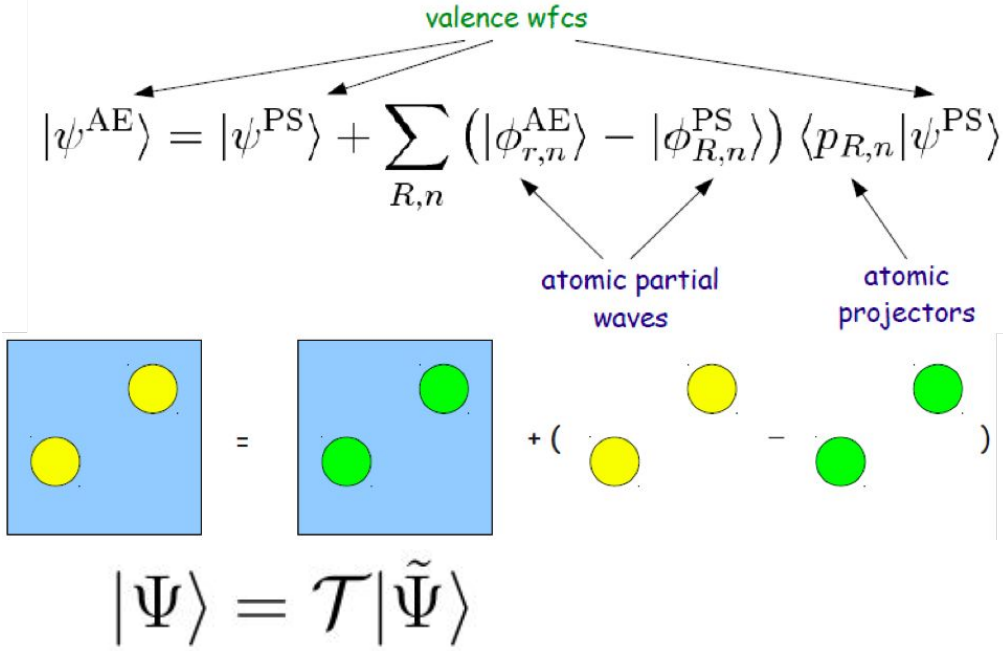
- Perturbation to Hamiltonian
- Perturbed Eigenstates
- Induced stationary current
- Induced field from induced current

Chemical shift

$$\delta = -(\sigma - \sigma_{\text{ref}})$$

Time \propto (Nat x (Nk x (Ni x (Nbx**NlogN**))))

ab initio NMR with Projector Augmented Wave



$$\mathbf{A}(\mathbf{r}) = 1/2\mathbf{B} \times \mathbf{r}$$

$$\Psi'(\mathbf{r}) = e^{(i/2c)\mathbf{r}\cdot\mathbf{t}\times\mathbf{B}}\Psi(\mathbf{r})$$

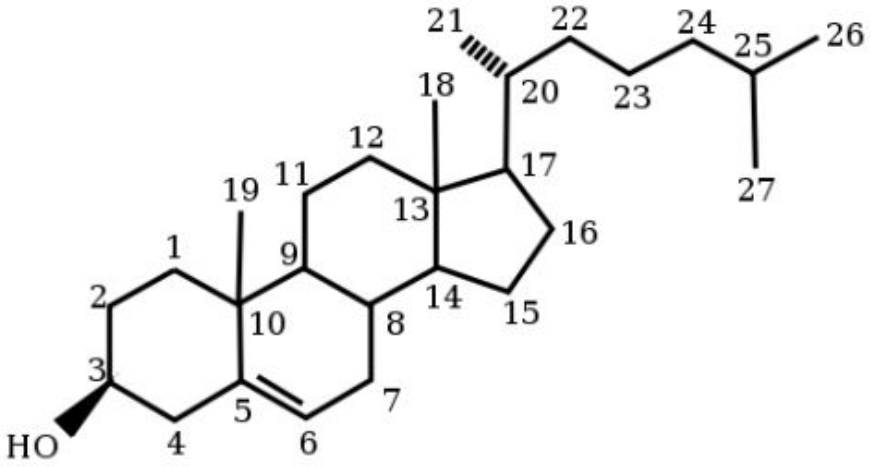
$$T_{\mathbf{B}} = 1 + \sum_{\mathbf{R},n} e^{(i/2c)\mathbf{r}\cdot\mathbf{R}\times\mathbf{B}}$$

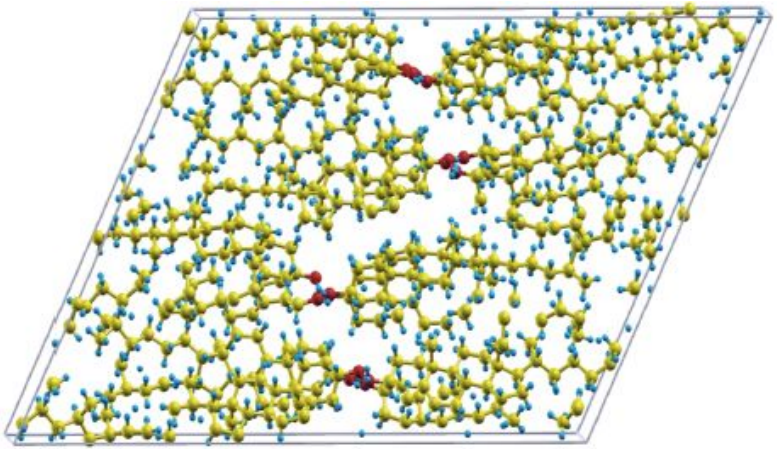
$$\left(\left|\phi^{AE}_{\mathbf{R},n}\right\rangle - \left|\phi^{PS}_{\mathbf{R},n}\right\rangle\right)$$

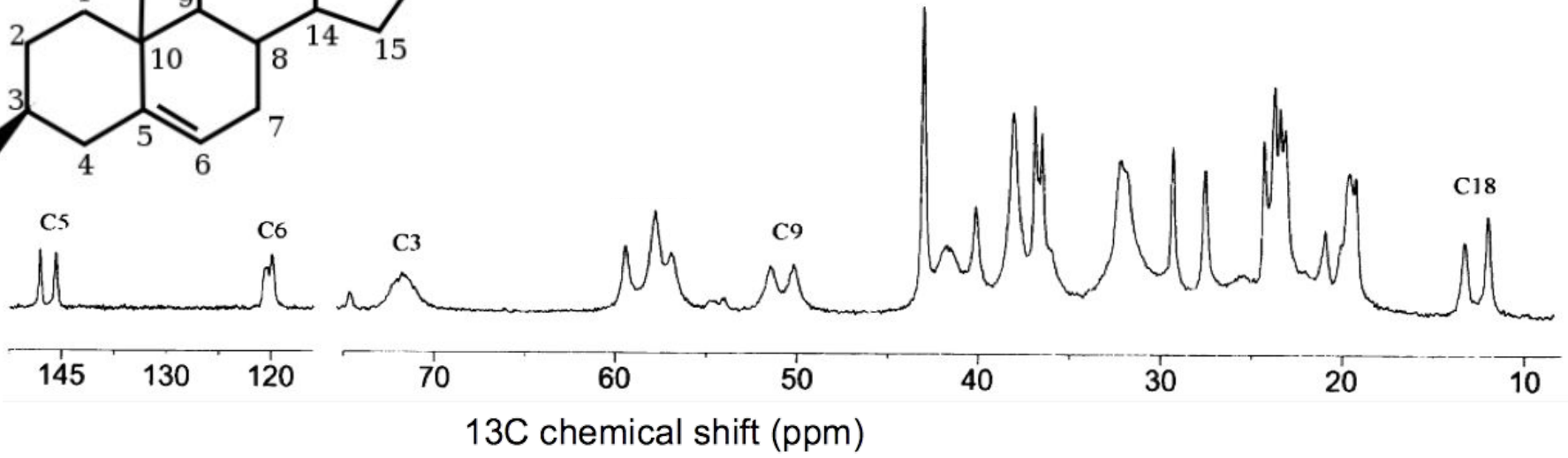
$$\left\langle p^{PS}_{\mathbf{R},n}\right| e^{-(i/2c)\mathbf{r}\cdot\mathbf{R}\times\mathbf{B}}$$

Blochl, PRB 50,17953 (1994)

13C NMR for CLR

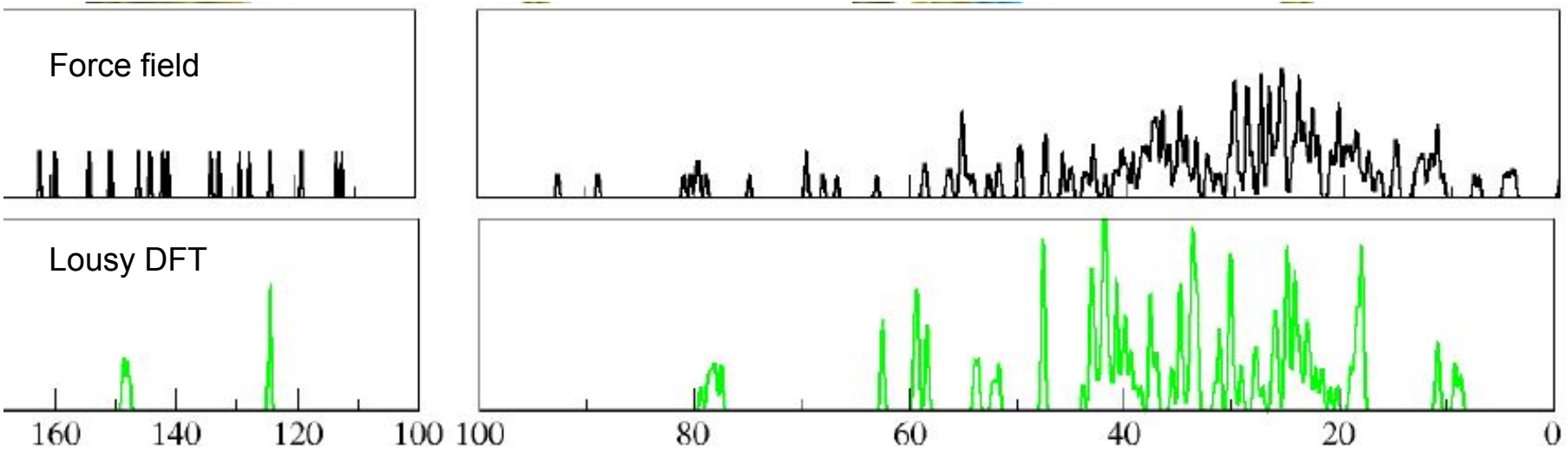






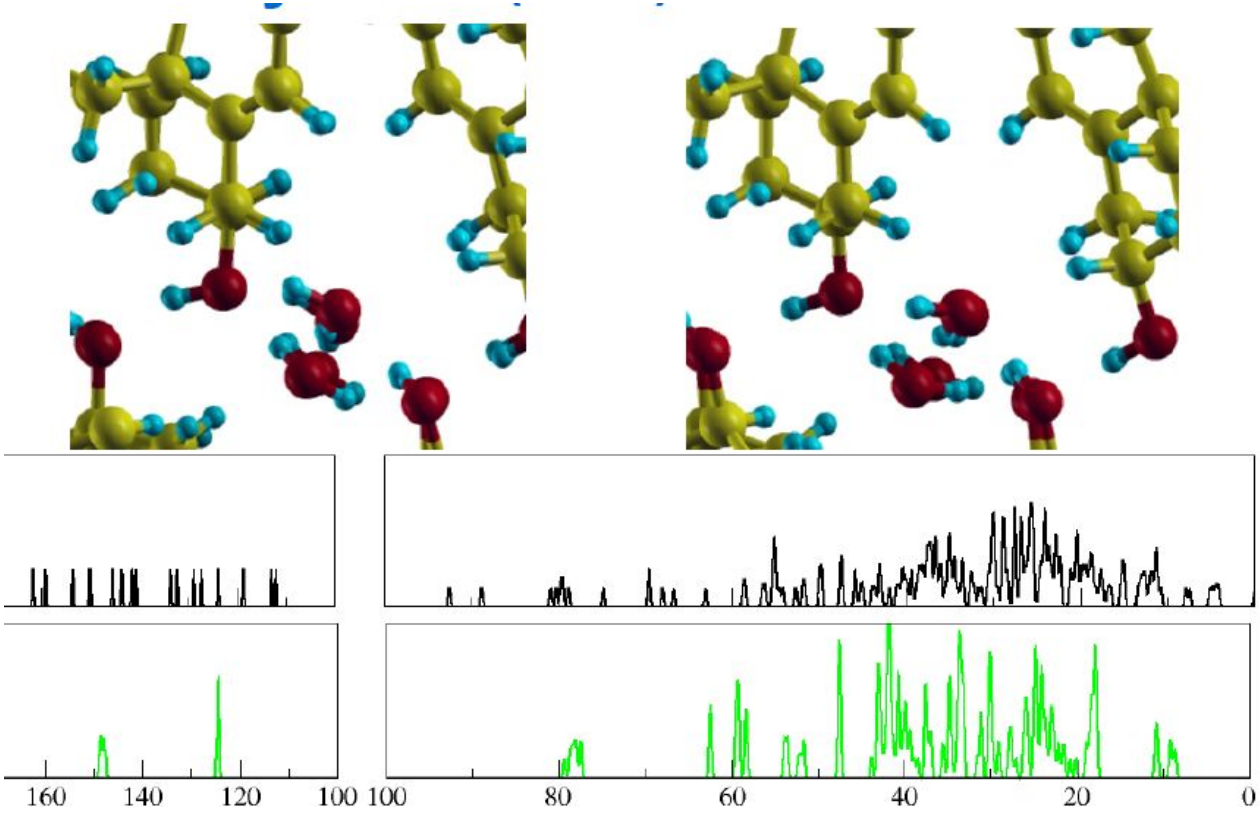
Guo et al, Biophys. J. 71: 2857-2868 (1996)

Geometry optimization



Cournia et al. JCompChem 26, 1383(2005)

Force fields for geometry optimization



Force field was not good enough to distinguish the spectra of different polymorphs,
or even to verify the same polymorph

Cournia et al. JCompChem 26, 1383(2005)

CNN -- in practice

Layer 1: Bonds
(mask index: c)

$\tilde{r}_c$

Activations:

$$t^2_{ijc} = R(w_c t^1_i t^1_j G(|r_{ij}| - \tilde{r}_c))$$

+Positions

$$r^2_i = [x_i, y_i, z_i]$$

+Bond directions

$$v^2_{ij} = [x_j - x_i, y_j - y_i, z_j - z_i]$$

Trimers are convolutions of two 2-atom nodes that share a corner.

An alternative here, would be to add a non-corner-sharing dimer-dimer mask. Ongoing.

Layer 2: Valence-angle trimer
(mask index: d)

$\tilde{\theta}_d$

Activations:

$$t^3_{ijkd} = R(w_d t^2_{ijc} t^2_{ikc'} G(\theta_{jik} - \tilde{\theta}_d))$$

+Positions

$$r^3_i = [x_i, y_i, z_i]$$

+Trimer Orientations

$$q^3_{ijk} = [q^y_{ijk}, q^x_{ijk}, q^z_{ijk}, q^w_{ijk}]$$

Number of possible feature masks
(and weights to get their contribution to energy) increase dramatically:
1- Nspec
2- Nspec^2 * Nbond
3- Nspec^3 * Nangle

We can keep only some trimers that get activated the most (max-pooling ~ coarse graining) $t^{3'}_{id} = \max_{jk} t^3_{ijkd}$

Fingerprint references

- Ref1: K. Hansen, F. Biegler, O. A. von Lilienfeld, K.-R. M€uller, A. Tkatchenko,Nat. Commun. https://pubs.acs.org/doi/pdfplus/10.1021/acs.jpclett.5b00831 Ref2: https://pubs.acs.org/doi/pdfplus/10.1021/ci020345w Ref3: https://arxiv.org/pdf/1509.09292.pdf ref4: https://journals.aps.org/prl/pdf/10.1103/PhysRevLett.107.085504 ref5:https://aip.scitation.org/doi/10.1063/1.4818005

Brainstorming for NNs in CSP

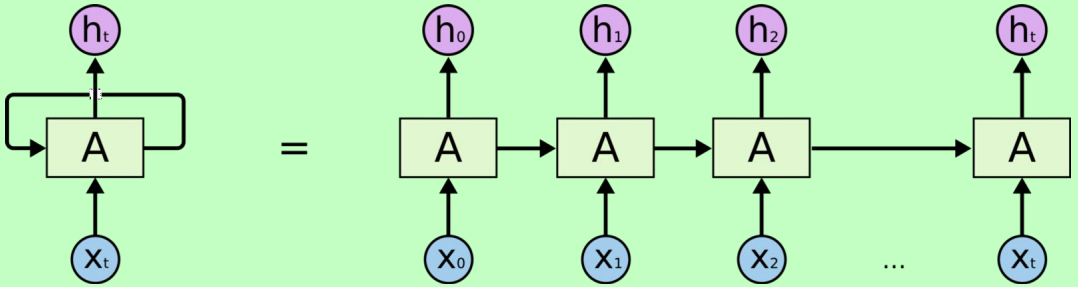
- Neural networks and where CSP can help:
  - Accelerating the exploration of polymorphic phase space -
    - the supposedly low hanging fruit is the geometry relaxation. This means having a very flexible and well performing force field/potential for the molecule at hand. Next step would of course be a force field transferable across different molecules.
    - Relaxation as a winning move: As number of atoms, species, network parameters increase, calculating forces as derivatives might become the bottleneck. Reinforcement learning methods, where the value of each atomistic move is learnt, can be viable alternatives to making quick optimization steps.
    - Evolutionary algorithm accelerated: A jump here would be accelerating the phase space search by exploring more intelligently than evolutionary algorithm using reinforcement learning where the reward this time can be finding a low energy or increasing population diversity or matching the experimental characterization.
  - Accelerating characterization:
    - Characterization could simply be total energy, but also possibly enthalpy, NMR, EPR, elastic constants etc. Challenges: Efficiency & accuracy (representation, ground truth

Recurrent networks:

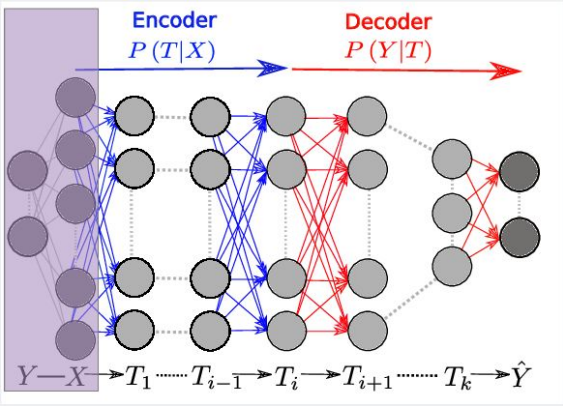
Neural networks useful to analyze “time series”.

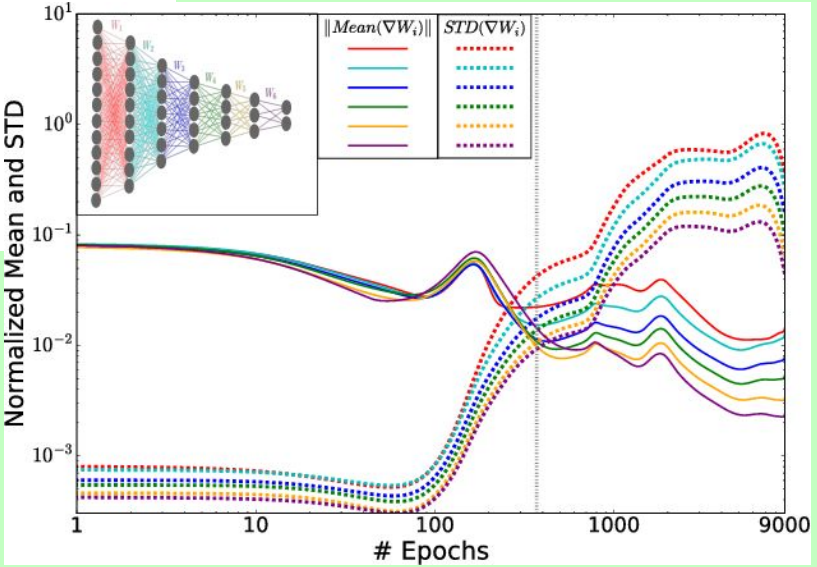
Main idea: add to the input at time t some information from the output at time t-1 ➜ acts like a “memory” for the network.

Typically one takes N timesteps and optimizes all output with SGD or similar. This can lead to “gradient dilution” for long time correlation ➜ introduce “gated units” to preserve memories for longer times (see GRU, LSTM).



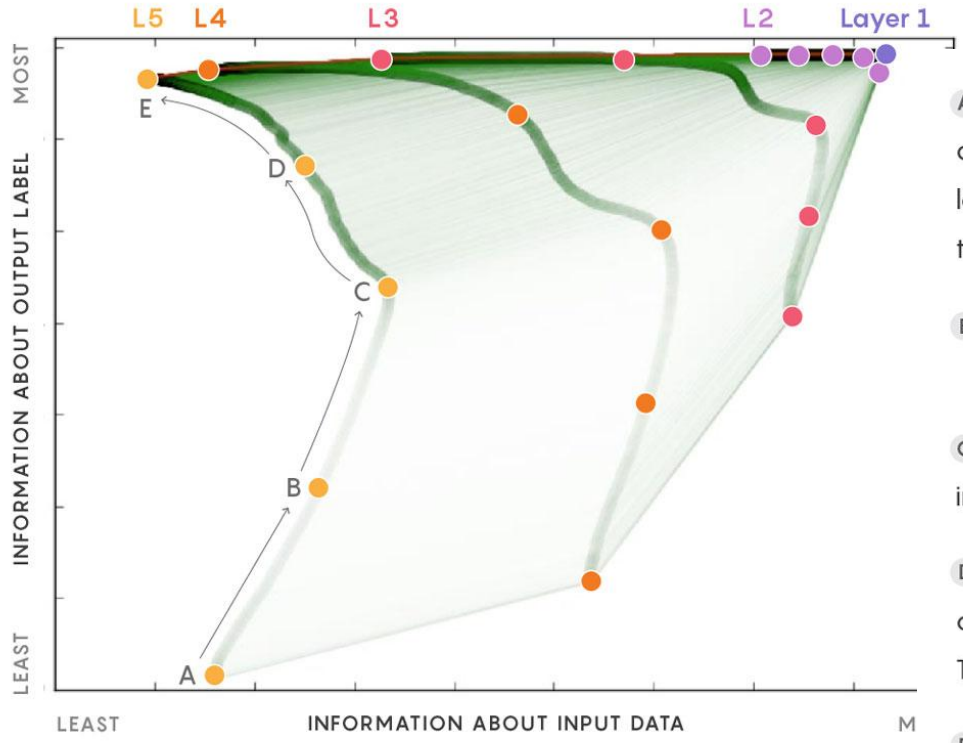
- At each layer information is processed to remove the inessential (data processing inequality)
- In the information plane we can follow the training as deeper layer gain mutual info with output
- There seem to be 2 phases: fitting and compression
- SGD → “compression by diffusion”
- More layers seem to be useful to break diffusion in simpler steps





Inside Deep Learning

New experiments reveal how deep neural networks evolve as they learn.



A INITIAL STATE: Neurons in Layer 1 encode everything about the input data, including all information about its label. Neurons in the highest layers are in a nearly random state bearing little to no relationship to the data or its label.

B FITTING PHASE: As deep learning begins, neurons in higher layers gain information about the input and get better at fitting labels to it.

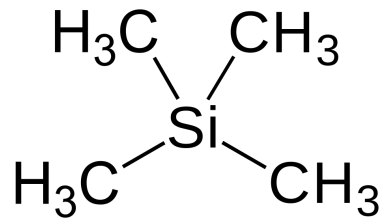
C PHASE CHANGE: The layers suddenly shift gears and start to "forget" information about the input.

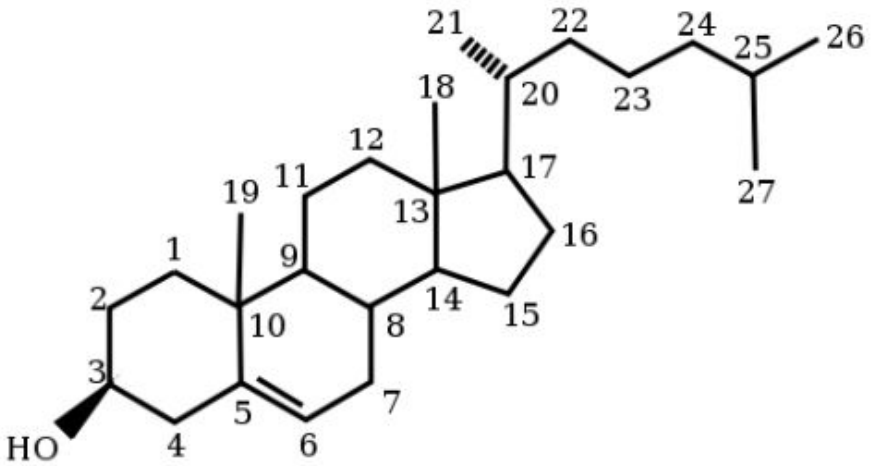
D COMPRESSION PHASE: Higher layers compress their representation of the input data, keeping what is most relevant to the output label. They get better at predicting the label.

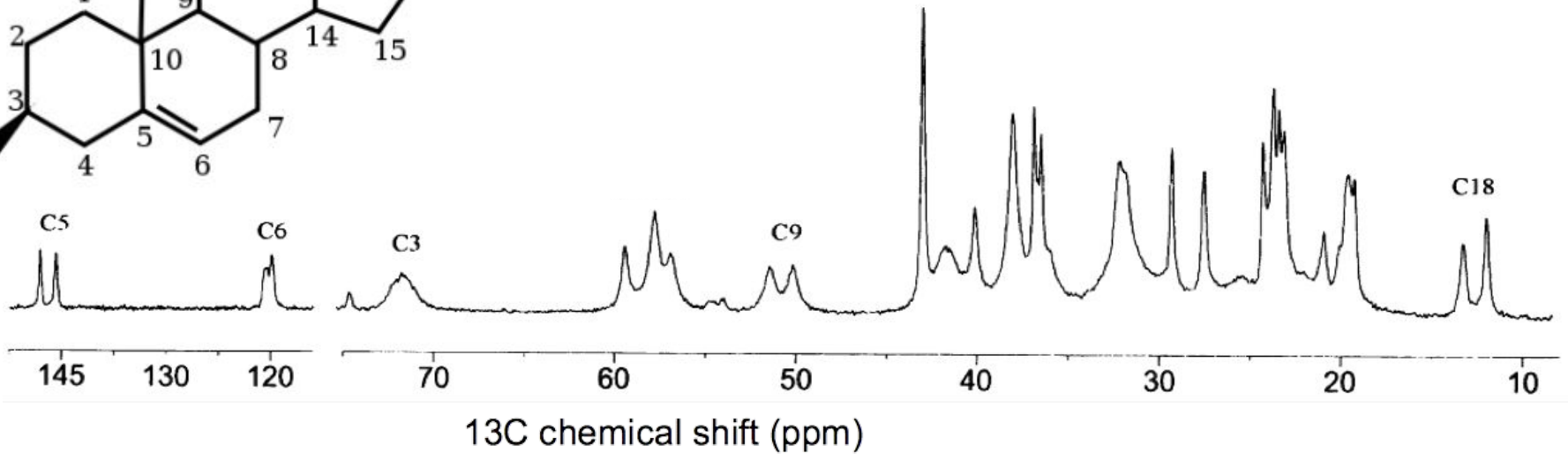
E FINAL STATE: The last layer achieves an optimal balance of accuracy and compression, retaining only what is needed to predict the label.

13C NMR for CLR

Shift reference:

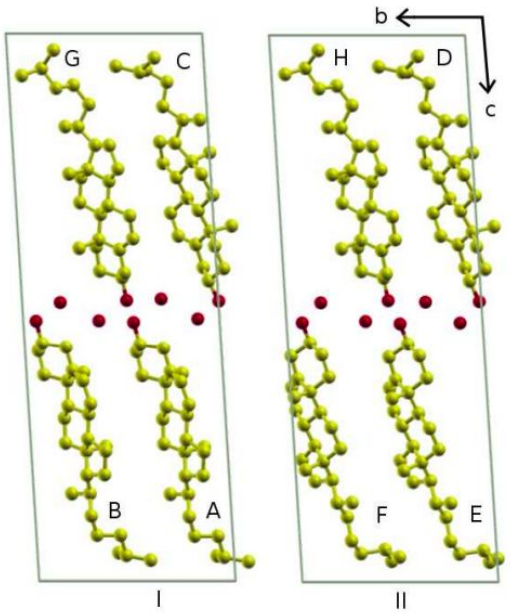




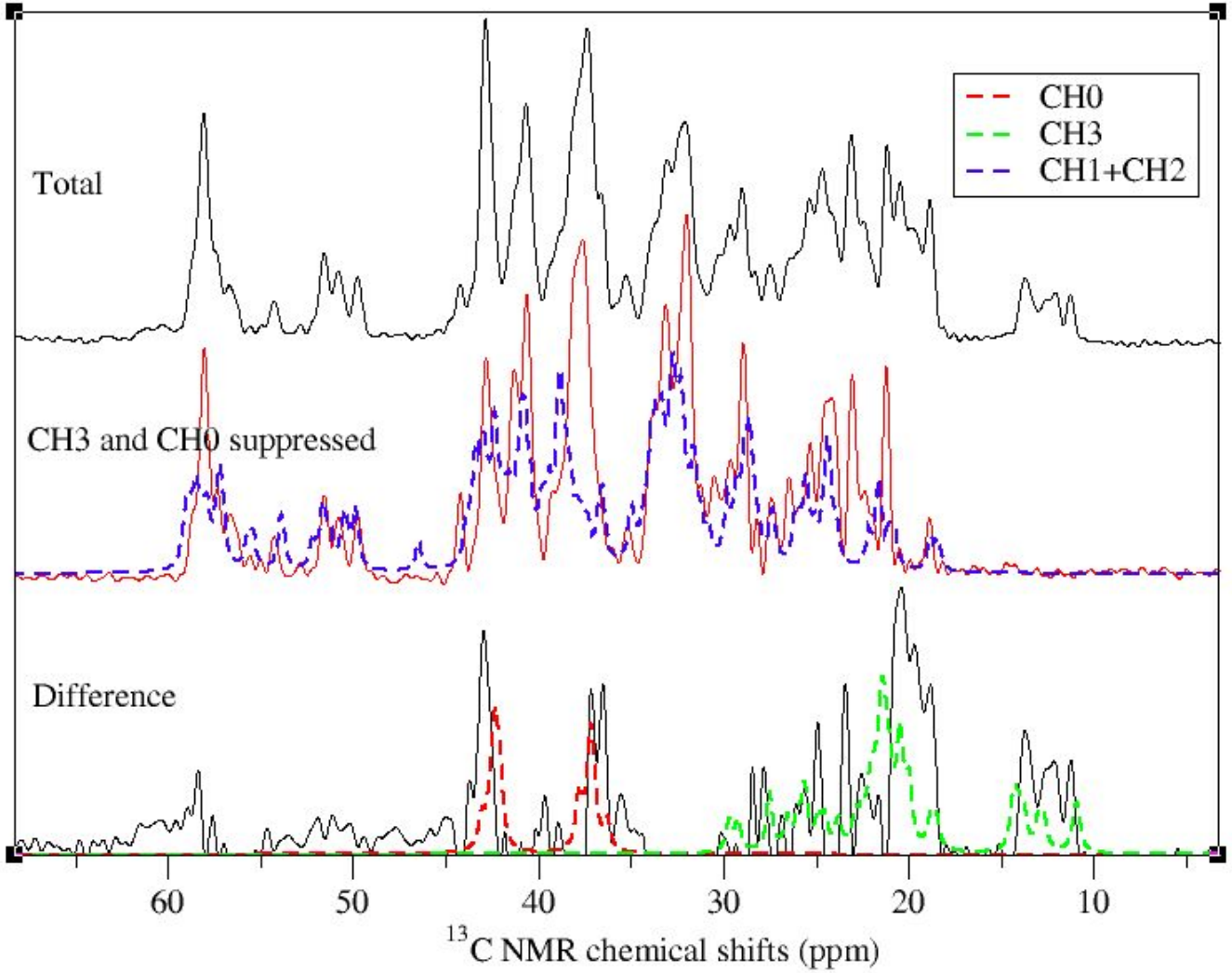


Guo et al, Biophys. J. 71: 2857-2868 (1996)

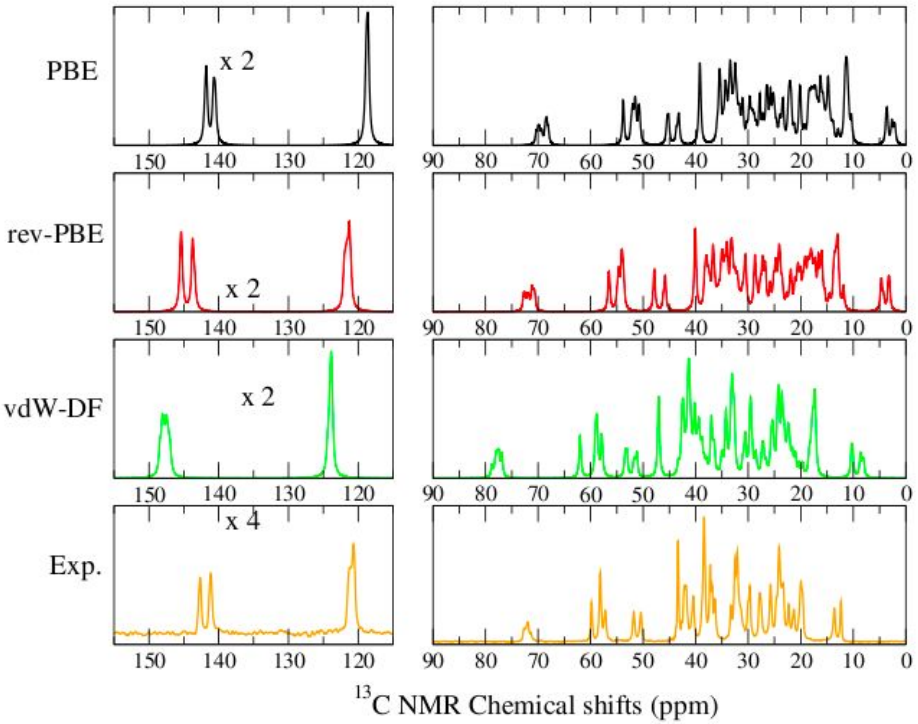
Spectral editing



ChM



Different functionals before secondary referencing

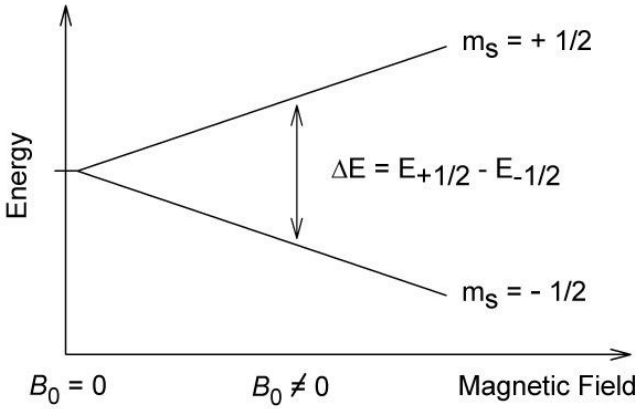


For ab initio non-local functionals (i.e. van der Waals aware functionals) one should be careful in not comparing apples to oranges:

In vdW functionals, we have F[n(r),n(r')] terms. Hence this means they also include local correlations as well as non-local. Due to this source of 'double counting', when people fit the parameters of a vdW functional, they don't do it for any local XC functional but they choose one in particular. For example vdW-DF is parametrized assuming revPBE xc, while vdW-DF2 is built on rpw86.

People often forget as they compare vdW treatments, that the other parts of the functional change also, and it might be the significant part. So, compare vdw-DF to revpbe first, if you want to see the effect of exchange.

ab initio NMR



$$\Delta E = \hbar \gamma B_{\text{tot}} \Delta m_s = \hbar\omega$$

Chemical shielding $H_{\text{cs}} = -\hbar\gamma\, \mathbf{I} \cdot \boldsymbol{\sigma} \cdot \mathbf{B}_0$

New addition to hamiltonian (p): $\mathbf{A}(\mathbf{r}) = 1/2\mathbf{B} \times \mathbf{r}$

$$\mathbf{j}^{(1)}(\mathbf{r}') = 2\sum_o [\langle \bar{\psi}_n^{(1)} | \, \bar{\mathbf{J}}^{(0)}(\mathbf{r}') \, | \bar{\psi}_n^{(0)} \rangle + \langle \bar{\psi}_n^{(0)} | \, \bar{\mathbf{J}}^{(0)}(\mathbf{r}') \, | \bar{\psi}_n^{(1)} \rangle + \langle \bar{\psi}_n^{(0)} | \, \bar{\mathbf{J}}^{(1)}(\mathbf{r}') \, | \bar{\psi}_n^{(0)} \rangle]$$

$$\mathbf{B}_{\text{in}}^{(1)}(\mathbf{r}) = -\vec{\sigma}(\mathbf{r})\mathbf{B}_{\text{ext}} = \frac{1}{c}\int d^3\mathbf{r}' \mathbf{j}^{(1)}(\mathbf{r}') \times \frac{\mathbf{r} - \mathbf{r}'}{|\mathbf{r} - \mathbf{r}'|^3}$$

$\delta = -(\sigma - \sigma_{\text{ref}})$ Chemical shift

Some terms in GIPAW:

$$\Delta \mathbf{J}^p_{\mathbf{R}}(\mathbf{r}') = \sum_{n,m} |\tilde{p}_{\mathbf{R},n}\rangle \left[\langle \phi_{\mathbf{R},n} | \mathbf{J}^p(\mathbf{r}') | \phi_{\mathbf{R},m} \rangle - \langle \tilde{\phi}_{\mathbf{R},n} | \mathbf{J}^p(\mathbf{r}') | \tilde{\phi}_{\mathbf{R},m} \rangle \right] \langle \tilde{p}_{\mathbf{R},m} |$$

$$\begin{aligned} \mathbf{j}^{(1)}_{\Delta p}(\mathbf{r}') = & \sum_{\mathbf{R}',o} Re[\langle \bar{\psi}^{(0)}_o | \Delta \mathbf{J}^p_{\mathbf{R}'}(\mathbf{r}') G(\varepsilon^{(0)}_o)(\bar{H}^{(1)} - \varepsilon^{(0)}_o \bar{S}^{(1)}) | \bar{\psi}^{(0)}_o \rangle \\ & - \langle \bar{\psi}^{(0)}_o | \Delta \mathbf{J}^p_{\mathbf{R}'}(\mathbf{r}') G(\varepsilon^{(0)}_o) \sum_{\mathbf{R}} \frac{\mathbf{B} \times \mathbf{R}'}{2c} \cdot \mathbf{v}(\varepsilon^{(0)}_o) | \bar{\psi}^{(0)}_o \rangle] \\ & - \sum_{\mathbf{R},\mathbf{R}',o,o'} \langle \bar{\psi}^{(0)}_o | \Delta \mathbf{J}^p_{\mathbf{R}'}(\mathbf{r}') | \bar{\psi}^{(0)}_{o'} \rangle \langle \bar{\psi}^{(0)}_{o'} | (\mathbf{R} - \mathbf{R}') \times \frac{1}{2ic} [\mathbf{r}, Q_{\mathbf{R}}] \cdot \mathbf{B} | \bar{\psi}^{(0)}_o \rangle \end{aligned}$$

How does QE scale?

Time spent in scf calculation: Tscf

Niter*Titer + Tinit ; Titer=Nk*Tdiag + Trho + Thxcv ; Tdiag=NhpsiThpsi + Torth + Tsub

Thspi=aNbandNpw + bNbandN123log(N123) + cNbandNpwProjall [N123~8Npw]

Torth=dNpw(Nbandtrial)^2~dNpw(2Nbnd)^2 ; Tsub = e(2Nbndtrial)^3

Trho=fNbdn(8Npw*)log(8Npw*) + gNbnd(8Npw*) + Tpaw

Thxcv=j(8Npw*) + k8Npw*log(8Npw*)

For NMR via linear response ~Natom (as an approximation to Nproj) x (7directions x Nk) x (Nband x N123logN123)