Molecular Property Prediction

MolProPred

Apr. 2023

Outline

➢ Message-Passing GNN

- Initialization
- Message-Passing functions
- Readout

➢ Molecular GNN

- 1D String
- 2D Graphs
- 3D Euclidean Space

➢ SchNet Algorithm

- Molecular representation
- Interaction blocks
- Filter-generating networks

➢ SchNet for Scalar Coupling

- Main result & issues
- Tests
- Score & Rank

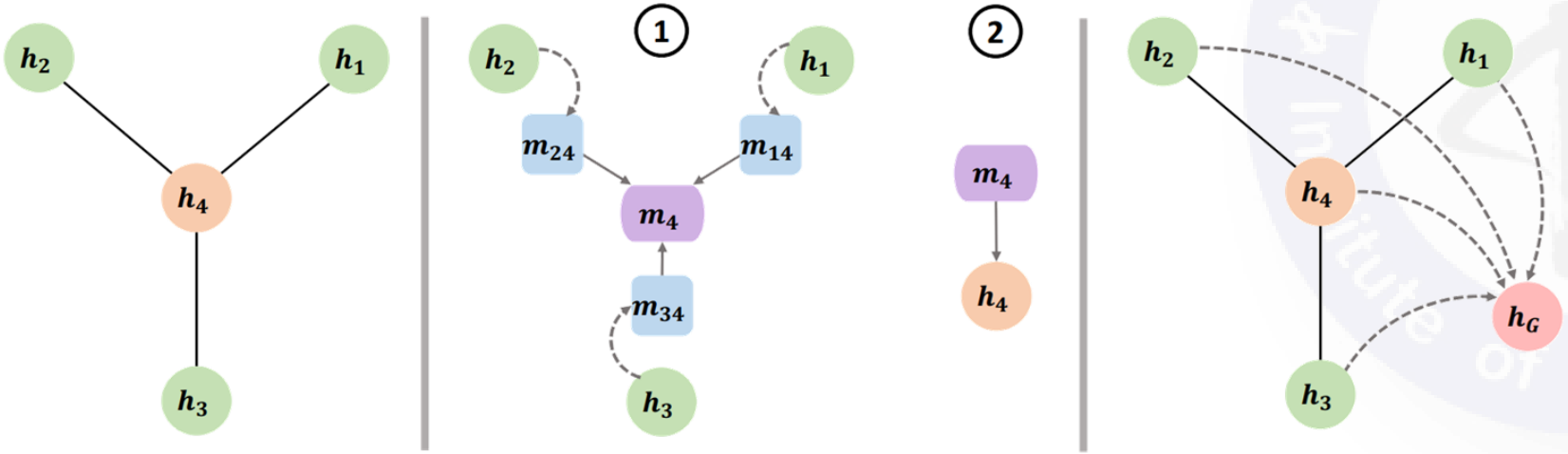
Message-Passing GNN

(a) Initialization

$G=(V, E)$, i-th node $u_i \in V$, edge $e_{i,j} \in E$.

u_i initialized as $\mathbf{h}_i^{(0)} = \mathrm{Emb}_n(u_i)$

$e_{i,j}$ initialized as $\mathbf{a}_{ij} = \mathrm{Emb}_e(e_{ij})$



(a) An example graph (b) Message-passing function (c) Readout function

arXiv:2209.05582

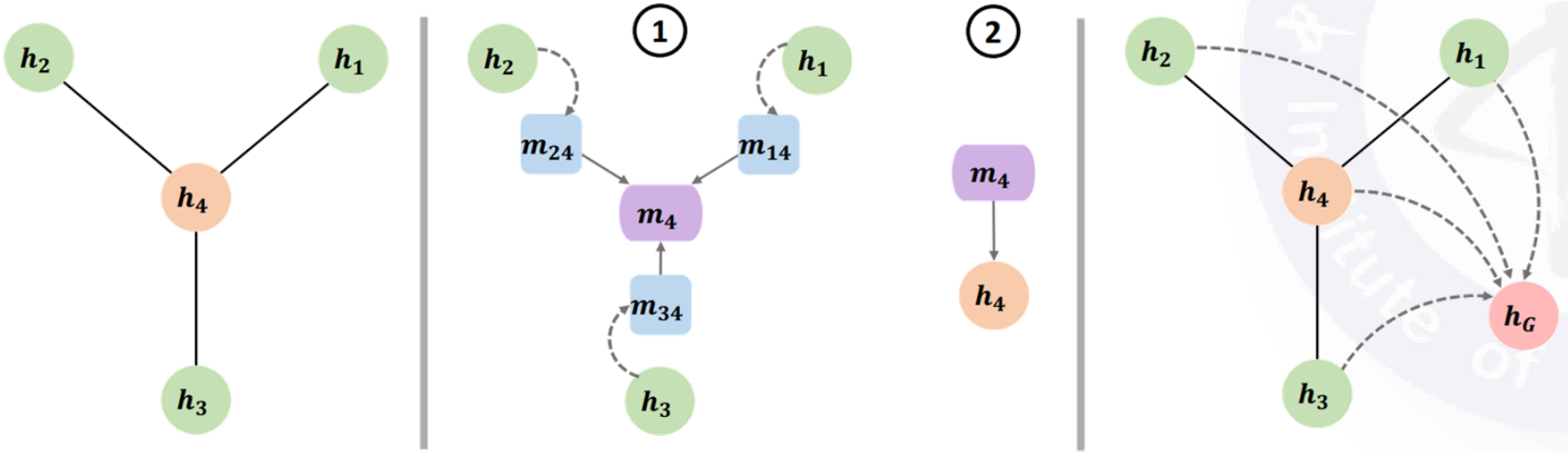
Message-Passing GNN

(b) Message-passing function

① Aggregate: $$\mathbf{m}_i^{(k)} = \sum_{u_j \in \mathrm{N}(u_i)} \phi_m^{(k)} \left(\mathbf{h}_i^{(k-1)}, \mathbf{h}_j^{(k-1)}, \mathbf{a}_{ij}\right)$$

② Update (transformer): $$\mathbf{h}_i^{(k)} = \phi_u^{(k)} = \left(\mathbf{h}_i^{(k-1)}, \mathbf{m}_i^{(k)}\right)$$

$\phi_m^{(k)}(\cdot), \phi_u^{(k)}(\cdot)$: message/update functions.



(a) An example graph (b) Message-passing function (c) Readout function

arXiv:2209.05582

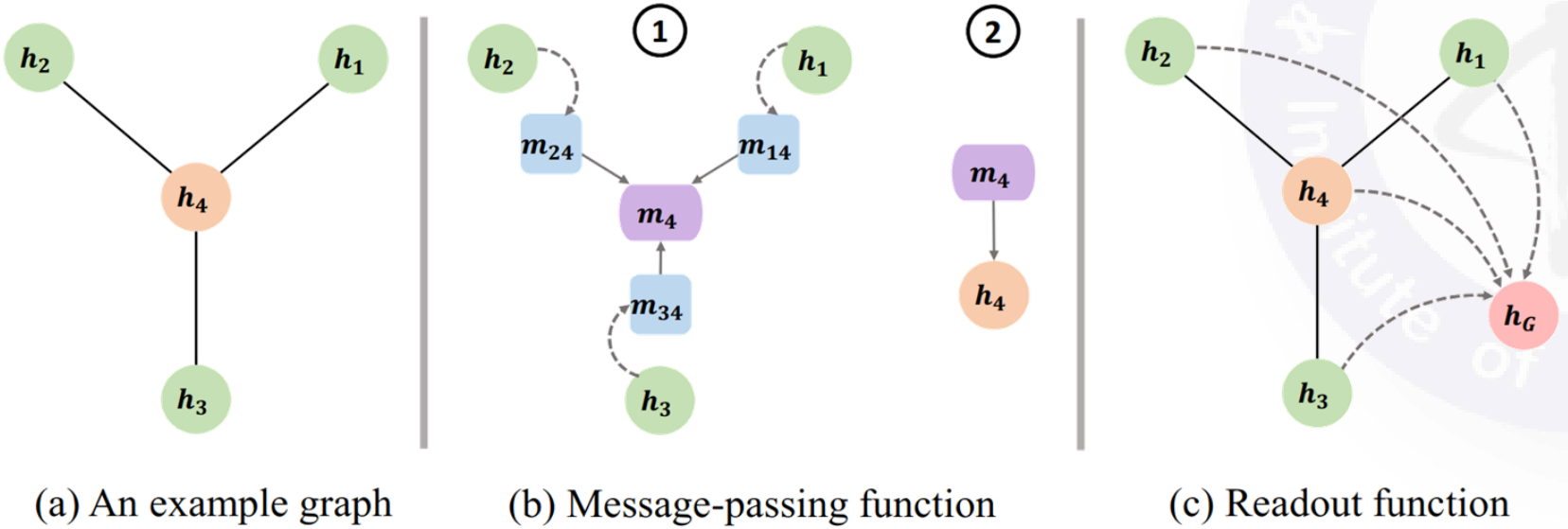
Message-Passing GNN

(c) Readout operations

After the last message-passing layer:

$$\mathbf{h}_G = R\left(\{\mathbf{h}_i^{(k)} \mid u_i \in \mathrm{V}\}\right)$$

$R(\cdot)$: readout function.



(a) An example graph (b) Message-passing function (c) Readout function

arXiv:2209.05582

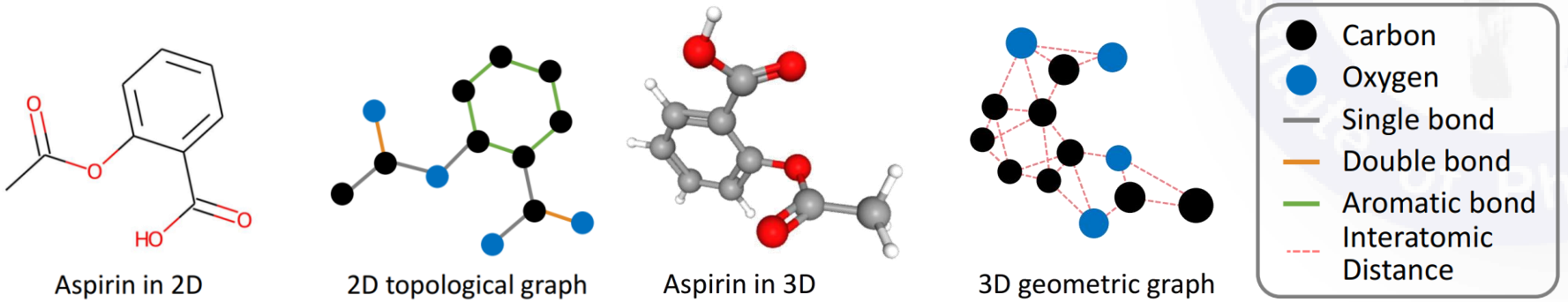
Molecular GNN

1D string:

- SMILES, SMARTS, SELFIES, ECFP, MACCS, . . .
- Struggle to directly model topological and geometric information.

2D graph: only topological info.

- Duvenaud et al., Hu et al., MPNN, . . .
- Ignore distance and angle information.



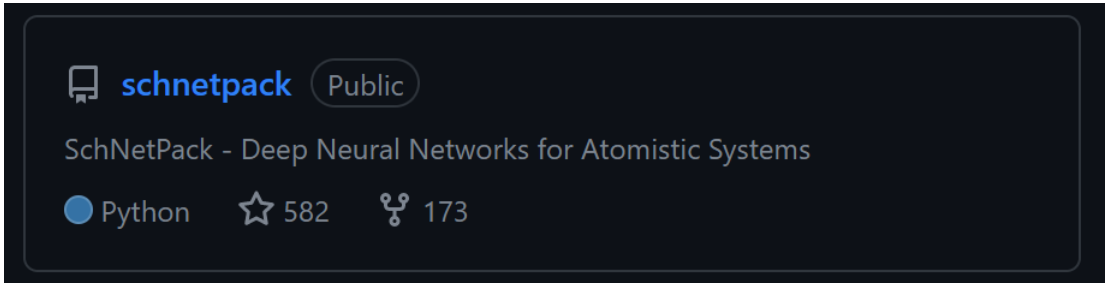
Aspirin in 2D | 2D topological graph | Aspirin in 3D | 3D geometric graph

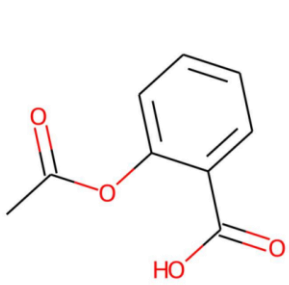
arXiv:2209.05582

Molecular GNN

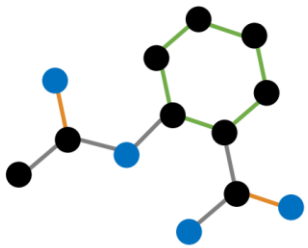
GNNs for molecules in 3D Euclidean space:

- Deep tensor neural network (DTNN): model distance in message-passing.
- **SchNet**: well designed layers to model local correlations, effectively encodes 3D distance to GNN, and inspires many follow-up works.

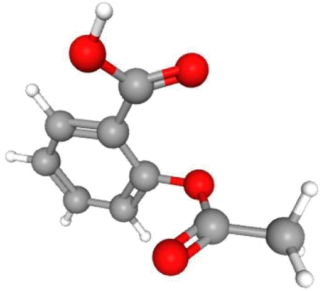




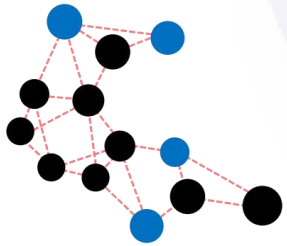
Aspirin in 2D



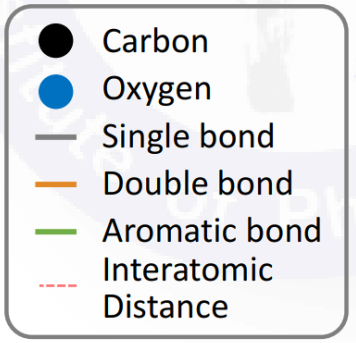
2D topological graph



Aspirin in 3D



3D geometric graph



arXiv:2209.05582

SchNet - Introduction

SchNet: model atomistic systems by making use of continuous-filter convolutional layers(model interaction between atoms).

Allow to model complex atomic interactions.

Predict potential energy surfaces.

Speed up the exploration of chemical space.

Consider fundamental symmetries of atomistic systems.

rotational and translational invariance as well as invariance to atom indexing

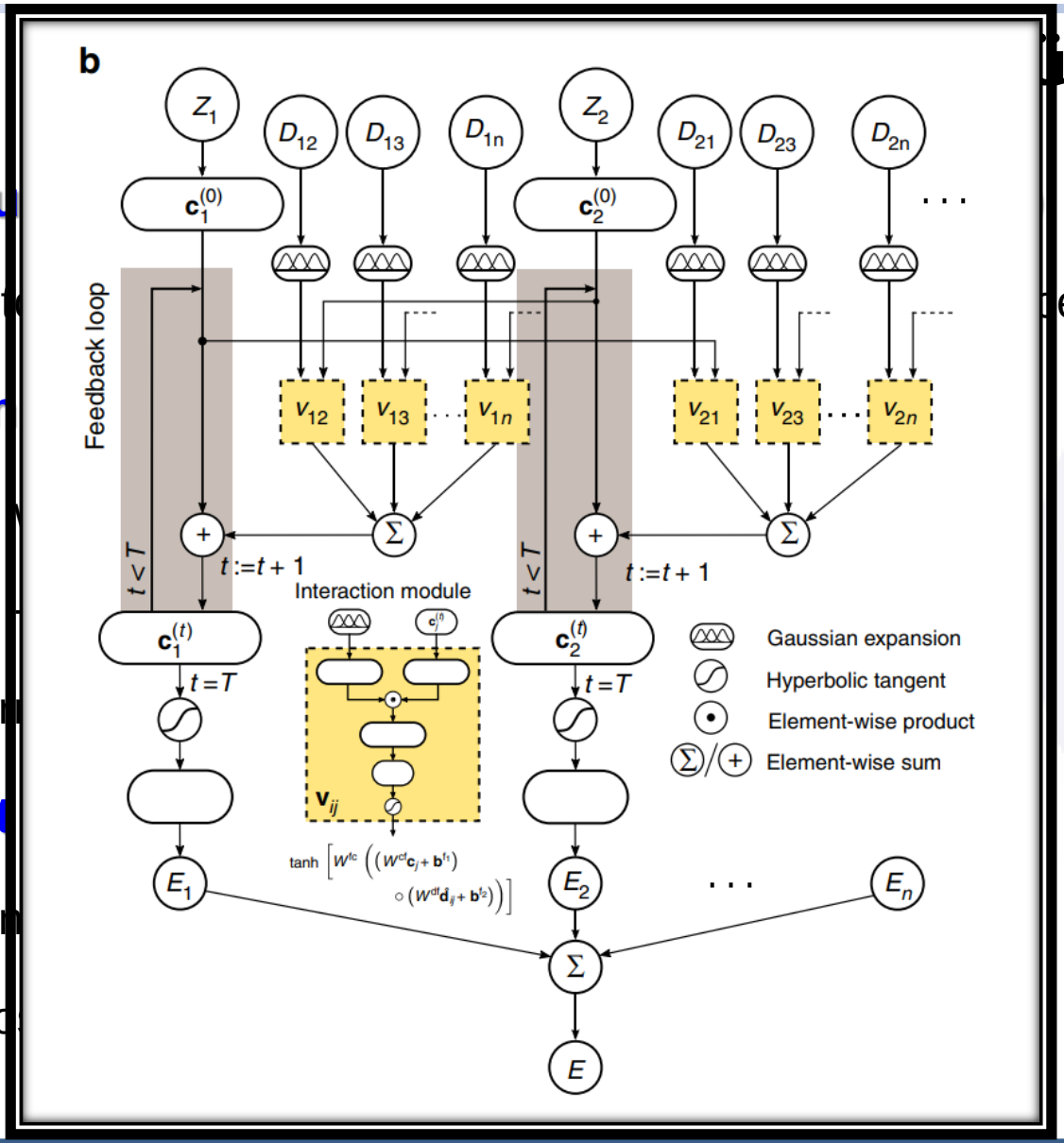
LEARNING HOW TO EXPLAIN

- **Quantu**
  - Vect
- **Patterr**
  - The
    -
  - Lear
- **Schnet**
  - learn

  acro

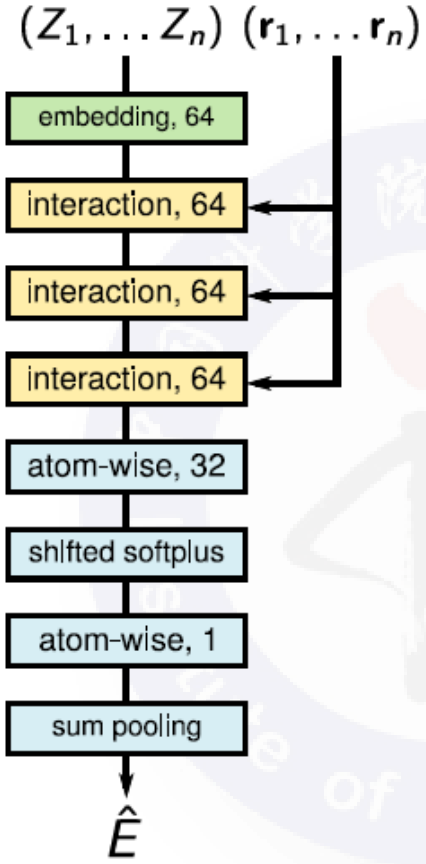
itt

es D.



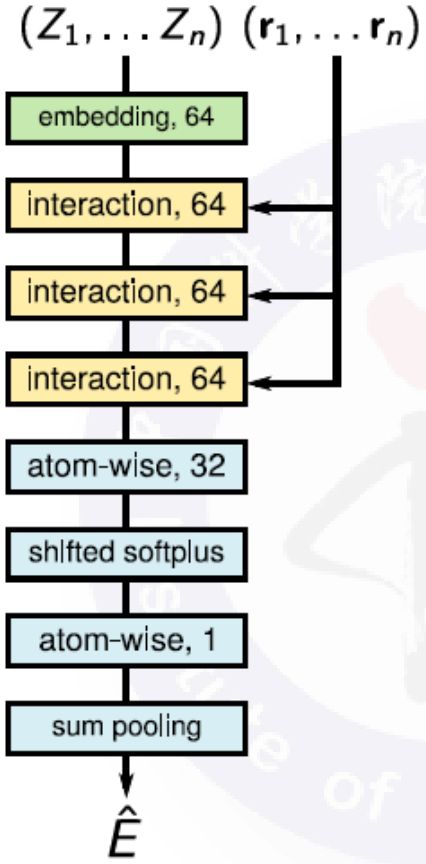
Method

- SchNet is a variant of the earlier proposed Deep Tensor Neural Networks(DTNN).
  - DTNN: interactions are modeled by tensor layers, i.e., atom representations and interatomic distances are combined using a parameter tensor.
  - SchNet: makes use of continuous-filter convolutions with filter-generating networks to model the interaction term.



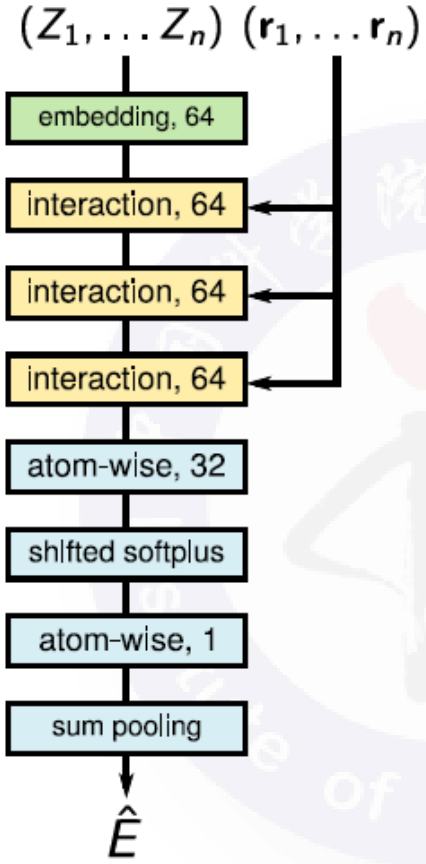
Method

- At each layer, the molecule is represented atom-wise analogous to pixels in an image.
- Interactions between atoms are modeled by the three interaction blocks.
- The final prediction is obtained after atom-wise updates of the feature representation and pooling of the resulting atom-wise energy.



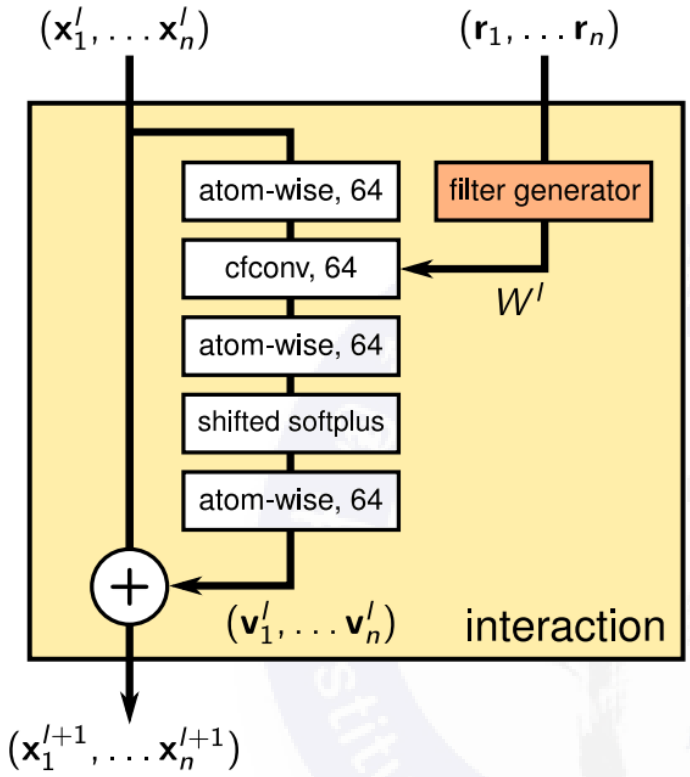
Molecular representation

- Nuclear charges $Z=(Z_1,\ldots,Z_n)$
- Positions $R=(\mathbf{r}_1,\ldots,\mathbf{r}_n)$
- The atoms are described by a tuple of features $X^l=(\mathbf{x}_1^l,\ldots,\mathbf{x}_n^l)$
  - $\mathbf{x}_i^l \in \mathbb{R}^F$ F: number of feature maps
  - n: number of atoms
  - l: current layer
- $\mathbf{x}_i^0$ is initialized using an embedding dependent on the atom type Z_i. $\mathbf{x}_i^0 = \mathbf{a}_{Z_i}$
- The atom type embeddings $\mathbf{a}_Z$ are initialized randomly and optimized during training.



Interaction blocks

- Updating the atomic representations based on pair-wise interactions with the surrounding atoms.
- Continuous-filter convolutional layers, a generalization of the discrete convolutional layers commonly used.
  - Atoms are located at arbitrary positions.
- Model the filters continuously with a filter-generation neural network W^l that maps the atom positions to the corresponding values of the filter bank.



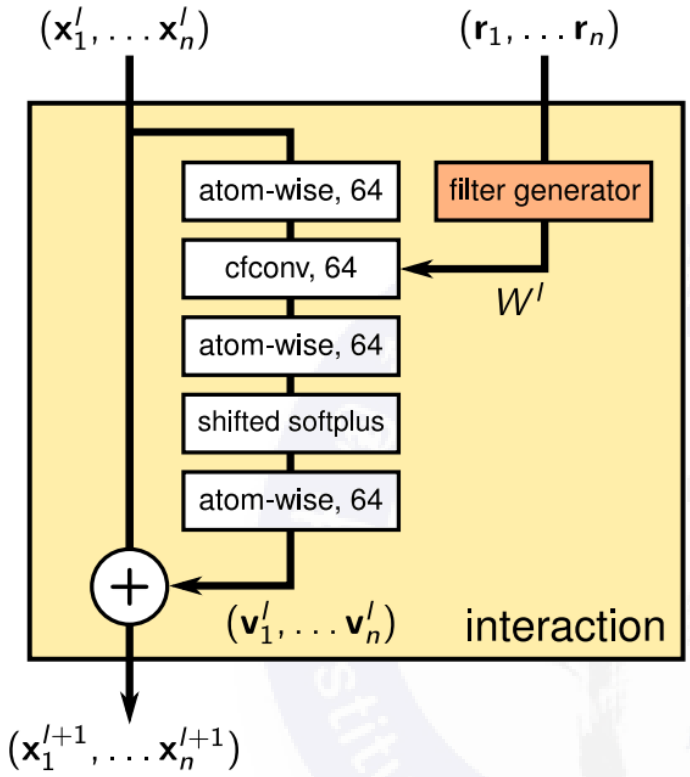
$$\mathbf{x}_i^{l+1} = (X^l * W^l)_i$$
$$= \mathbf{x}_j^l \circ W^l(\mathbf{r}_j - \mathbf{r}_i)$$

Interaction blocks

- Activation functions: shifted softplus

$$\mathrm{ssp}(x)=\ln(0.5\, e^{x}+0.5)$$

  - ssp(0) = 0.
  - Improves the convergence of the network while having infinite order of continuity.

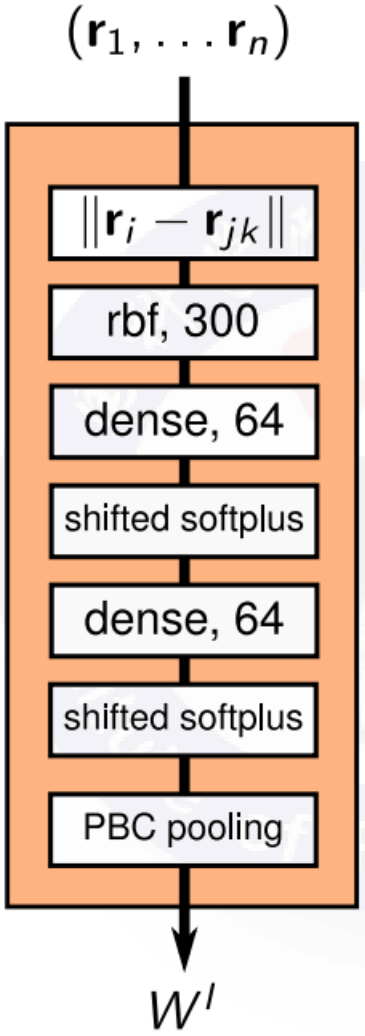


- Obtain
  - smooth potential energy surfaces.
  - force fields.
  - second derivatives that are required for training with forces as well as the calculation of vibrational modes.

Filter-generating networks

- Determines how interactions between atoms are modeled.
- Constrain the model and include chemical knowledge.
- Input: a fully-connected neural network that takes the vector pointing from atom *i* to its neighbor *j*.
- Rotationally invariant: requirements for modeling molecular energies. Obtained by using interatomic distances:

$$d_{ij} = \|\mathbf{r}_i - \mathbf{r}_j\|$$



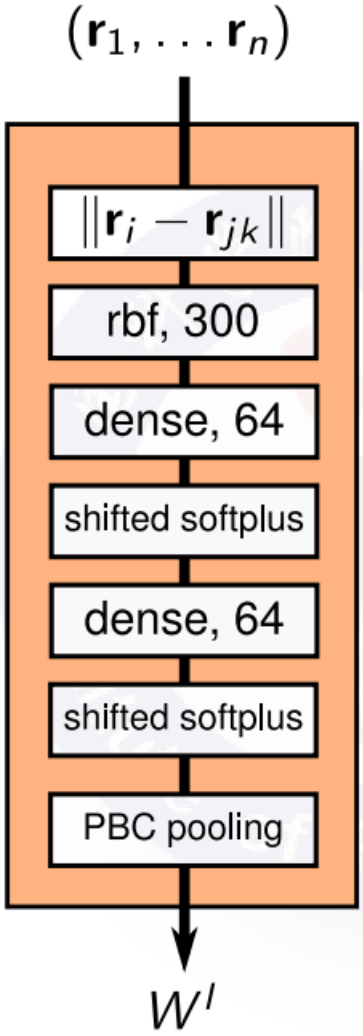
Filter-generating networks

Rotationally invariant

- Filters would be highly correlated: a neural network after initialization is close to linear.
- Expand the distances in a basis of Gaussians:

$$e_k(\mathbf{r}_i - \mathbf{r}_j) = \exp\left[-\gamma\left(\|\mathbf{r}_i - \mathbf{r}_j\| - \mu_k\right)^2\right]$$

- μ_k: chosen on a uniform grid between zero and the distance cutoff.
- The number of Gaussians and the hyper parameter γ determine the resolution of the filter.



Filter-generating networks

Periodic boundary conditions

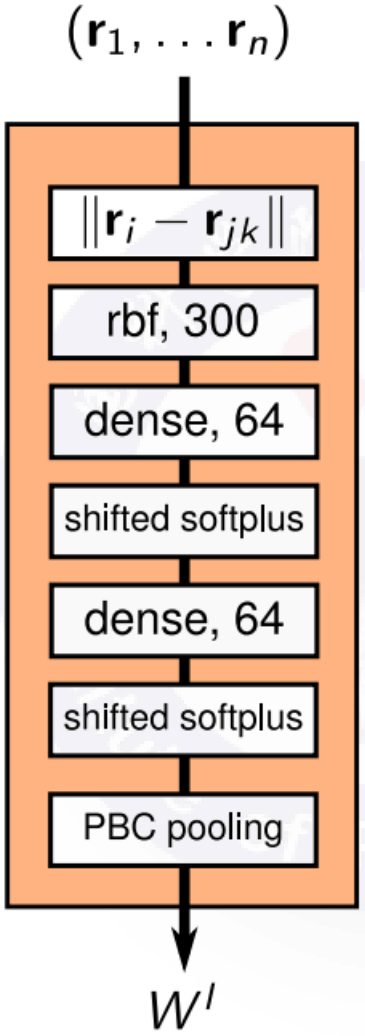
- Each atom-wise feature vector $\mathbf{x}_i$ has to be equivalent across all periodic repetitions.
- Given a filter $\tilde{W}^l(\mathbf{r}_{jb} - \mathbf{r}_{ia})$ *a,b*: unit cell

over all atoms with $\|\mathbf{r}_{jb} - \mathbf{r}_{ia}\| < r_{\text{cut}}$.

$$\mathbf{x}_i^{l+1} = \mathbf{x}_{im}^{l+1} = \frac{1}{n_{\text{neighbors}}} \sum_{j,n} \mathbf{x}_{jn}^l \circ \tilde{W}^l(\mathbf{r}_{jn} - \mathbf{r}_{im})$$

$$= \frac{1}{n_{\text{neighbors}}} \sum_j \mathbf{x}_j^l \circ \underbrace{\left(\sum_n \tilde{W}^l(\mathbf{r}_{jn} - \mathbf{r}_{im}) \right)}_{W}$$

- More stable when normalizing the filter response $\mathbf{x}_i^{l+1}$ by the number of atoms within the cutoff range.

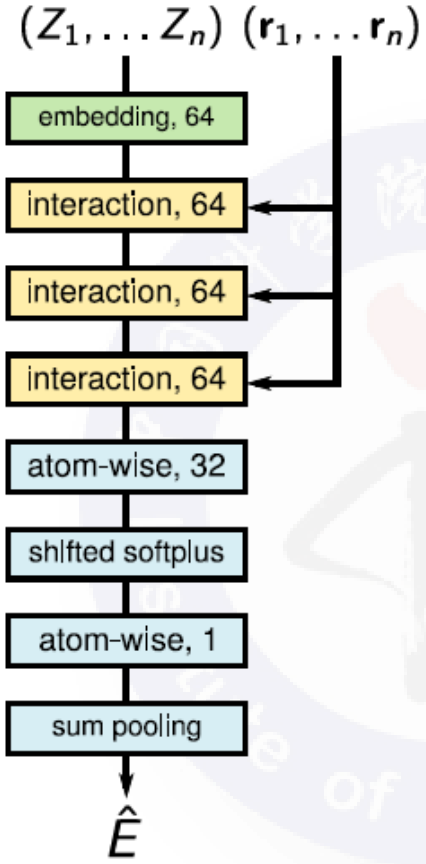


Atom-wise layers

- Are dense layers.
- Applied separately to the representations $\mathbf{x}_i^l$ of each atom *i*:

$$\mathbf{x}_i^{l+1} = W^l \mathbf{x}_i^l + \mathbf{b}^l$$

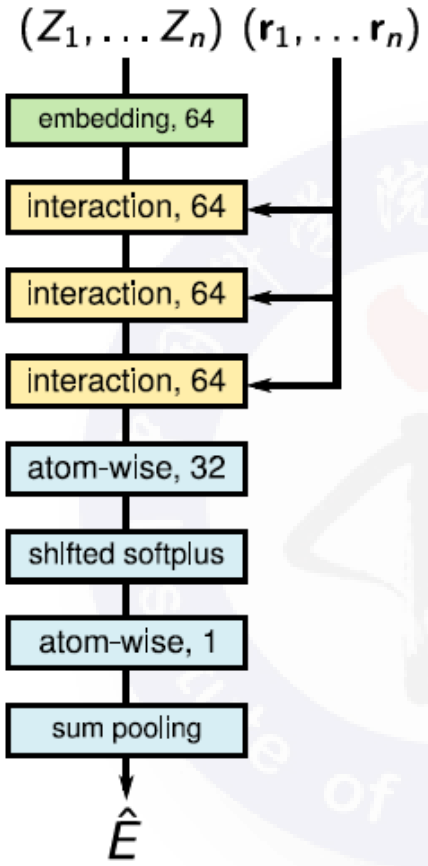
  - Weights W^l and biases $\mathbf{b}^l$ are shared across atoms.
  - The architecture remains scalable with respect to the number of atoms.
- These layers are responsible for the recombination of feature maps.



Property prediction

- Compute atom-wise contributions $\hat{P}_i$ from the fully-connected prediction network.
- Calculate the final prediction $\hat{P}$ by summing (intensive) or averaging (extensive) over the atomic contributions, respectively.
- As SchNet yields rotationally invariant energy predictions, the force predictions are rotationally equivariant by construction.
- Predicting atomic forces:

$$\hat{\mathbf{F}}_i(Z_1,\ldots,Z_n,\mathbf{r}_1,\ldots,\mathbf{r}_n) = -\frac{\partial \hat{E}}{\partial \mathbf{r}_i}(Z_1,\ldots,Z_n,\mathbf{r}_1,\ldots,\mathbf{r}_n)$$



Training

- Train SchNet for each property target *P* by minimizing the squared loss:

$$\ell(\hat{P}, P) = \left\| P - \hat{P} \right\|^2$$

- Train energies and forces with combined loss:

$$\ell\left(\left(\hat{E}, \mathbf{F}_1, \ldots, \mathbf{F}_n\right), \left(E, \mathbf{F}_1, \ldots, \mathbf{F}_n\right)\right) = \rho \left\| E - \hat{E} \right\|^2 + \frac{1}{n_{\text{atoms}}} \sum_{i=0}^{n_{\text{atoms}}} \left\| \mathbf{F}_i - \left(-\frac{\partial \hat{E}}{\partial \mathbf{R}_i} \right) \right\|^2$$

  - ρ: trade-off between energy and force loss.

- In each experiment, we split the data into a training set of given size *N* and use a validation set for early stopping.

- Remaining data is used for computing the test errors.

Kaggle: predict scalar coupling constant

Coupling Tensor → **Scalar Coupling Constant**

$$\ddot{J}_{KL} = h\frac{\gamma_K}{2\pi}\frac{\gamma_L}{2\pi}\frac{d^2E}{d\mathbf{M}_K d\mathbf{M}_L} \quad \rightarrow \quad J_{KL} = \frac{1}{3}\mathrm{Tr}\ddot{J}_{KL}$$

Electronic Energy $E(\mathbf{M}_K, \mathbf{M}_L; \lambda_S, \lambda_T)$

Quantum Mechanics

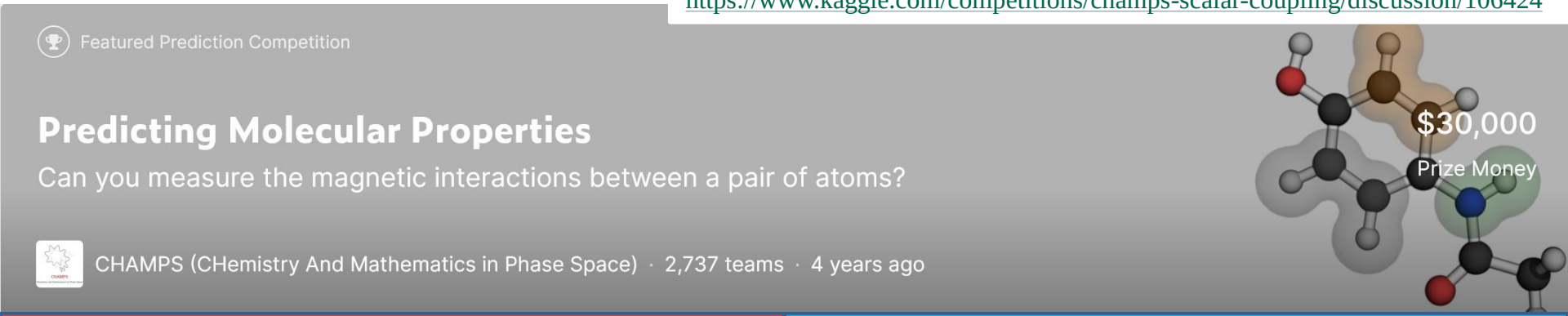
$$\frac{d^2E}{d\mathbf{M}_K d\mathbf{M}_L} = \frac{\partial^2 E}{\partial \mathbf{M}_K \partial \mathbf{M}_L} + \frac{\partial^2 E}{\partial \mathbf{M}_K \partial \lambda_S}\frac{\partial \lambda_S}{\partial \mathbf{M}_L} + \frac{\partial^2 E}{\partial \mathbf{M}_K \partial \lambda_T}\frac{\partial \lambda_T}{\partial \mathbf{M}_L}$$

DSO — Ground State; PSO — Singlet and Triplet Excited States; FC+SD

SchNet

$$\frac{d^2E}{d\mathbf{M}_K d\mathbf{M}_L} = \sum_{i=1}^{n}\frac{d^2 e_i}{d\mathbf{M}_K d\mathbf{M}_L} = \sum_{i=1}^{n}\ddot{\zeta}_i$$

https://www.kaggle.com/competitions/champs-scalar-coupling/discussion/106424



Kaggle

All | Your Work | Shared With You | Bookmarks | Most Votes

Molecular Properties EDA and models
Updated 4y ago
Score: 0.289 · 131 comments · Predicting Molecular Properties +1
674 · Gold

Brute force feature engineering
Notebook copied with edits from Andrew Lukyanenko · Updated 4y ago
Score: -0.595 · 72 comments · Predicting Molecular Properties +1
373 · Gold

Using meta-features to improve model
Updated 4y ago
Score: -0.721 · 46 comments · Predicting Molecular Properties
260 · Gold

Keras Neural Net for CHAMPS
Updated 4y ago
Score: -1.073 · 38 comments · Predicting Molecular Properties +1
259 · Gold

How To: Easy Visualization of Molecules.
Updated 4y ago
31 comments · Predicting Molecular Properties
248 · Gold

Distance - is all you need. LB -1.481
Notebook copied with edits from a private notebook · Updated 4y ago
Score: -1.481 · 24 comments · Predicting Molecular Properties
199 · Gold

Giba R + data.table + Simple Features -1.17 LB
Notebook copied with edits from a private notebook · Updated 4y ago
Score: -1.176 · 53 comments · Predicting Molecular Properties
196 · Gold

Kaggle: Community

Andrew Lukyanenko
Posted 4 years ago · 8th in this Competition

Yes, his MPNN was a great beginning for building advanced neural net. Though the code itself... you know... 😂

Reply

8

Mycroft Holmes
Posted 4 years ago · 57th in this Competition

@hengck23 is #3 discussion Grandmaster. The reward system seems adequate.

Reply

1

CPMP
Posted 4 years ago · 7th in this Competition

We may be an exception but none of us reused Heng's code, we all used keras ;)

Reply

1

Alexpartisan
Posted 4 years ago · 15th in this Competition

Shout out to Heng!

Reply

1

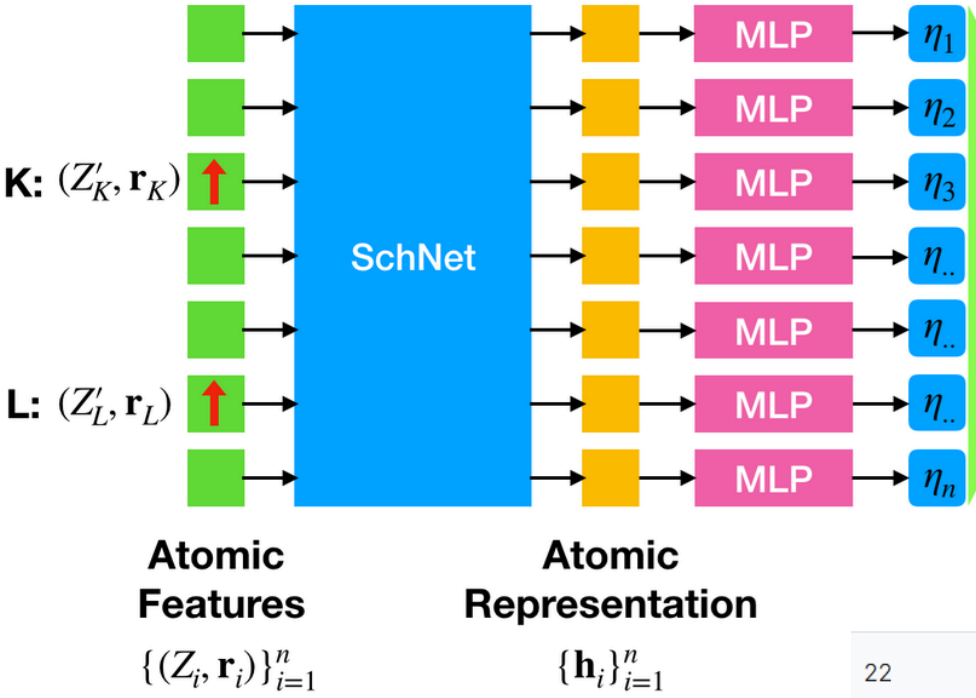
XiaokangWang
Posted 4 years ago · 121st in this Competition

I also benefited a lot from Heng' code. He is kind of like Andrew Ng, who made machine learning available to everyone what is interested.

Reply

2

Kaggle: predict scalar coupling constant



Scalar Coupling Constant

$$J_{KL} = \sum_i \eta_i$$

$$\eta_i = h\frac{\gamma_K}{2\pi}\frac{\gamma_L}{2\pi}\frac{1}{3}\mathrm{Tr}\ddot{\zeta}_i$$

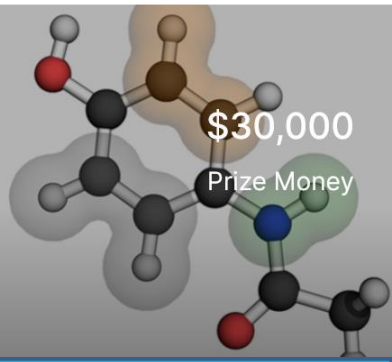
22	—	Vanilla SchNet (Tony Y.)	

https://www.kaggle.com/competitions/champs-scalar-coupling/discussion/106424

Featured Prediction Competition

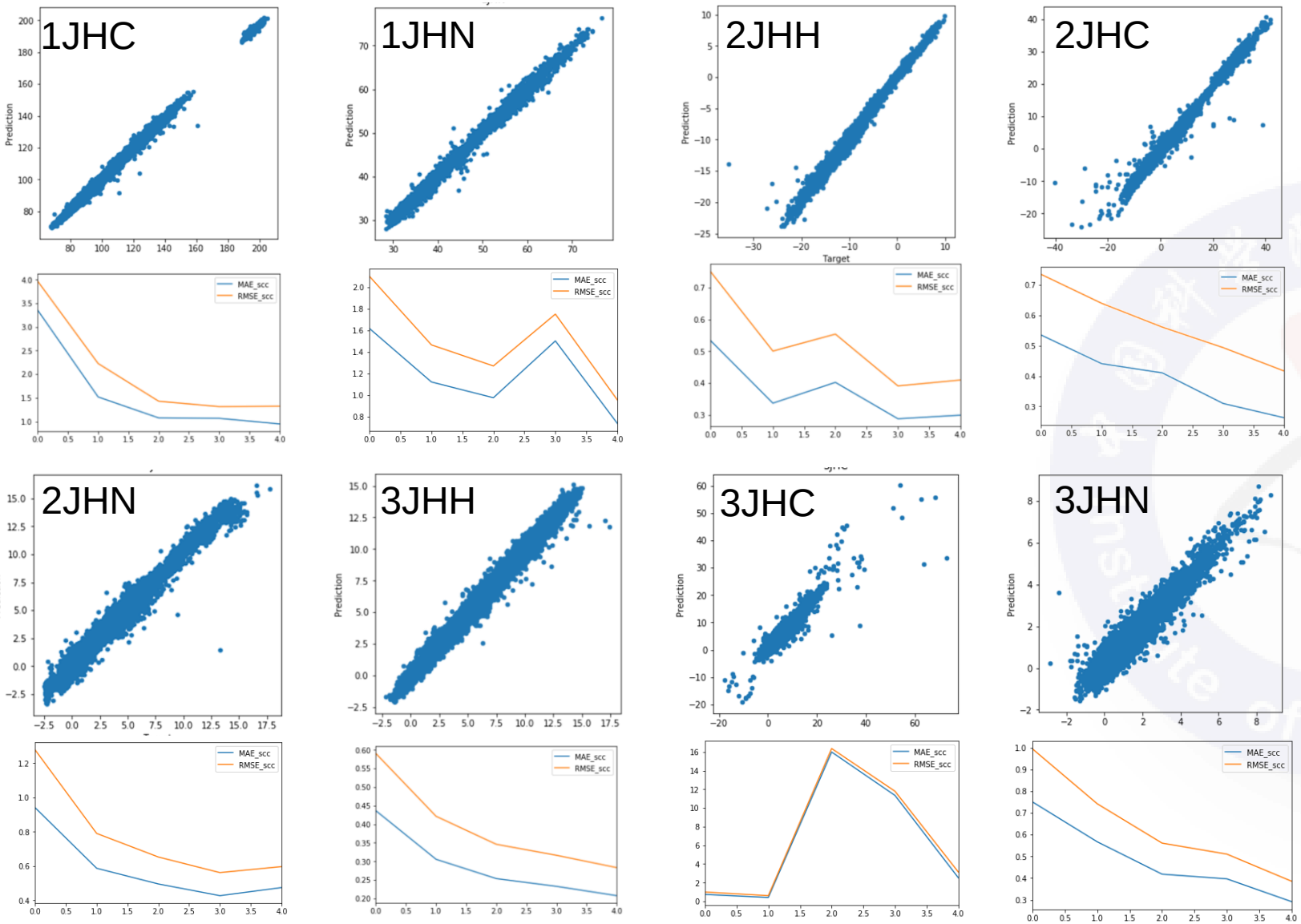
Predicting Molecular Properties

Can you measure the magnetic interactions between a pair of atoms?

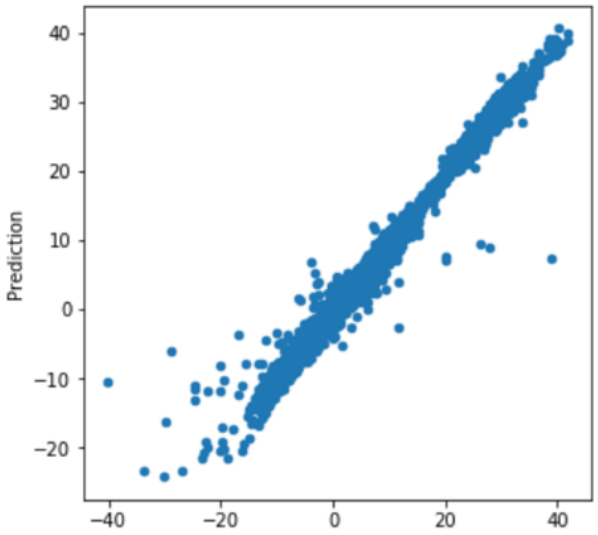


CHAMPS (CHemistry And Mathematics in Phase Space) · 2,737 teams · 4 years ago

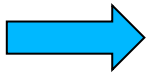
Result

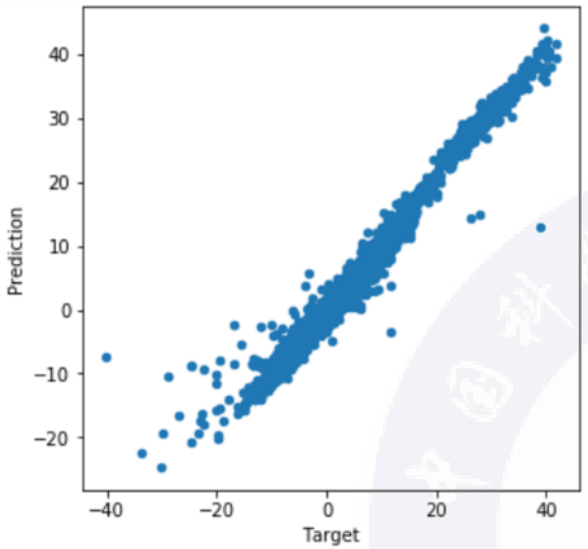

1JHC
1JHN
2JHH
2JHC
2JHN
3JHH
3JHC
3JHN
Prediction
Target
MAE_scc
RMSE_scc

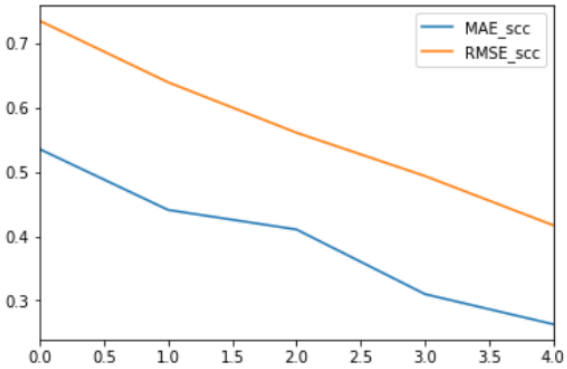
Test: increase epoch steps to 10

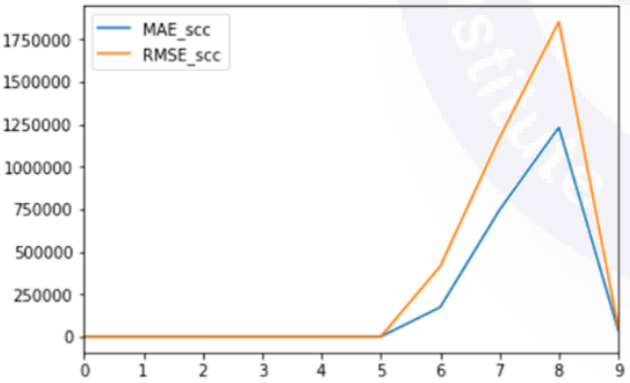


2JHC

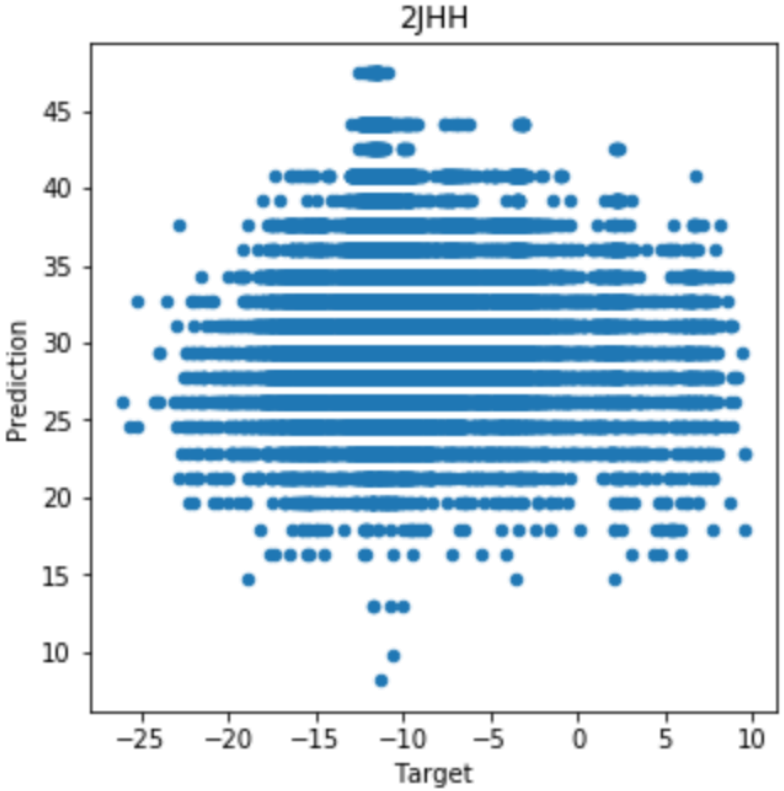






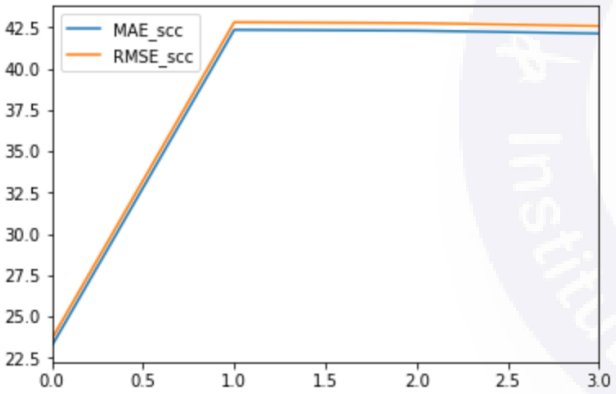


Test: increase learning rate (3 epochs)



	Time	Learning rate	Train loss	Validation loss	MAE_scc	RMSE_scc
0	405.813638	0.01	4.721478e+12	559.274815	23.224133	23.648992
1	814.477964	0.01	9.940983e+18	1832.295706	42.347378	42.805323
2	1224.116501	0.01	4.320177e+06	1827.820094	42.295350	42.753013
3	1635.426513	0.01	4.299901e+06	1813.133102	42.124165	42.580901

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ade7efdd2e8>
```

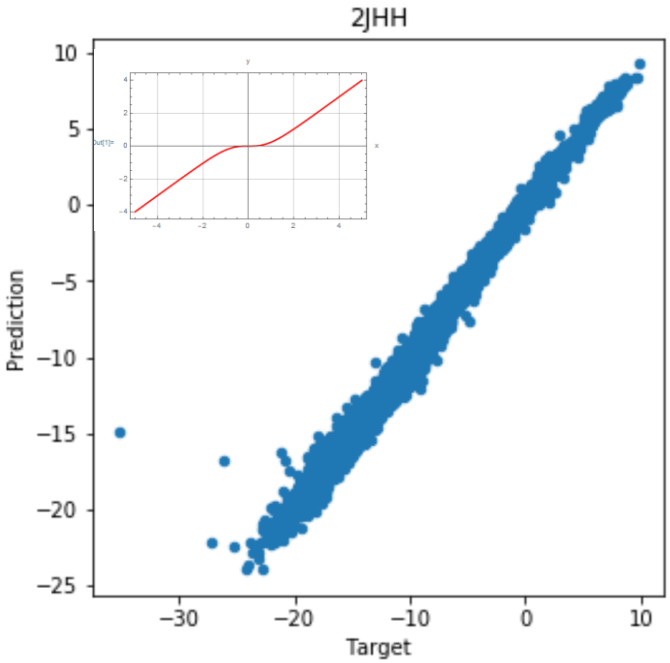


Test: change activation function

TORCH.NN.FUNCTIONAL.TANHSHRINK

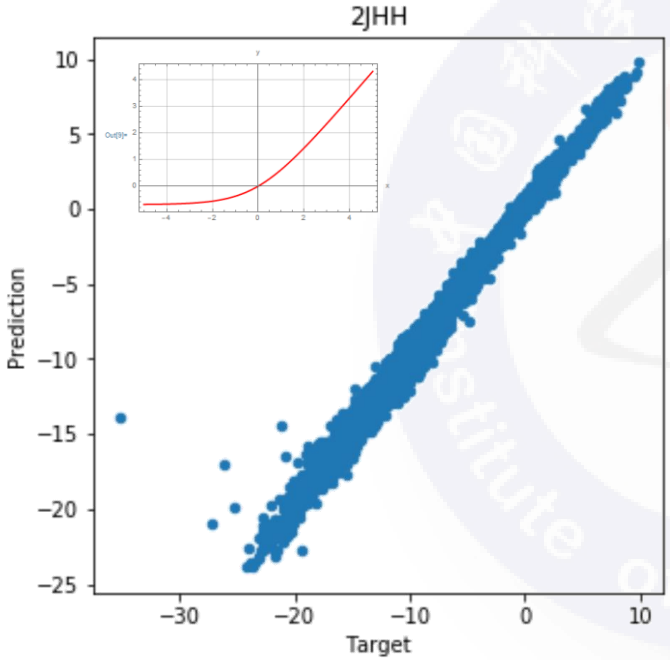
torch.nn.functional.tanhshrink(*input*) → Tensor [SOURCE]

Applies element-wise, $\text{Tanhshrink}(x) = x - \text{Tanh}(x)$



	RMSE	MAE	log(MAE)
0	0.346743	0.233726	-1.453607

Default: ssp(x)=ln(0.5 e^x+0.5)



	RMSE	MAE	log(MAE)
0	0.406484	0.287703	-1.245828

Score & Rank

1380th

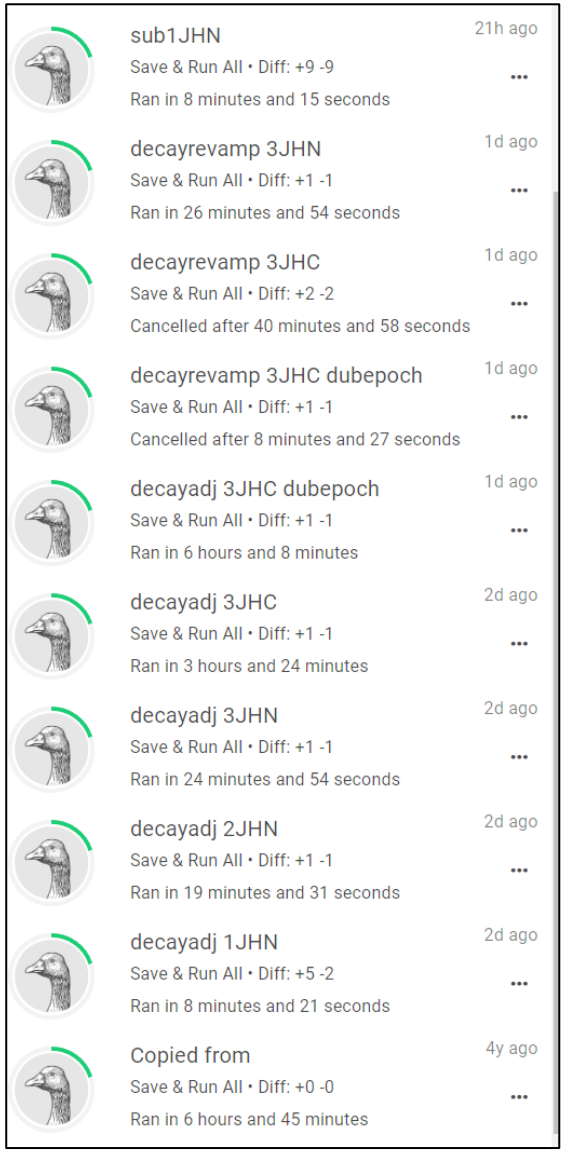
2,737 teams

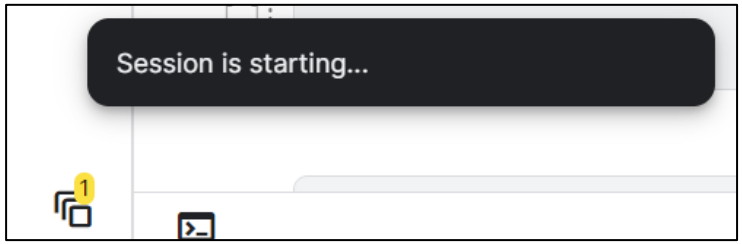
Submission and Description	Private Score	Public Score
submission1.csv Complete (after deadline) · 8m ago · Test.	**-0.94916**	**-0.95319**

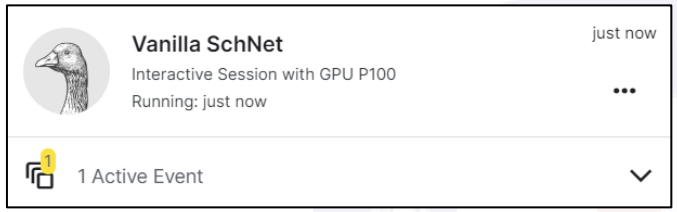
1378		-0.95711
1379		-0.95659
1380		-0.95096
1381		-0.95002

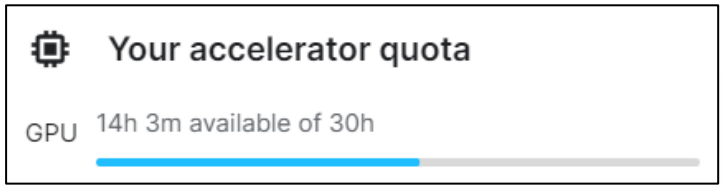
Repo: https://github.com/InterStellarAlice/MolProPred

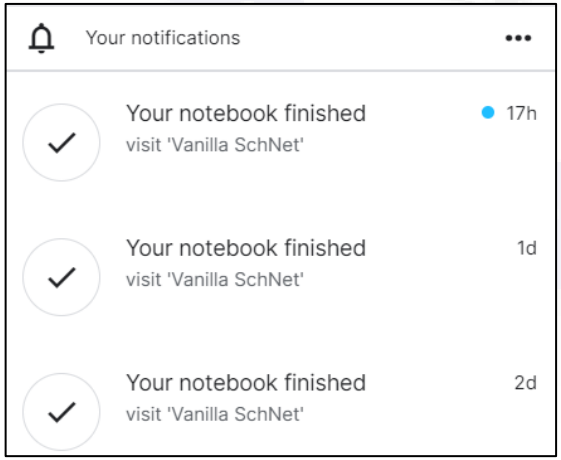
A Wonderful Experience



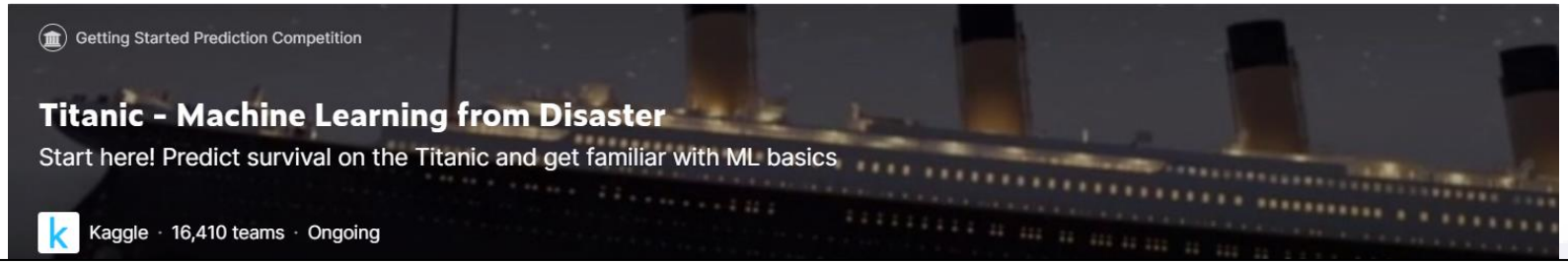








The Real First Step

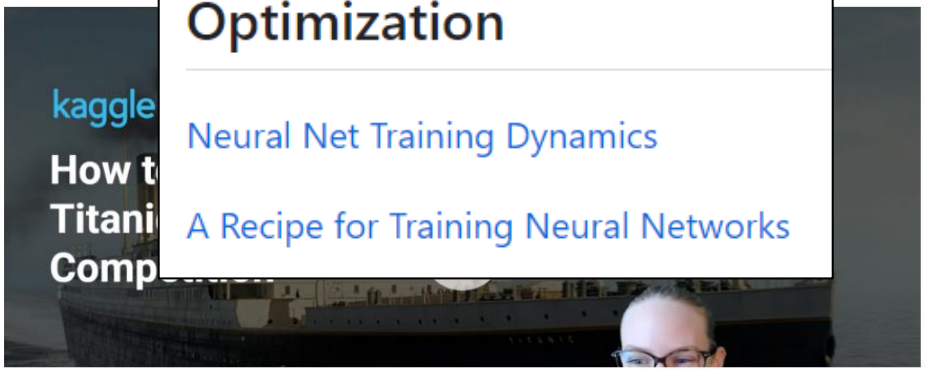


main ml4p / resources / Go to file

wangleiphy slides ✓ 5257f1b last week History

..		
README.md	slides	last week
jax.pdf	rename	last month

make your first submission!



THANKS!