XACS
Xiamen Atomistic Computing Suite
XACScloud.com

MLatom.com

ML potentials and their training

Pavlo O. Dral

Xiamen University, P.R. China

Visiting Professor in
Nicolaus Copernicus University, Poland

4 July 2024

You have already used the KREG models.

- The operating principle of these models is the same: for a given geometry, they predict energies.
- Such models are called **machine learning (interatomic) potentials** (MLP or MLIP), as they represent potential energy surfaces (PES) of molecules (function of energy E with respect to nuclear coordinates R in the Born–Oppenheimer approximation):

$$E = f(\mathbf{R})$$

- QM methods provide the first-principles way to calculate this energy, but they are slow.

Chemistry and Machine Learning

(Supervised)
Machine learning serves for function approximation[1]

ML takes little time for making new predictions

Quantum Chemical Property(molecule) = function(nuclear coordinates)

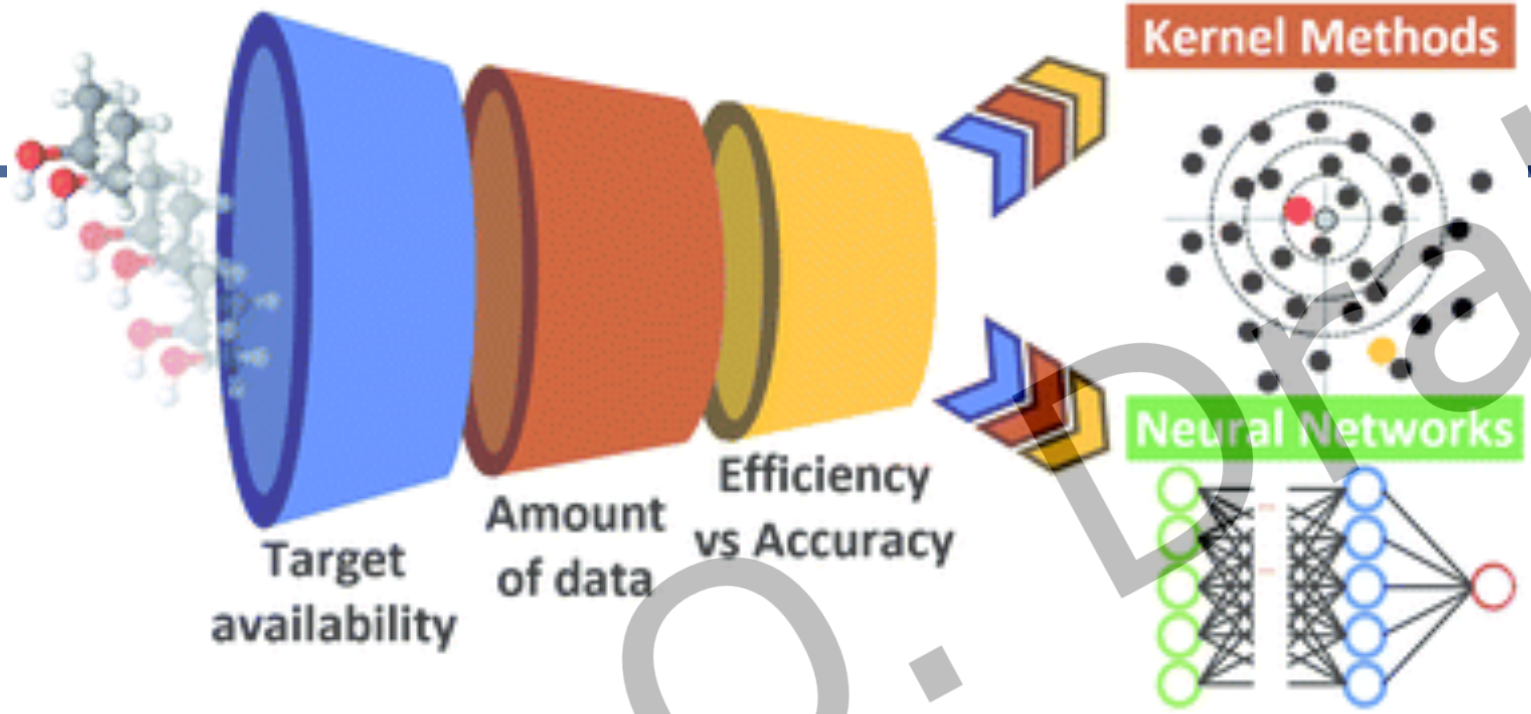
[1] Hastie, Tibshirani, Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed.; Springer-Verlag, **2009**

Zoo of machine learning potentials

Slide by Fuchun Ge

Mario Barbatti

Max Pinheiro Jr



Fuchun Ge

What machine learning potential to choose?

M. Pinheiro Jr, F. Ge, N. Ferré, P. O. Dral, M. Barbatti. *Chem. Sci.* **2021**, *12*, 14396–14413
P. O. Dral, F. Ge, B.-X. Xue, Y.-F. Hou, M. Pinheiro Jr, J. Huang, M. Barbatti. *Top. Curr. Chem.* **2021,** *379*, 27

Categories of machine learning potentials

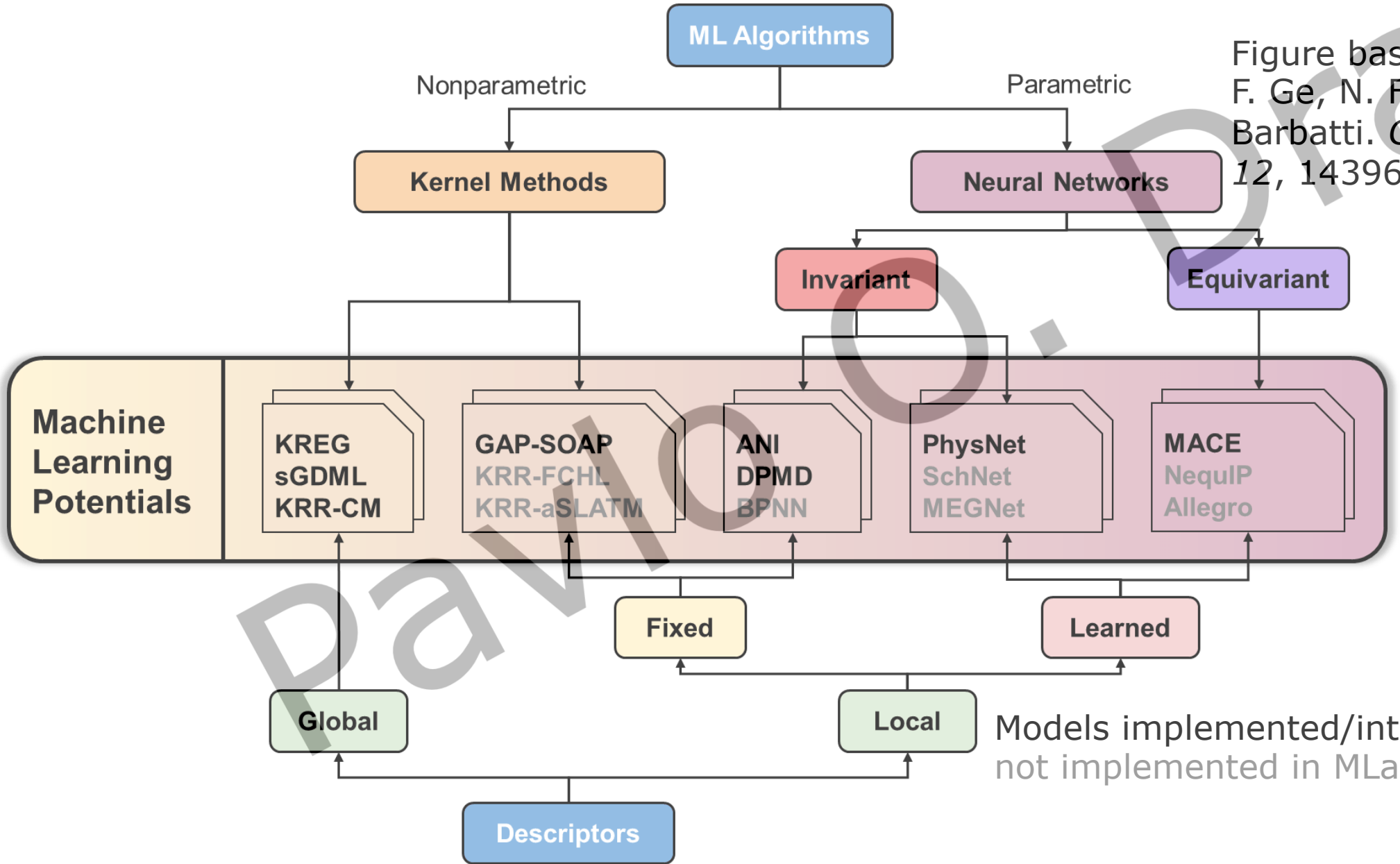


Figure based on M. Pinheiro Jr, F. Ge, N. Ferré, P. O. Dral, M. Barbatti. *Chem. Sci.* **2021**, *12*, 14396–14413

Models implemented/interfaced in MLatom

not implemented in MLatom

Example PES1.

Train the ANI-type MLP for the H_2 molecule (see instructions below) and obtain the bond length with this model.

Questions:

1. How much time did it take to train the model?
2. What are the training and validation errors?
3. What is the bond length of H_2 obtained with your model?
4. How do these results compare to the KREG model you used in previous task?

Training the model

```
createMLmodel                # Specify the task for MLatom
MLmodelType=ANI              # Specify the model type
MLmodelOut=energies_ani.pt   # Save model in energies_ani.pt
XYZfile=h2.xyz               # File with XYZ geometries
Yfile=E_FCI_451.dat          # The file with FCI energies but can be any other property
ani.max_epochs=1000          # Only train 1000 epochs
```

Using the model, e.g., for optimization

```
geomopt                      # Request geometry optimization
MLmodelType=ANI              # of the KREG type
MLmodelIn=energies_ani.pt    # in energies_ani.pt file
XYZfile=h2_init.xyz          # The file with initial guess
optXYZ=eq_ANI.xyz            # optimized geometry output
```

ML algorithms

Linear regression

$$f(\mathbf{x}_i;\boldsymbol{\beta}) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots$$

Kernel ridge regression (KRR)

$$f(\mathbf{x}_i;\boldsymbol{\alpha}) = \sum_{j=1}^{N_{\mathrm{tr}}} \alpha_j k(\mathbf{x}_i, \mathbf{x}_j)$$

'nonparametric model'

Examples: KREG, KRR-CM

Neural networks (NN)

Number of parameters is fixed

$$f(\mathbf{x};\boldsymbol{\beta}) = \beta_0 + \beta_1 Z_1 + \beta_2 Z_2 + \cdots$$

$$Z_m = \sigma(\alpha_{m0} + \alpha_{m1} x_1 + \alpha_{m2} x_2 + \cdots)$$

'parametric model'

Examples: ANI-1ccx, AIQM1

The KREG model

$$f(\mathbf{x}_i) = \sum_{j=1}^{N_{\text{tr}}} \alpha_j k(\mathbf{x}_i, \mathbf{x}_j)$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{2\sigma^2}\sum_{s}^{N_x}(x_{i,s} - x_{j,s})^2\right)$$

the Gaussian kernel function
(σ is the kernel width)

$$x = \begin{pmatrix} \dots & \frac{R^{eq}}{R} & \dots \end{pmatrix}^T$$

RE descriptor

Analytical solution for the regression coefficients α given N_{tr} training points

$$\begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) + \lambda & \cdots & k\left(\mathbf{x}_1, \mathbf{x}_{N_{\text{tr}}}\right) \\ \vdots & \ddots & \vdots \\ k\left(\mathbf{x}_{N_{\text{tr}}}, \mathbf{x}_1\right) & \cdots & k\left(\mathbf{x}_{N_{\text{tr}}}, \mathbf{x}_{N_{\text{tr}}}\right) + \lambda \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_{N_{\text{tr}}} \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_{N_{\text{tr}}} \end{pmatrix}$$

λ is the regularization parameter ensuring transferability

$$(\mathbf{K} + \lambda\mathbf{I})\boldsymbol{\alpha} = \mathbf{y}$$

Training NNs

Back-propagation:

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{N} (f(\mathbf{x}_i; \boldsymbol{\theta}) - y_i)^2$$

gradient descent update with learning rate γ

$$\theta_k^{(r+1)} = \theta_k^{(r)} - \gamma \frac{\partial L(\boldsymbol{\theta})}{\partial \theta_k}$$

Well parallelized:

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{N} L_i = \sum_{i=1}^{N} (f(\mathbf{x}_i; \boldsymbol{\theta}) - y_i)^2$$

The training set is often split into the minibatches (batches)

Update of parameters after the sweep over the entire training set is called an *epoch*.

P. O. Dral, A. Kananenka, F. Ge, B.-X. Xue, Neural Networks. In *Quantum Chemistry in the Age of Machine Learning*, 1st ed.; P. O. Dral, Ed. Elsevier: 2023.

Training NNs: Issues

No analytical solution, hence, one can get **different NNs fitted on the same data**!

One can exploit this:

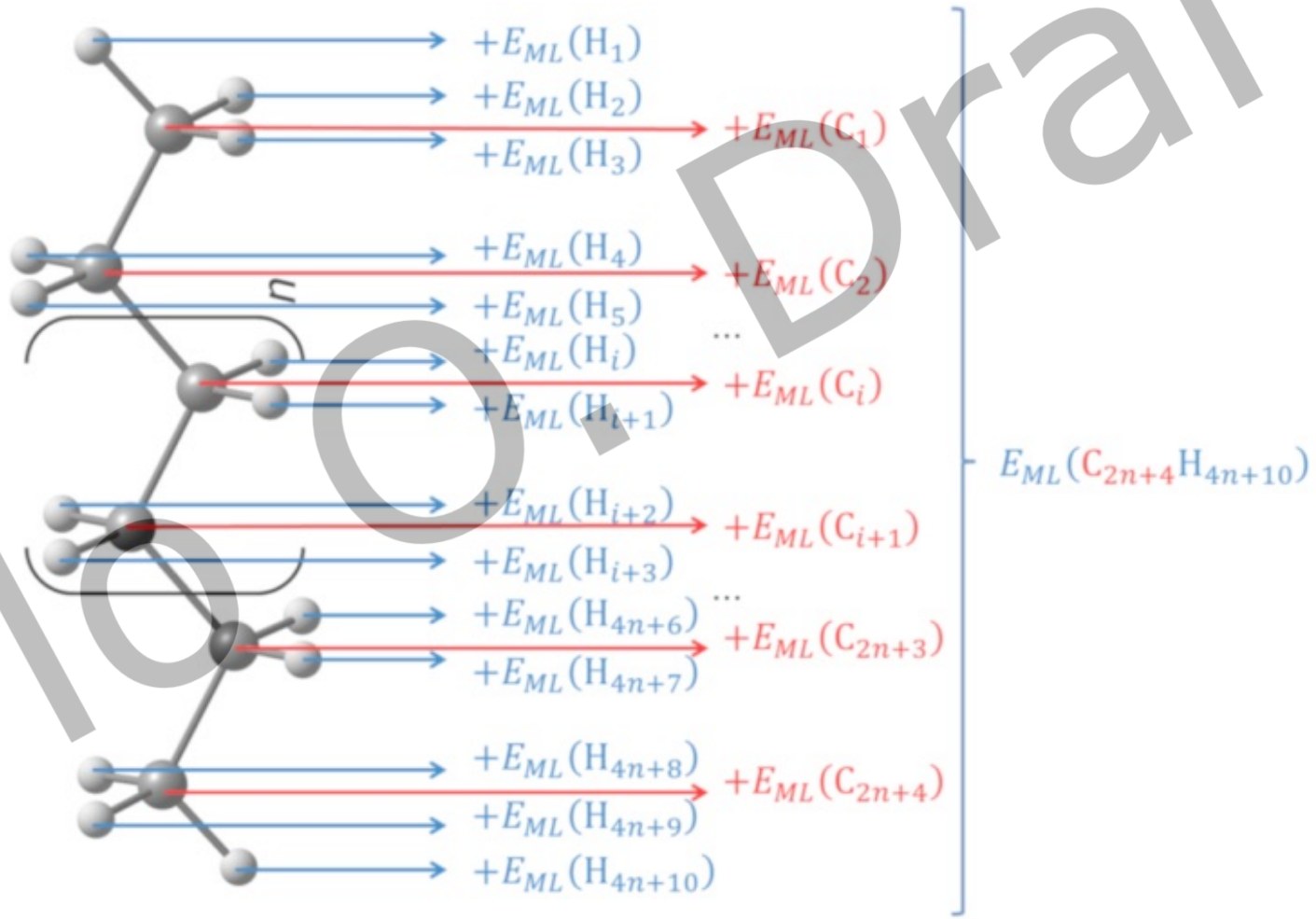
- Take average of multiple NNs to get more stable prediction
- Use deviation between NN predictions to estimate prediction uncertainty (e.g. useful in our previous examples for heats of formation and in active learning)

P. O. Dral, A. Kananenka, F. Ge, B.-X. Xue, Neural Networks. In *Quantum Chemistry in the Age of Machine Learning*, 1st ed.; P. O. Dral, Ed. Elsevier: 2023.

Locality in quantum chemistry

Behler–Parrinello
ANI
PhysNet
SchNet
GAP-SOAP
FCHL
aSLATM
…

Overview & benchmark in:
M. Pinheiro Jr, F. Ge,
N. Ferré, P. O. Dral,
M. Barbatti.
Chem. Sci. **2021**, *12*,
14396–14413



Approach: Behler, Parrinello, *Phys. Rev. Lett.* **2007**, *98*, 146401

Figure: P. O. Dral, *J. Phys. Chem. Lett.* **2020**, *11*, 2336

ANI family of potentials

ANI environment vectors (AEVs) consist of radial and angular atomic terms. Each element has its own subAEV:

$$G_k^R = \sum_{j \neq i} e^{-\eta\left(R_{ij} - R_s^{(k)}\right)^2} f_c\left(R_{ij}\right)$$

$$G_{p,q}^A = 2^{1-\zeta} \sum_{j,k \neq i} \left(1 + \cos\left(\theta_{ijk} - \theta_s^{(q)}\right)\right)^\zeta e^{-\eta\left(\frac{R_{ij}+R_{ik}}{2} - R_s^{(p)}\right)^2} f_C\left(R_{ij}\right) f_C\left(R_{ik}\right).$$

the cutoff function

$$f_{\text{cut}}(r) = \begin{cases} 1, & r \leq r_{\text{cut}} - r_\Delta, \\ \frac{1}{2}\left(\cos\left(\pi \frac{r - r_{\text{cut}} + r_\Delta}{r_\Delta}\right) + 1\right), & r_{\text{cut}} - r_\Delta < r \leq r_{\text{cut}}, \\ 0, & r > r_{\text{cut}}, \end{cases}$$

X. Gao, F. Ramezanghorbani, O. Isayev, J. S. Smith, A. E. Roitberg.
J. Chem. Inf. Model. **2020**, *60*, 3408

GAP-SOAP

Gaussian approximation potential (GAP)[1] with Smooth Overlap of Atomic Positions (SOAP)[2] descriptor

the atomic neighborhood density $$\rho_i(\mathbf{r}) = \sum_j \exp\left(-\frac{|\mathbf{r}-\mathbf{r}_{ij}|^2}{2\sigma_{atom}^2}\right) f_{\text{cut}}(|\mathbf{r}_{ij}|),$$

the cutoff function $$f_{\text{cut}}(r) = \begin{cases} 1, & r \le r_{\text{cut}} - r_\Delta, \\ \frac{1}{2}\left(\cos\left(\pi \frac{r - r_{\text{cut}} + r_\Delta}{r_\Delta}\right) + 1\right), & r_{\text{cut}} - r_\Delta < r \le r_{\text{cut}}, \\ 0, & r > r_{\text{cut}}, \end{cases}$$

[1] A. P. Bartók, M. C. Payne, R. Kondor, G. Csányi, Phys. Rev. Lett. **2010**, *104*, 136403
[2] A. P. Bartók, R. Kondor, G. Csányi, Phys. Rev. B **2013**, *87*, 187115

PhysNet

PhysNet is using message-passing NN and so called 'learned' local descriptors

the embedding vector

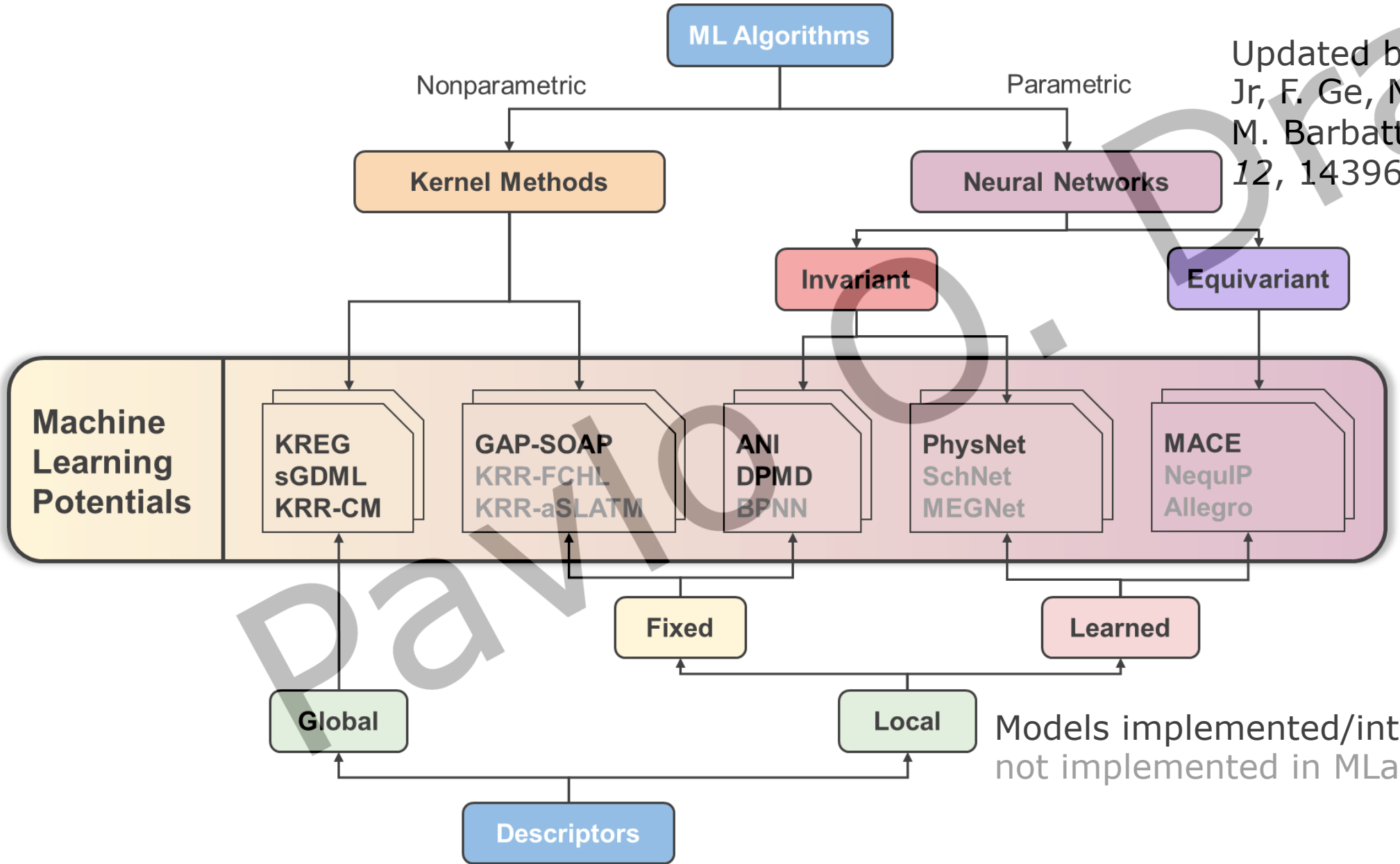
$$\mathbf{x}_i^0 = \mathbf{e}_{Z_i}$$

the coordinates are transformed to

$$g_k(r_{ij}) = f_c(r_{ij}) \cdot e^{-\beta_k(e^{-r_{ij}}-\mu_k)^2}$$

O. T. Unke, M. Meuwly. PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges. *J. Chem. Theory Comput.* **2019,** *15*, 3678

Categories of machine learning potentials

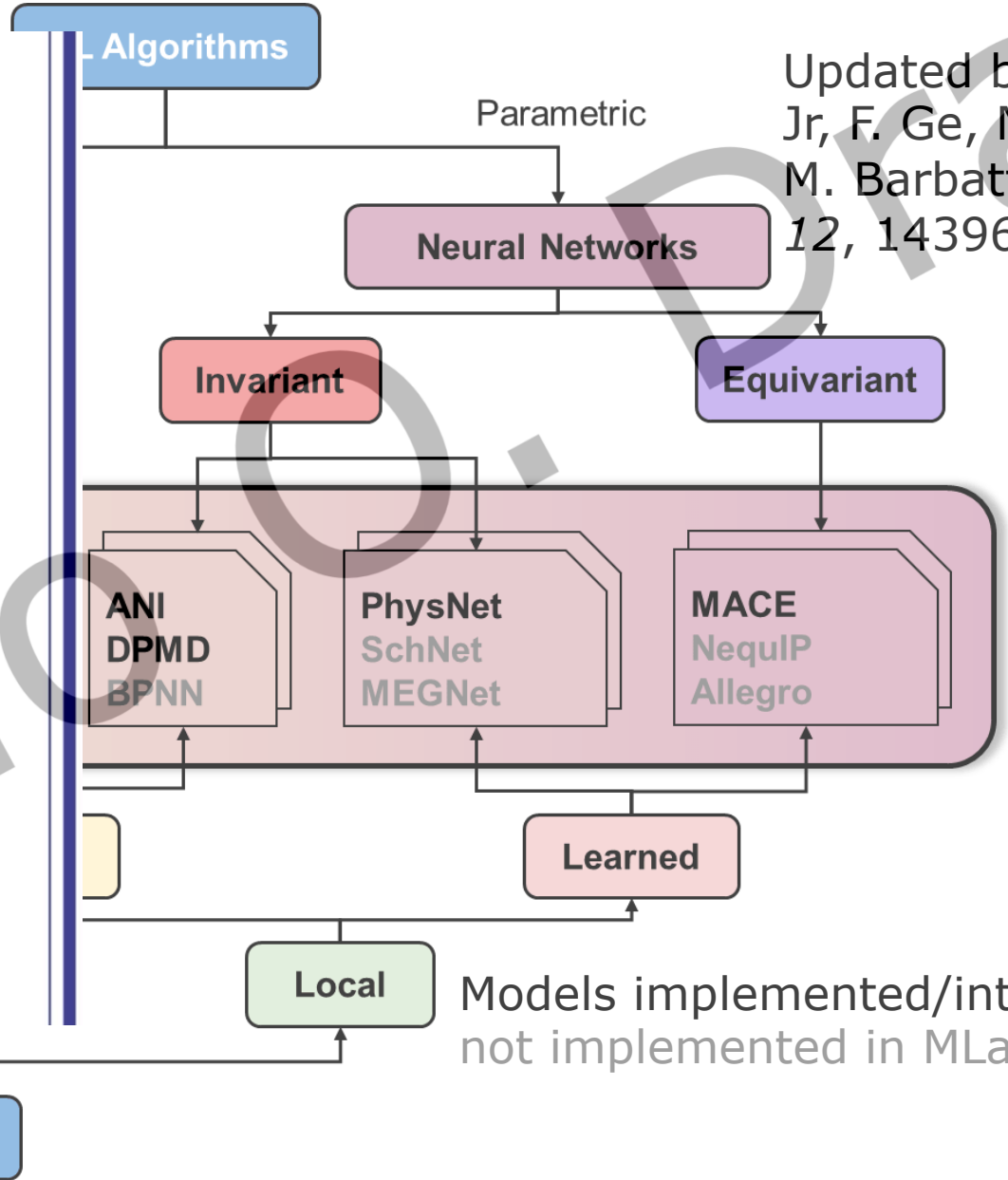


Updated based on M. Pinheiro Jr, F. Ge, N. Ferré, P. O. Dral, M. Barbatti. *Chem. Sci.* **2021**, *12*, 14396–14413

Models implemented/interfaced in MLatom

not implemented in MLatom

Categories of machine learning potentials

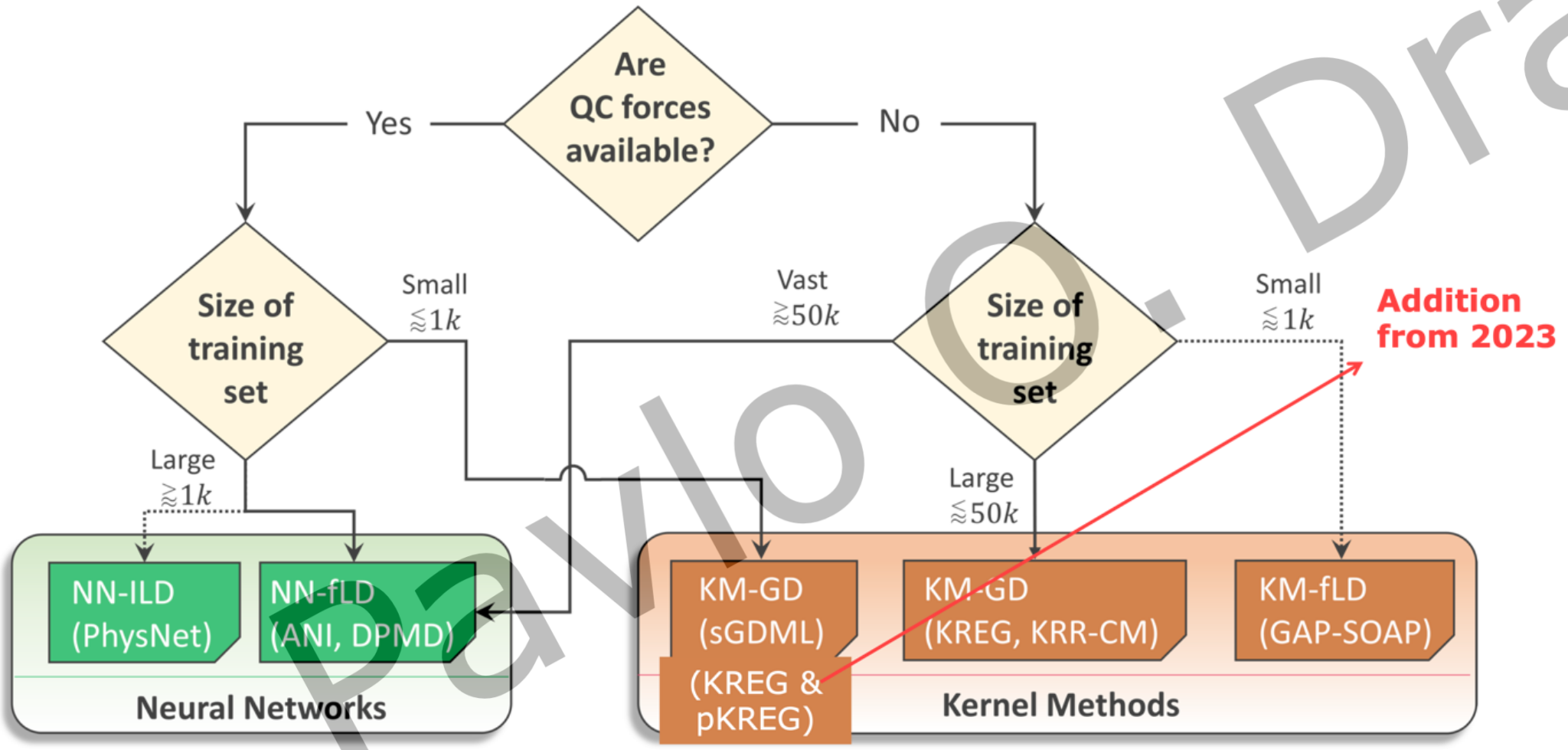


Updated based on M. Pinheiro Jr, F. Ge, N. Ferré, P. O. Dral, M. Barbatti. *Chem. Sci.* **2021**, *12*, 14396–14413

Models implemented/interfaced in MLatom
not implemented in MLatom

Choosing the 'Right' ML Potential for a single molecule

Recommendations from 2021



KREG & pKREG: Y.-F. Hou, F. Ge, P. O. Dral. *J. Chem. Theory Comput.* **2023**, *19*, 2369–2379

After 2021, open questions such as: where is the place for equivariant NNs?

M. Pinheiro Jr, F. Ge, N. Ferré, P. O. Dral, M. Barbatti. *Chem. Sci.* **2021**, *12*, 14396–14413

Example PES2.

Train the KREG model for the H_2 molecule with the settings provided below. With this model, calculate energies for the potential energy curve of H_2 (see instructions below).

Questions:

1. What is the training error (RMSE) of the model?
2. Does the potential energy curve (energy as a function of the bond length) look reasonable?

Input file for training KREG model:

```
createMLmodel           # Specify the task for MLatom
MLmodelType=KREG        # Specify the model type
MLmodelOut=kreg.unf # Save model in kreg.unf
XYZfile=h2.xyz          # File with XYZ geometries
Yfile=E_FCI_451.dat     # The file with FCI energies
Ntrain=50
sigma=0.01
lambda=0.0
```

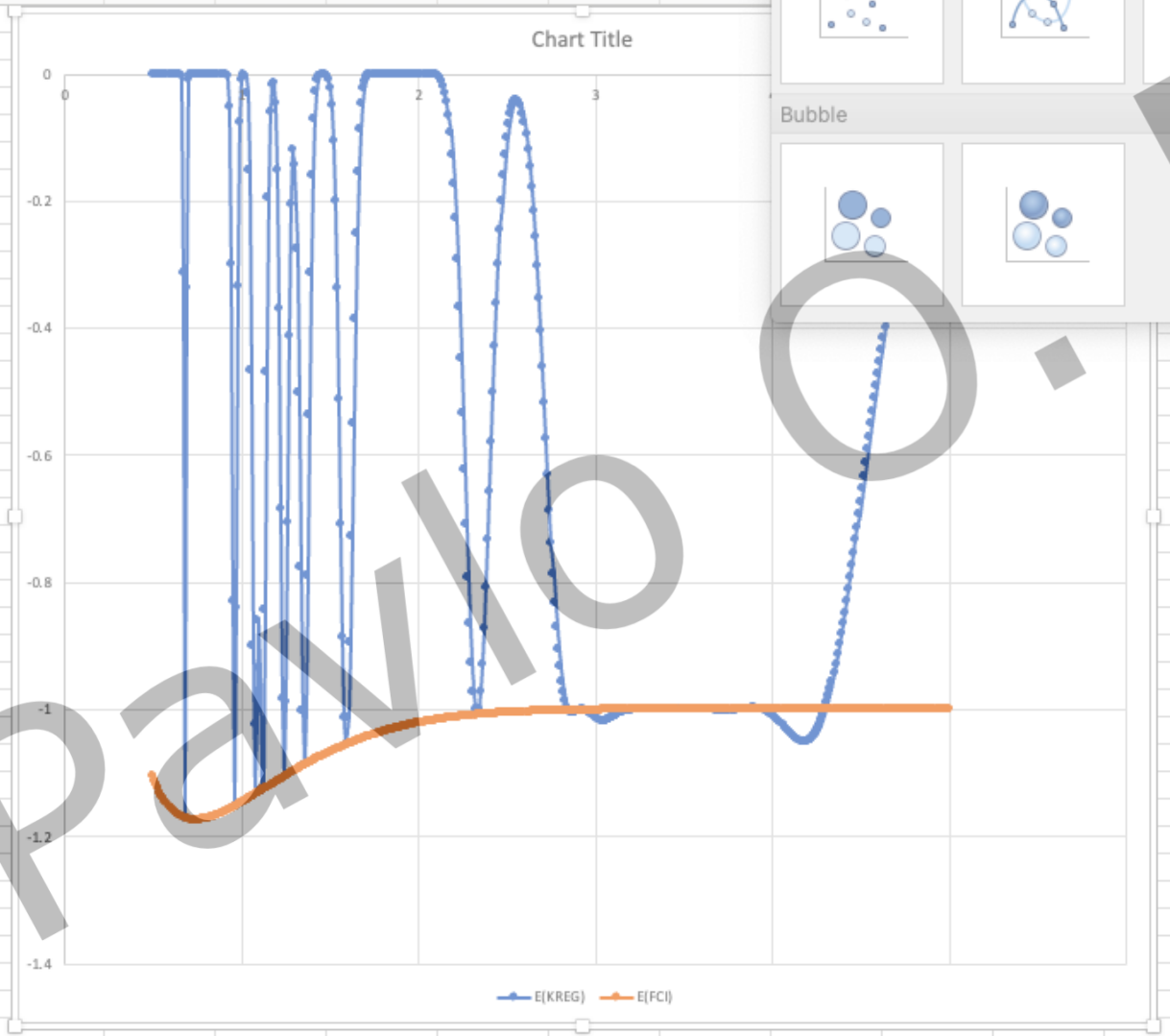
Input file for predicting energies with the ML model:

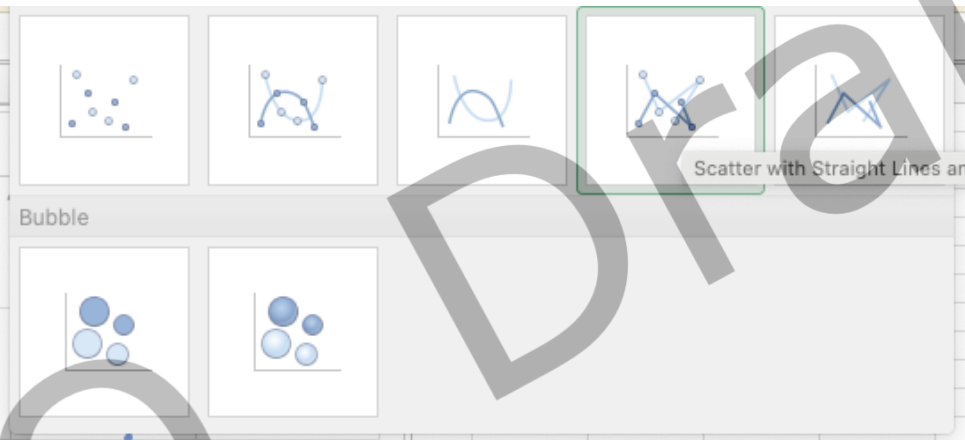
```
useMLmodel          # Specify the task for MLatom
MLmodelType=KREG        # Specify the model type
MLmodelIn=kreg.unf # Save model in kreg.unf
XYZfile=h2.xyz          # File with XYZ geometries
YestFile=E_FCI_451_kreg_est.dat     # The file with FCI energies
```

The predicted energies will be saved in `E_FCI_451_kreg_est.dat`. You can plot them in Excel as a function of the interatomic distance provided in the file R_451.dat and compare to the reference energies with full CI in the file E_FCI_451.dat.

Chart 2

R	E(KREG)	E(FCI)
0.5	0	-1.1041324
0.51	0	-1.1116602
0.52	0	-1.1185659
0.53	0	-1.1248928
0.54	0	-1.1306807
0.55	0	-1.1359664
0.56	0	-1.1407837
0.57	0	-1.145164
0.58	0	-1.1491365
0.59	0	-1.1527279
0.6	0	-1.1559632
0.61	0	-1.1588655
0.62	0	-1.1614566
0.63	0	-1.1637562
0.64	-1.056E-10	-1.1657832
0.65	-3.898E-06	-1.1675548
0.66	-0.0050971	-1.1690873
0.67	-0.3132403	-1.1703957
0.68	-1.1714941	-1.1714941
0.69	-0.3377581	-1.1723958
0.7	-0.0093216	-1.1731131
0.71	-3.003E-05	-1.1736575
0.72	-1.354E-08	-1.1740399
0.73	-1E-12	-1.1742703
0.74	0	-1.1743585
0.75	0	-1.1743132
0.76	0	-1.1741428
0.77	0	-1.1738553
0.78	0	-1.173458
0.79	0	-1.1729579
0.8	0	-1.1723616
0.81	0	-1.1716753
0.82	0	-1.1709047
0.83	0	-1.1700554
0.84	0	-1.1691325
0.85	0	-1.1681409
0.86	0	-1.1670852





Everyone will get different results due to random sampling!

- Need to repeat many times to collect enough statistics
- For small data – use cross validation
- (You can fix splitting in special cases like debugging)

ML: Error Estimation

- Often ML error for its own training set is close to zero
- We should estimate errors on a **completely independent test set**

Example PES3.

Estimate the accuracy of the KREG model for the H_2 molecule for the test set.

Questions:

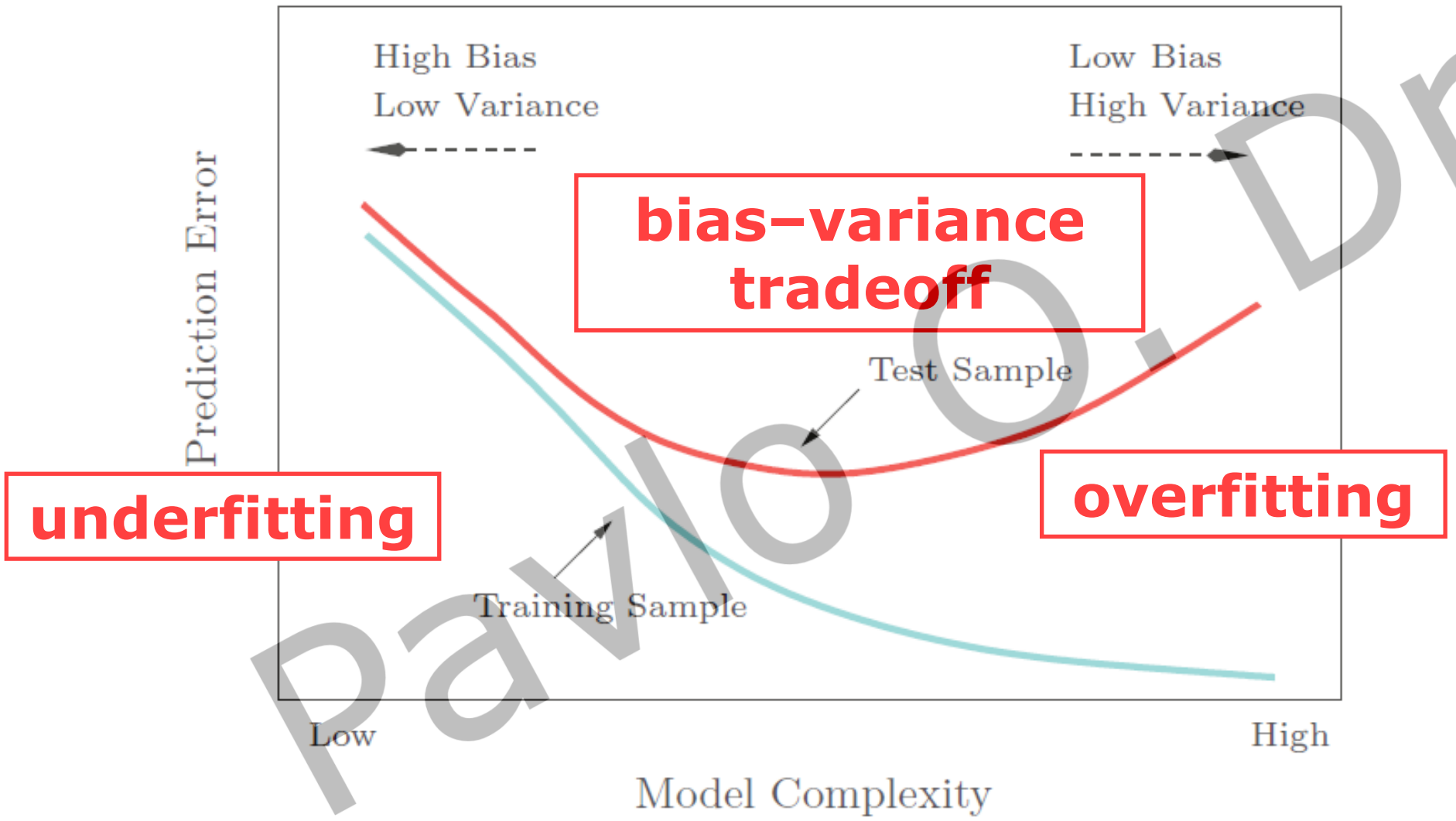
1. What is the test error (RMSE) of the model?
2. How does it compare to the training error?
3. Does the model test error reflect its performance for the potential energy curve in previous task?

Input file:

```
estAccMLmodel           # Specify the task for MLatom
MLmodelType=KREG        # Specify the model type
XYZfile=h2.xyz          # File with XYZ geometries
Yfile=E_FCI_451.dat     # The file with FCI energies
Ntrain=50
sigma=0.01
lambda=0.0
```

Use the auxiliary files from the previous tasks.

Model Selection



Hastie, Tibshirani, Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed.; Springer-Verlag, **2009**

Model selection (hyperparameter tuning)

Choosing ML model settings to make it work

The KREG model

$$f(\mathbf{x}_i) = \sum_{j=1}^{N_{\text{tr}}} \alpha_j k(\mathbf{x}_i, \mathbf{x}_j)$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{2\sigma^2}\sum_{s}^{N_x}(x_{i,s} - x_{j,s})^2\right)$$

the Gaussian kernel function
(σ is the kernel width)

$$x = \begin{pmatrix} \dots & \frac{R^{eq}}{R} & \dots \end{pmatrix}^T$$

RE descriptor

The KREG model

$$f(\mathbf{x}_i) = \sum_{j=1}^{N_{\text{tr}}} \alpha_j k(\mathbf{x}_i, \mathbf{x}_j)$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{2\sigma^2}\sum_{s}^{N_x}(x_{i,s} - x_{j,s})^2\right)$$

the Gaussian kernel function
(σ is the kernel width)

$$x = \begin{pmatrix} \dots & \frac{R^{eq}}{R} & \dots \end{pmatrix}^T$$

RE descriptor

Analytical solution for the regression coefficients α given N_{tr} training points

$$\begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) + \lambda & \cdots & k\left(\mathbf{x}_1, \mathbf{x}_{N_{\text{tr}}}\right) \\ \vdots & \ddots & \vdots \\ k\left(\mathbf{x}_{N_{\text{tr}}}, \mathbf{x}_1\right) & \cdots & k\left(\mathbf{x}_{N_{\text{tr}}}, \mathbf{x}_{N_{\text{tr}}}\right) + \lambda \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_{N_{\text{tr}}} \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_{N_{\text{tr}}} \end{pmatrix}$$

λ is the regularization parameter ensuring transferability

$$(\mathbf{K} + \lambda\mathbf{I})\boldsymbol{\alpha} = \mathbf{y}$$

Sub-training set | Validate → for tuning hyperparameters

Train

The **KREG** model (**K**ernel ridge regression [KRR] with **RE** descriptor and the **G**aussian kernel function; RE descriptor stands for Internuclear distances **R**elative to **E**quilibrium) to complete all the ML tasks.

P. O. Dral, A. Owens, S. Yurchenko, W. Thiel, *J. Chem. Phys.* **2017**, *146*, 244108

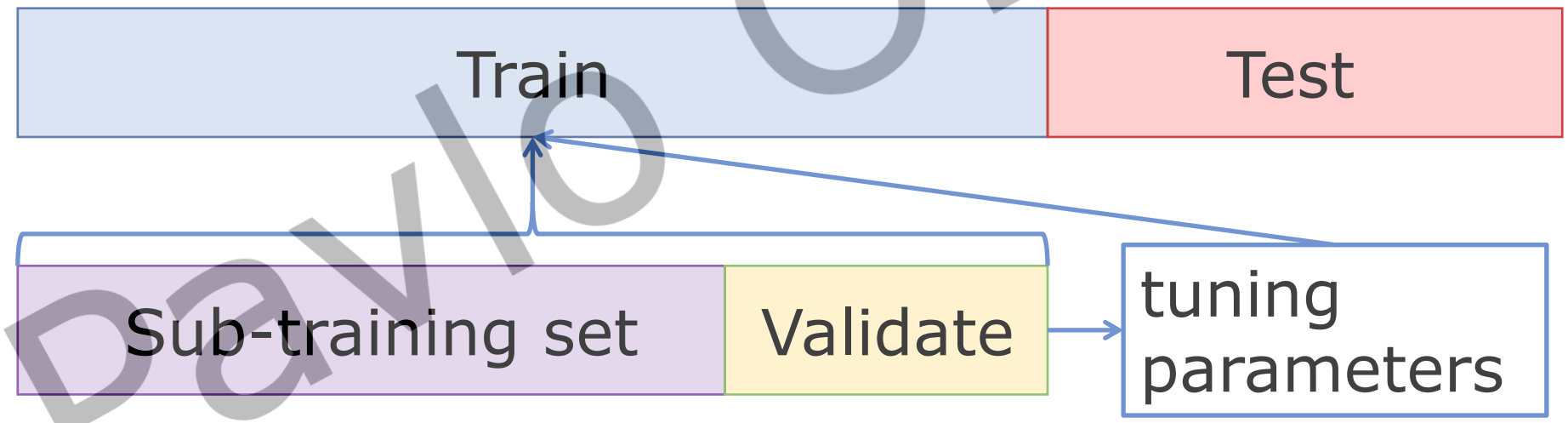
Model selection (hyperparameter tuning)

Choosing ML model settings to make it work

How to choose hyperparameters?

ML: Error Estimation

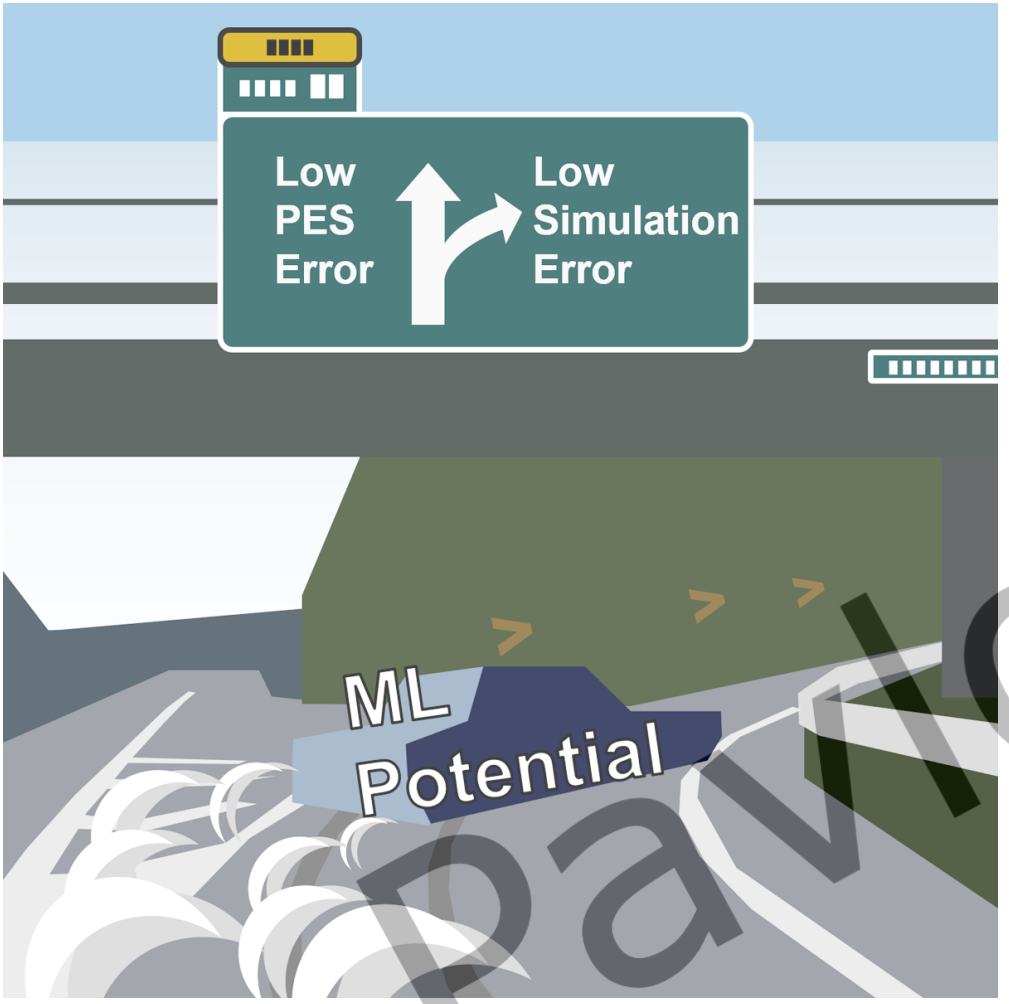
- Often ML error for its own training set is close to zero
- Using errors in the validation set would be also incorrect, because their minimization is a part of the training process
- We should estimate errors on a **completely independent test set**

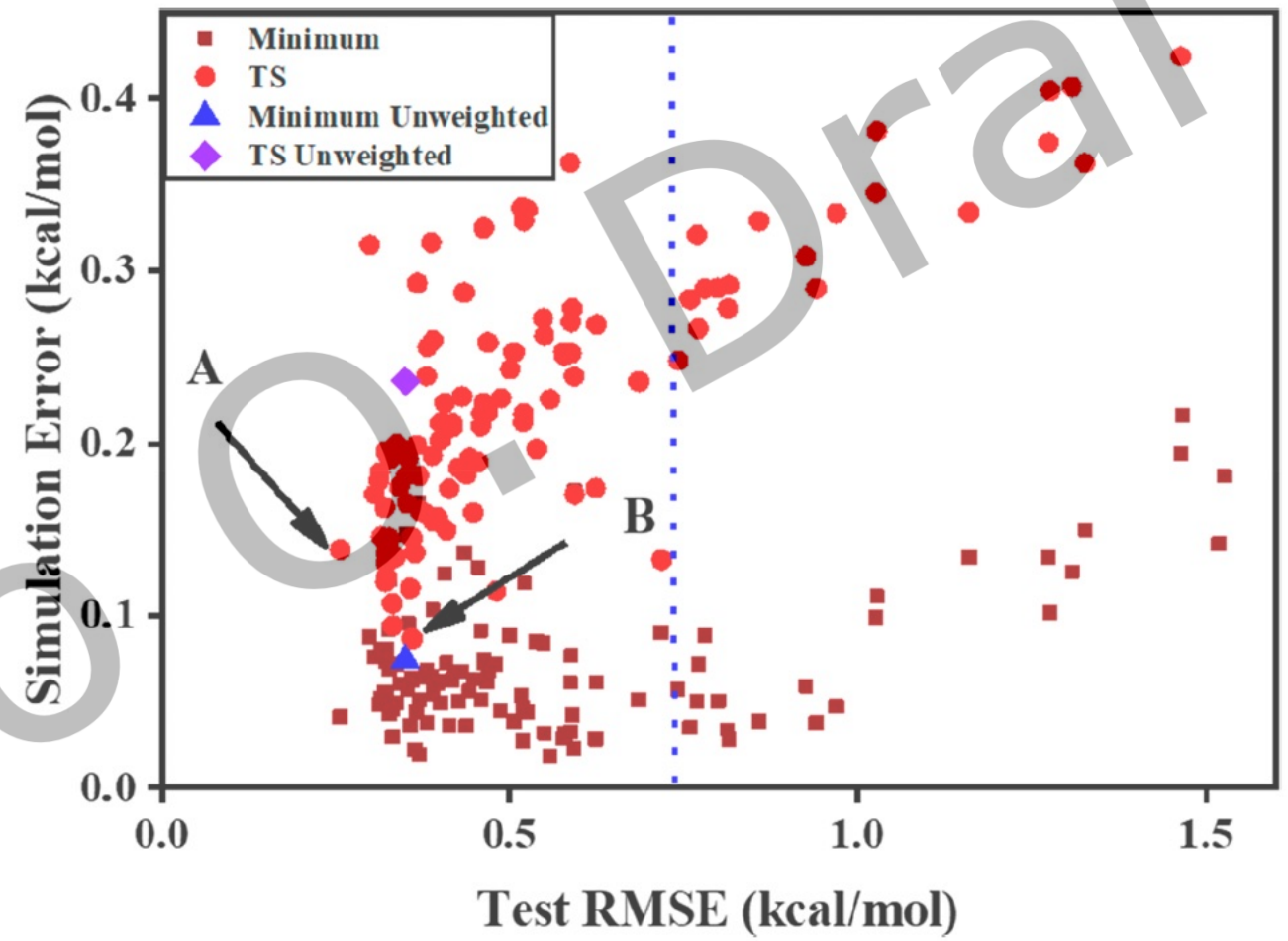


Hastie, Tibshirani, Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed.; Springer-Verlag, **2009**

The ultimate test is the performance in the required application!

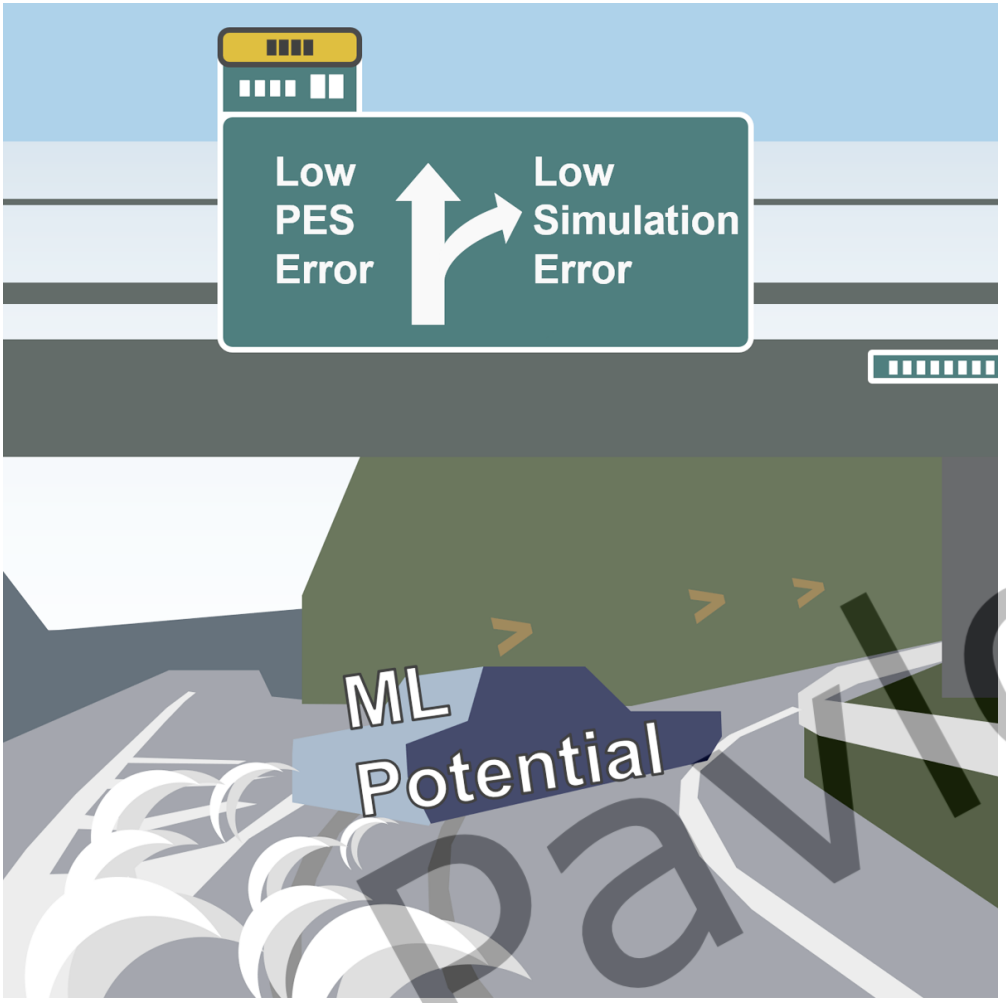
Quality of simulation ≠ quality of fit





F. Ge, R. Wang, C. Qu, P. Zheng, A. Nandi, R. Conte, P. L. Houston, J. M. Bowman, P. O. Dral. *J. Phys. Chem. Lett.* **2024**, *15*, 4451.

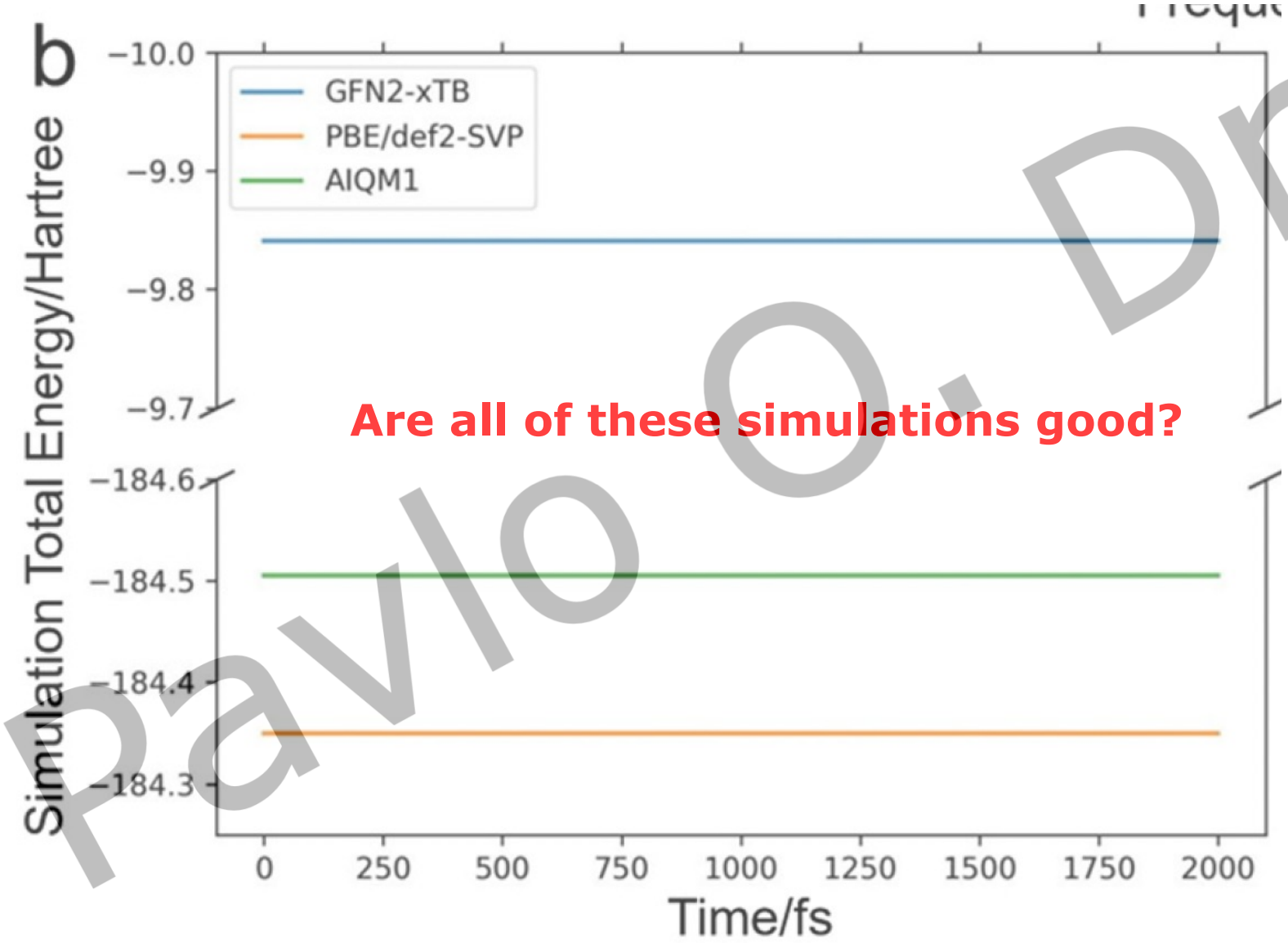
"DMC-certified"



"One [diffusion Monte Carlo] simulation for a single conformer requires 30000 walkers and 55000 steps comprising roughly **$1.6 \cdot 10^9$** potential evaluations with B3LYP."

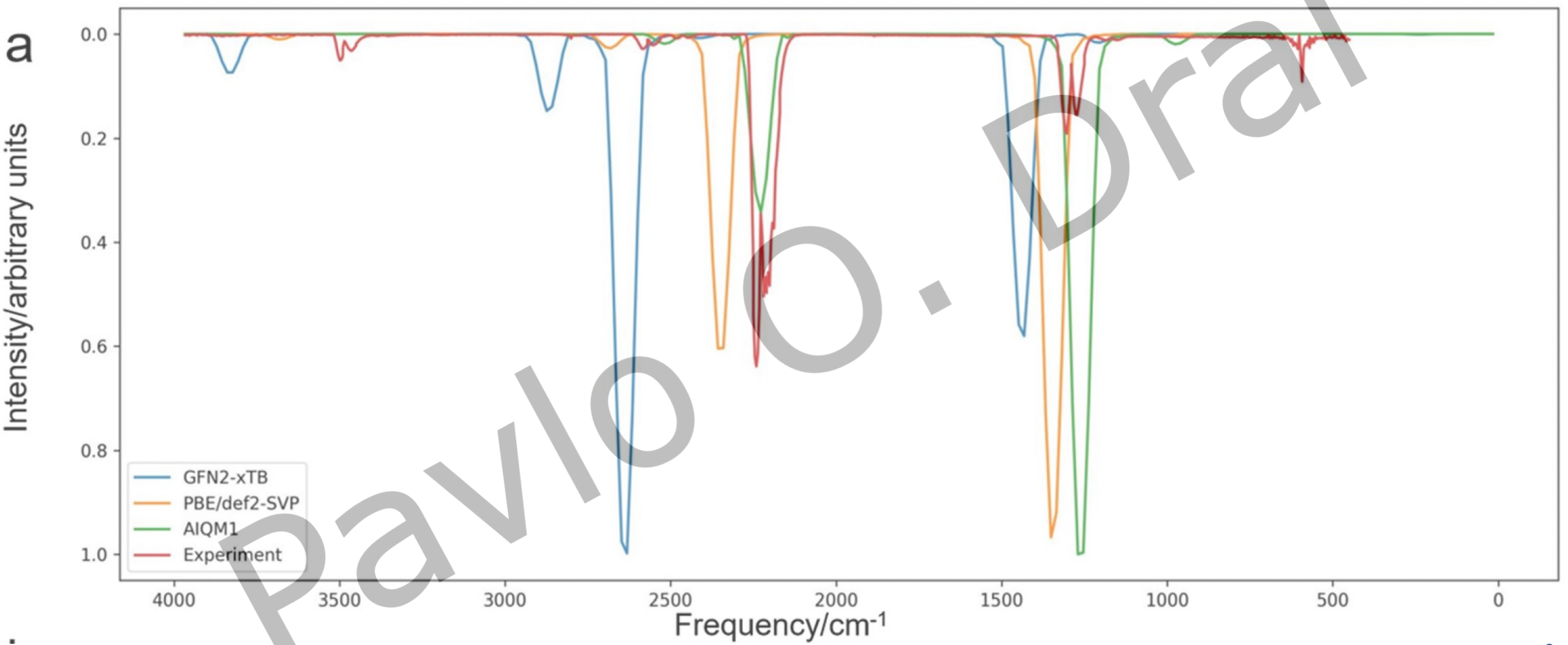
→ 60 hours on a single GPU with ANI…

F. Ge, R. Wang, C. Qu, P. Zheng, A. Nandi, R. Conte, P. L. Houston, J. M. Bowman, P. O. Dral. *J. Phys. Chem. Lett.* **2024**, *15*, 4451.



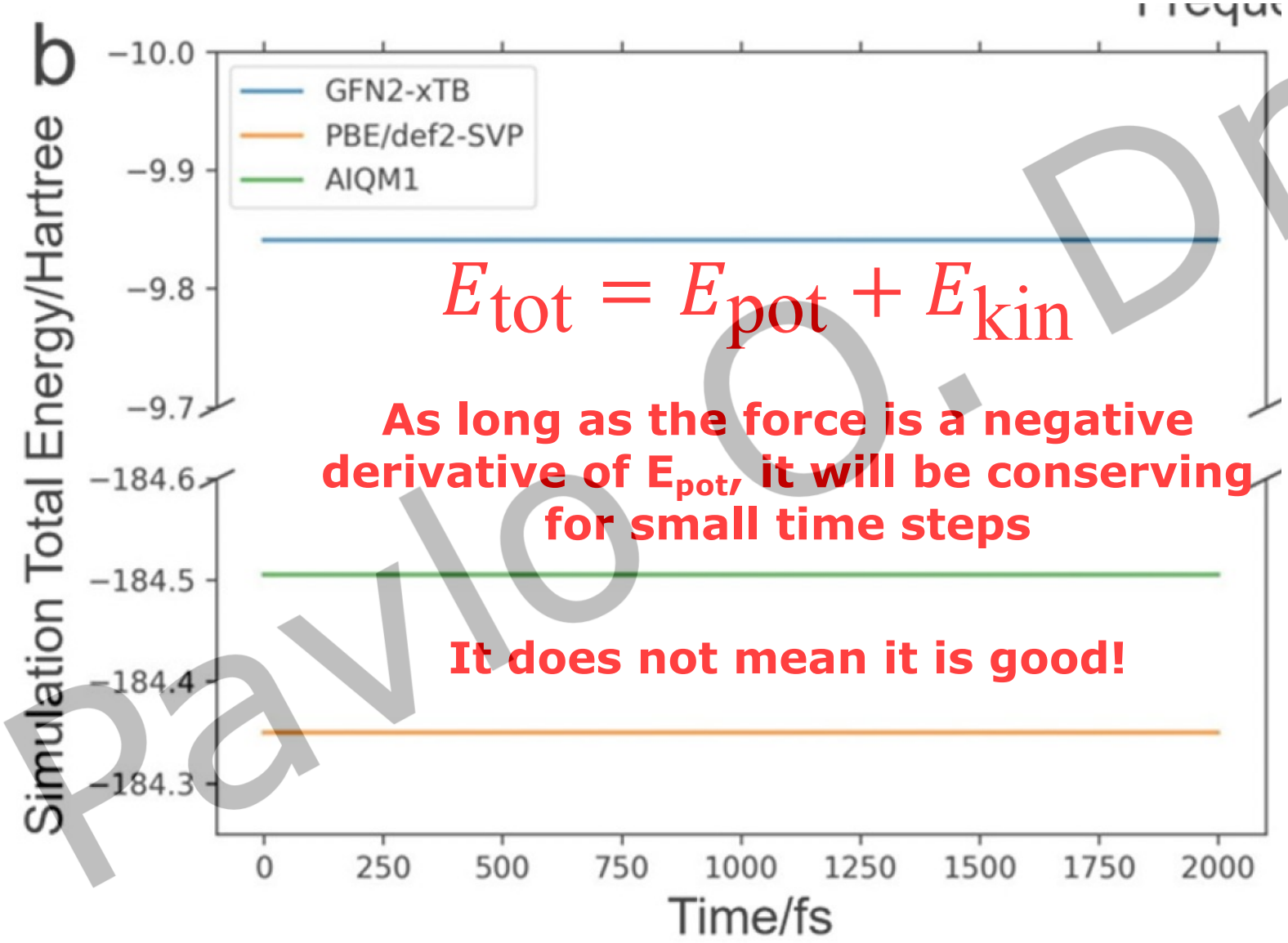
Are all of these simulations good?

L. Zhang & Y. Hou, F. Ge, P. O. Dral, *Phys. Chem. Chem. Phys.* **2023**, *25*, 23467



Why are they so different in accuracy?

L. Zhang & Y. Hou, F. Ge, P. O. Dral, *Phys. Chem. Chem. Phys.* **2023**, *25*, 23467

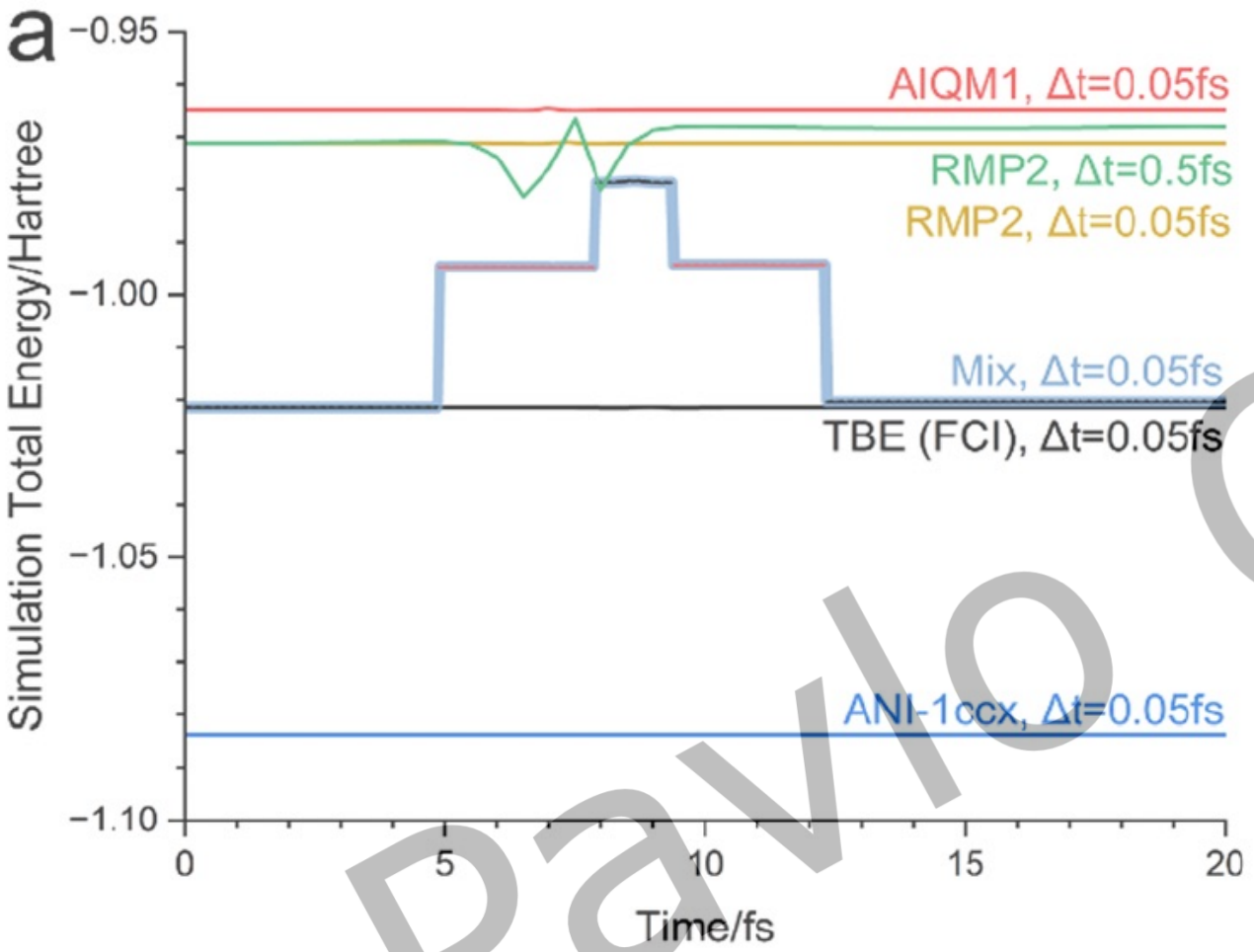


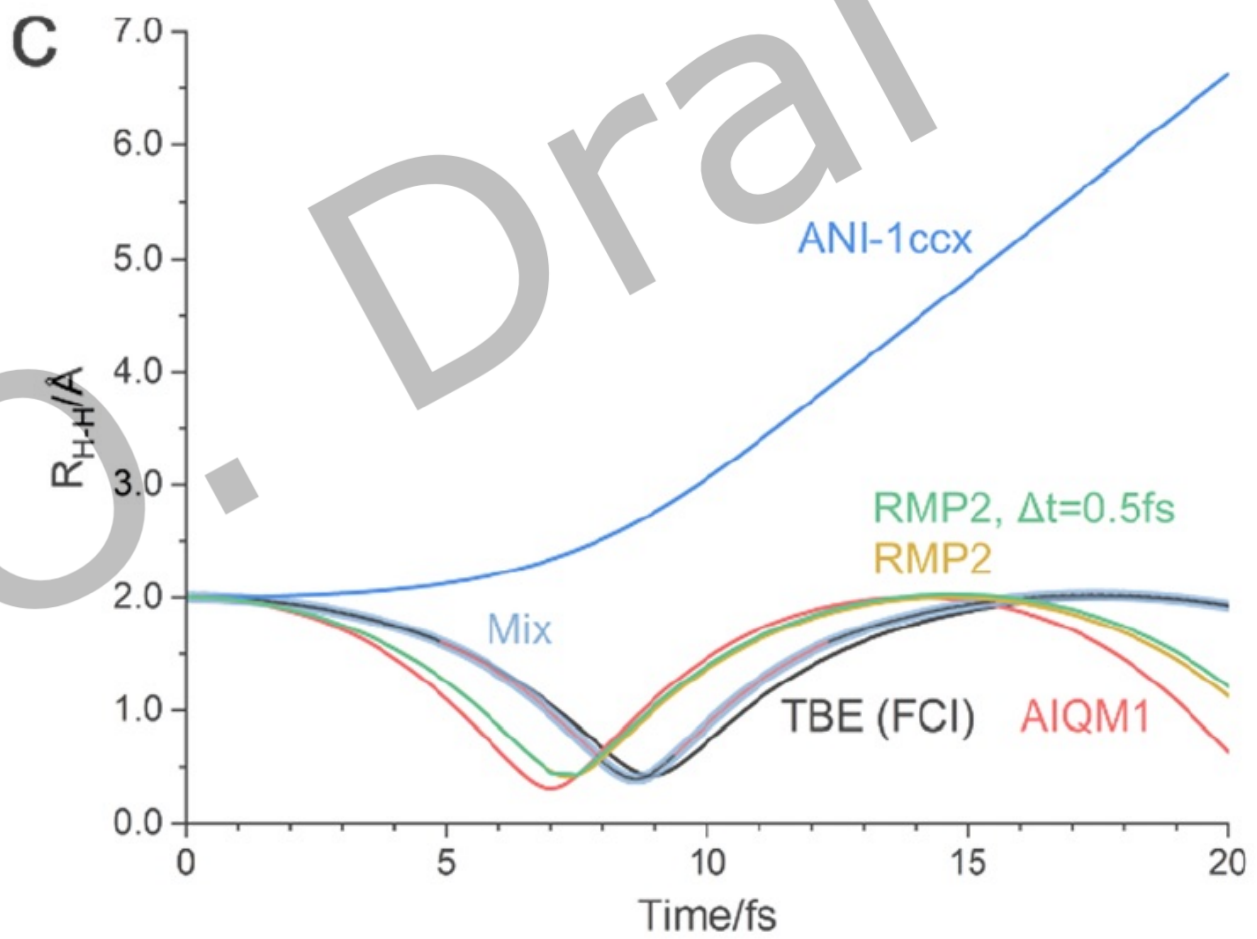
$$E_{\text{tot}} = E_{\text{pot}} + E_{\text{kin}}$$

As long as the force is a negative derivative of E_{pot}, it will be conserving for small time steps

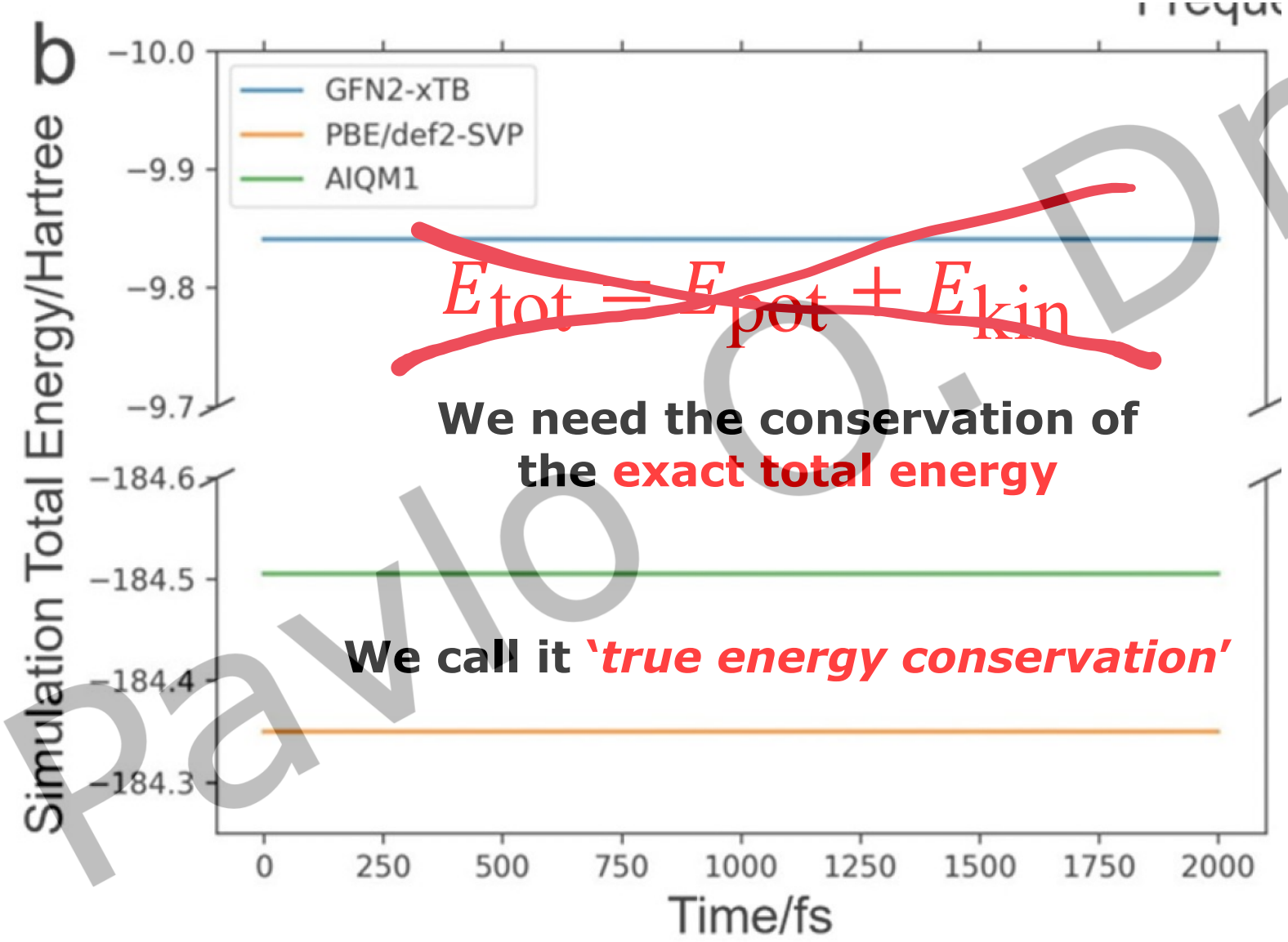
It does not mean it is good!

L. Zhang & Y. Hou, F. Ge, P. O. Dral, *Phys. Chem. Chem. Phys.* **2023**, *25*, 23467





L. Zhang & Y. Hou, F. Ge, P. O. Dral, *Phys. Chem. Chem. Phys.* **2023**, *25*, 23467

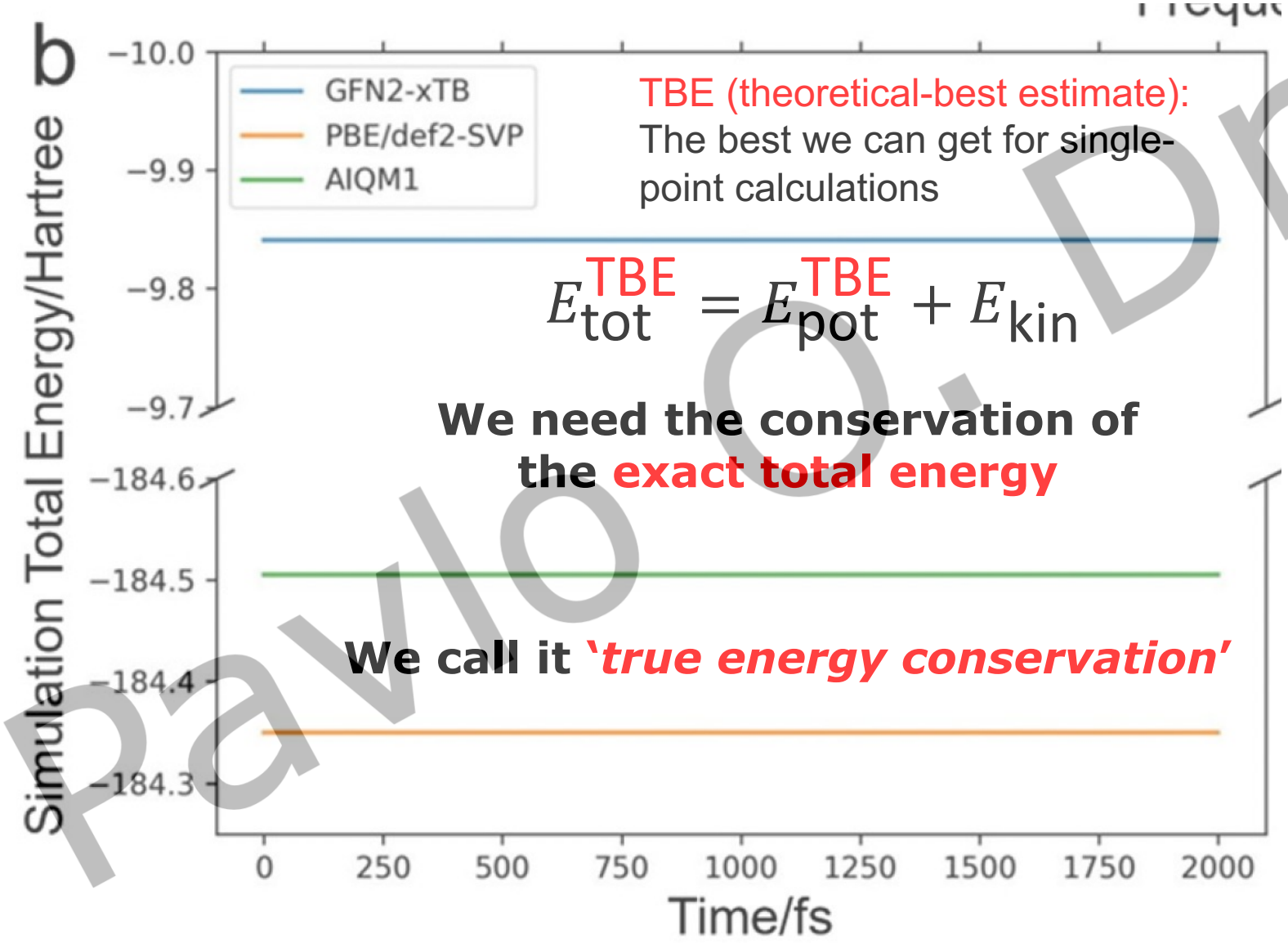


$E_{\mathrm{tot}} = E_{\mathrm{pot}} + E_{\mathrm{kin}}$

We need the conservation of the exact total energy

We call it *'true energy conservation'*

L. Zhang & Y. Hou, F. Ge, P. O. Dral, *Phys. Chem. Chem. Phys.* **2023**, *25*, 23467



TBE (theoretical-best estimate):
The best we can get for single-point calculations

$$E_{\text{tot}}^{\text{TBE}} = E_{\text{pot}}^{\text{TBE}} + E_{\text{kin}}$$

We need the conservation of the exact total energy

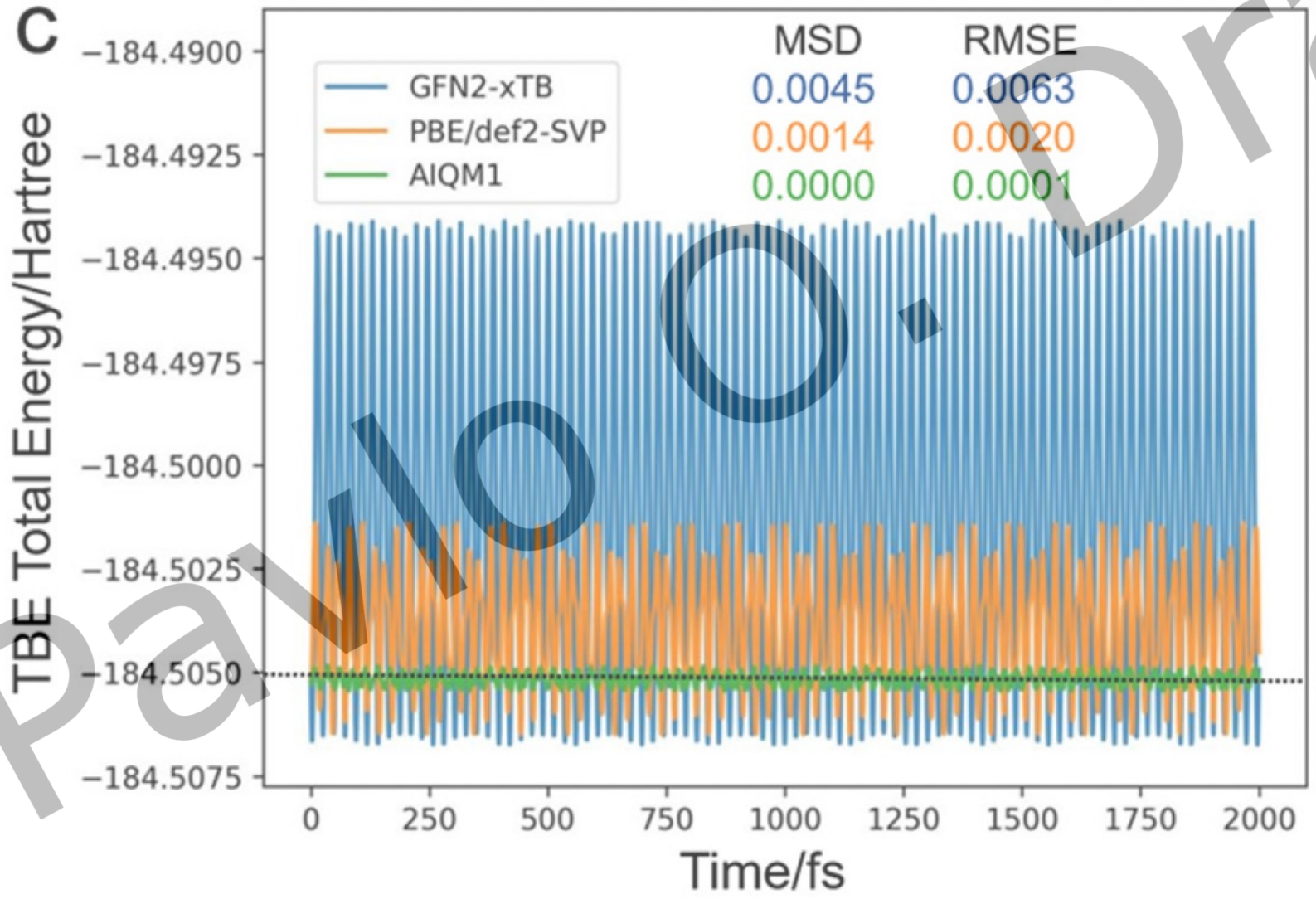
We call it *'true energy conservation'*

L. Zhang & Y. Hou, F. Ge, P. O. Dral, *Phys. Chem. Chem. Phys.* **2023**, *25*, 23467

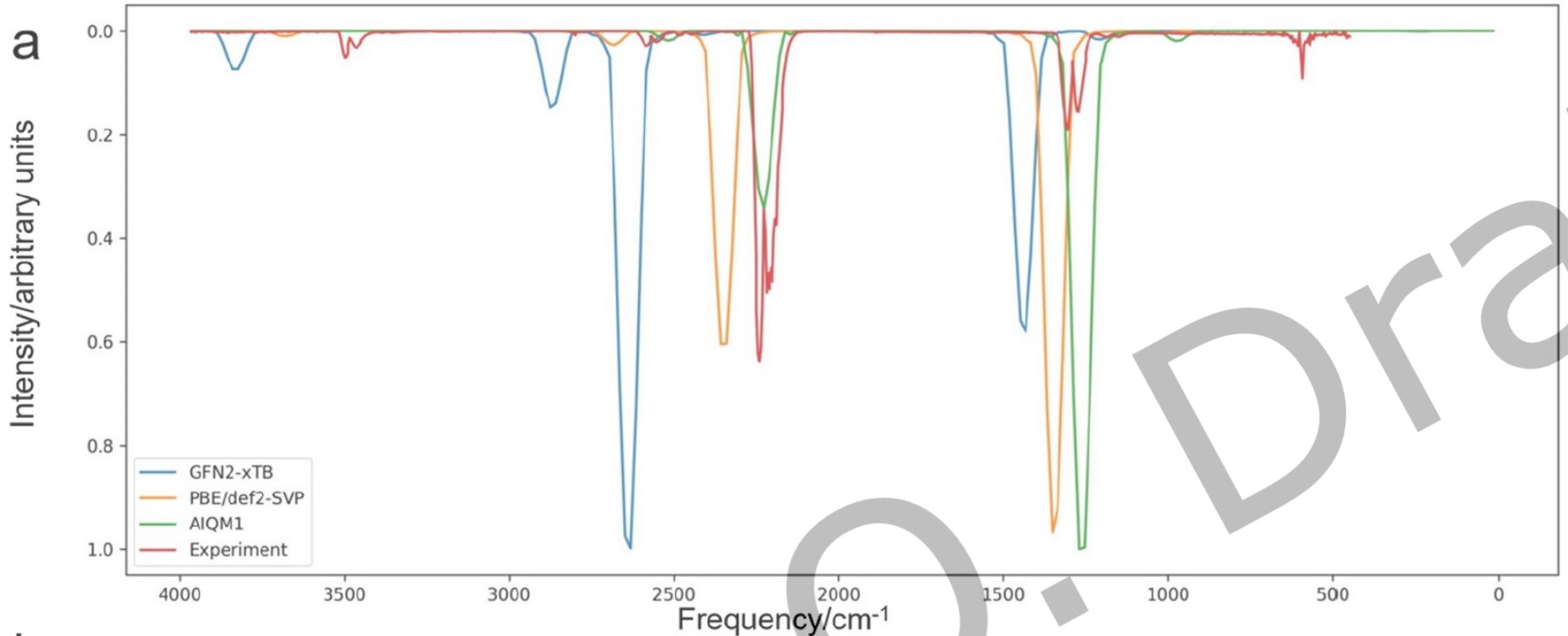
$$\text{RMSE} = \sqrt{1/\text{number of time steps} \sum \left(E_{\text{tot}}^{\text{TBE}}(\text{time step}) - E_{\text{tot}}^{\text{TBE}}(\text{time zero}) \right)^2}$$

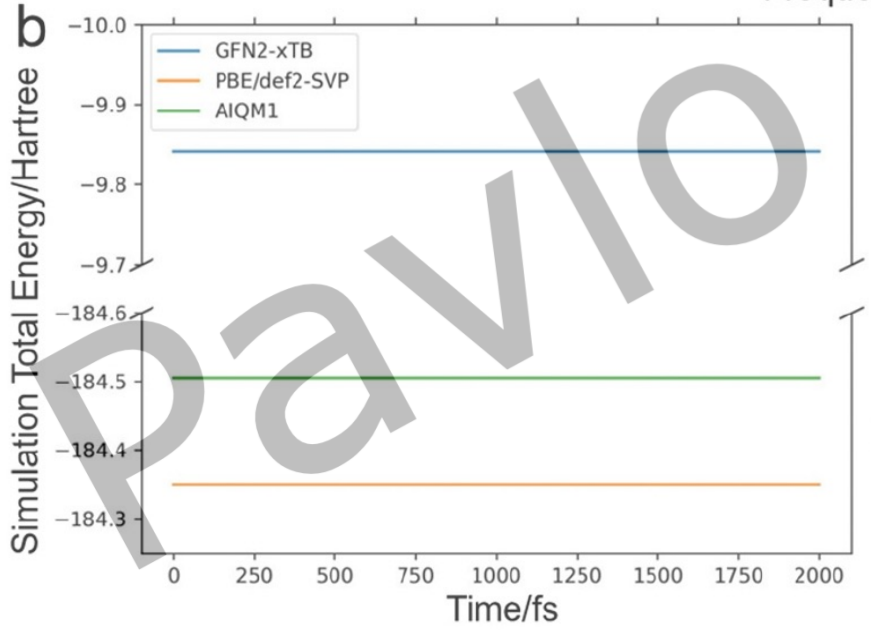
$$\text{MSD} = 1/\text{number of time steps} \sum \left(E_{\text{tot}}^{\text{TBE}}(\text{time step}) - E_{\text{tot}}^{\text{TBE}}(\text{time zero}) \right)$$

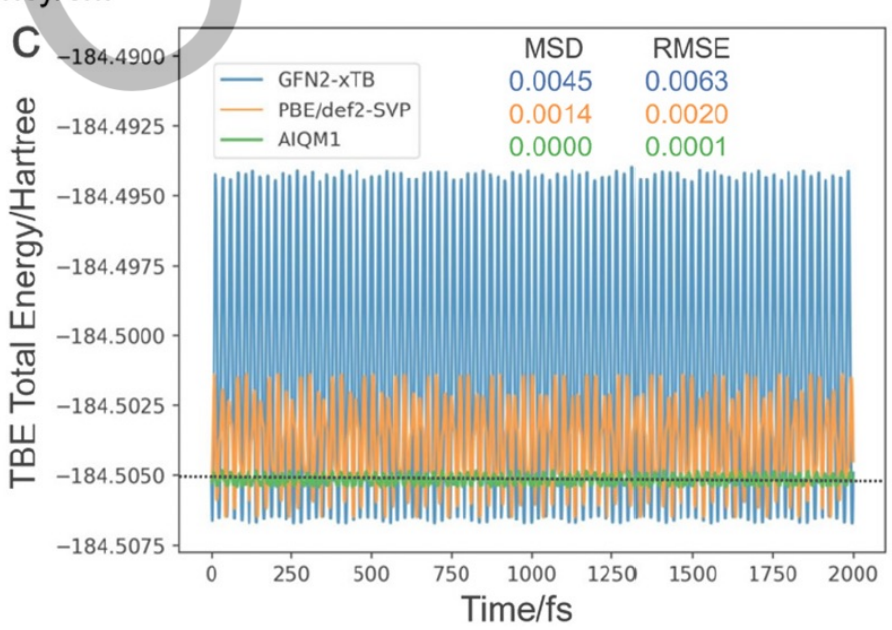
TBE: CCSD(T)*/CBS



L. Zhang & Y. Hou, F. Ge, P. O. Dral, *Phys. Chem. Chem. Phys.* **2023**, *25*, 23467







L. Zhang & Y. Hou, F. Ge, P. O. Dral, *Phys. Chem. Chem. Phys.* **2023**, *25*, 23467

Example PES4.

Estimate the accuracy of the KREG model for the H_2 molecule for the test set.

Questions:

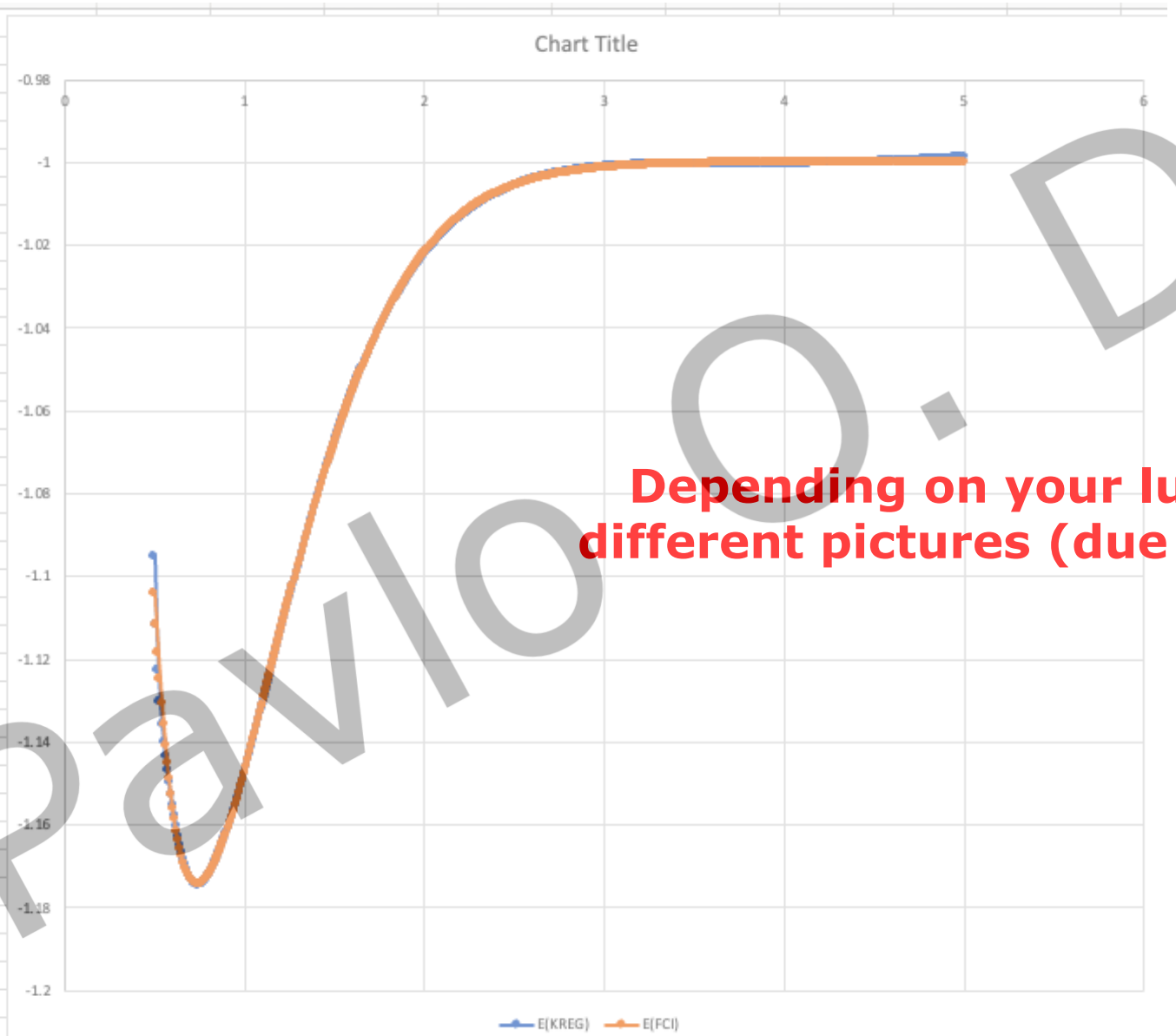
1. What are the training, validation, and test errors (RMSE) of the model?
2. How does the potential energy curve now looks like? (Adapt input file from the previous task)

Input file:

```
estAccMLmodel            # Specify the task for MLatom
MLmodelType=KREG         # Specify the model type
MLmodelOut=kreg_opt.unf # Save model in kreg_opt.unf
XYZfile=h2.xyz           # File with XYZ geometries
Yfile=E_FCI_451.dat      # The file with FCI energies
Ntrain=50
sigma=opt
lgSigmaL=-4              # Lower bound of log2(sigma)
lambda=opt
```

Use the auxiliary files from the previous tasks.

R	E(KREG)	E(FCI)
0.5	-1.0952055	-1.1041324
0.51	-1.1116607	-1.1116602
0.52	-1.1226388	-1.1185659
0.53	-1.1302139	-1.1248928
0.54	-1.1357334	-1.1306807
0.55	-1.1400459	-1.1359664
0.56	-1.1436606	-1.1407837
0.57	-1.1468859	-1.145164
0.58	-1.1498826	-1.1491365
0.59	-1.1527206	-1.1527279
0.6	-1.1554383	-1.1559632
0.61	-1.1580248	-1.1588655
0.62	-1.1604676	-1.1614566
0.63	-1.1627427	-1.1637562
0.64	-1.1648372	-1.1657832
0.65	-1.1667376	-1.1675548
0.66	-1.1684246	-1.1690873
0.67	-1.1698978	-1.1703957
0.68	-1.1711619	-1.1714941
0.69	-1.1722143	-1.1723958
0.7	-1.1730593	-1.1731131
0.71	-1.1737061	-1.1736575
0.72	-1.1741656	-1.1740399
0.73	-1.1744488	-1.1742703
0.74	-1.1745676	-1.1743585
0.75	-1.1745317	-1.1743132
0.76	-1.1743587	-1.1741428
0.77	-1.1740494	-1.1738553
0.78	-1.1736193	-1.173458
0.79	-1.1730809	-1.1729579
0.8	-1.1724471	-1.1723616
0.81	-1.1717182	-1.1716753
0.82	-1.1708958	-1.1709047
0.83	-1.1700126	-1.1700554
0.84	-1.1690501	-1.1691325
0.85	-1.1680262	-1.1681409



Depending on your luck, you may obtain different pictures (due to random sampling)

Example PES5.

Check how the test error changes when you include energy gradients during the training of an MLP. Note also, how much time you need to train with and without energy gradients.

You might need to repeat calculations couple of times and might want to change the number of training points.

Input file:

```
estAccMLmodel           # Specify the task for MLatom
MLmodelType=KREG        # Specify the model type
MLmodelOut=kreg_grad.unf # Save model in kreg_opt.unf
XYZfile=H2.xyz           # file with geometies
Yfile=H2_HF.en           # file with energies
YgradXYZfile=H2_HF.grad  # file with energy gradients
Ntrain=20
Nsubtrain=0.9           # request 90% of the training points to be used for validation
sigma=opt
lgSigmaL=-4             # Lower bound of log2(sigma)
lambda=opt
lgLambdaL=-8             # Lower bound of log2(lambda)
```

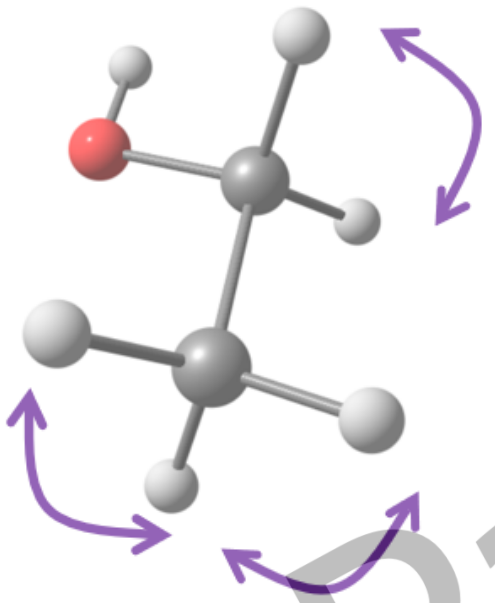
The provided data is generated at HF/STO-3G and you can download it as a zipped folder. It contains:

- H2.xyz - xyz coordinates
- H2_HF.en - energies at HF/STO-3G
- H2_HF.grad - energy gradients at HF/STO-3G

KREG & pKREG

(p)KREG

$$\begin{pmatrix} \mathbf{K}+\lambda_{\mathrm{v}} & \dfrac{\partial \mathbf{K}}{\partial \mathbf{M}} \\ \dfrac{\partial \mathbf{K}}{\partial \mathbf{M}} & \dfrac{\partial^2 \mathbf{K}}{\partial \mathbf{M}^2}+\lambda_{\mathrm{gxyz}} \end{pmatrix} \boldsymbol{\alpha} = \begin{pmatrix} \boldsymbol{y}-\boldsymbol{y}_{\mathrm{prior}} \\ \dfrac{\partial \boldsymbol{y}}{\partial \mathbf{M}} \end{pmatrix}$$



Permutations

RE descriptor

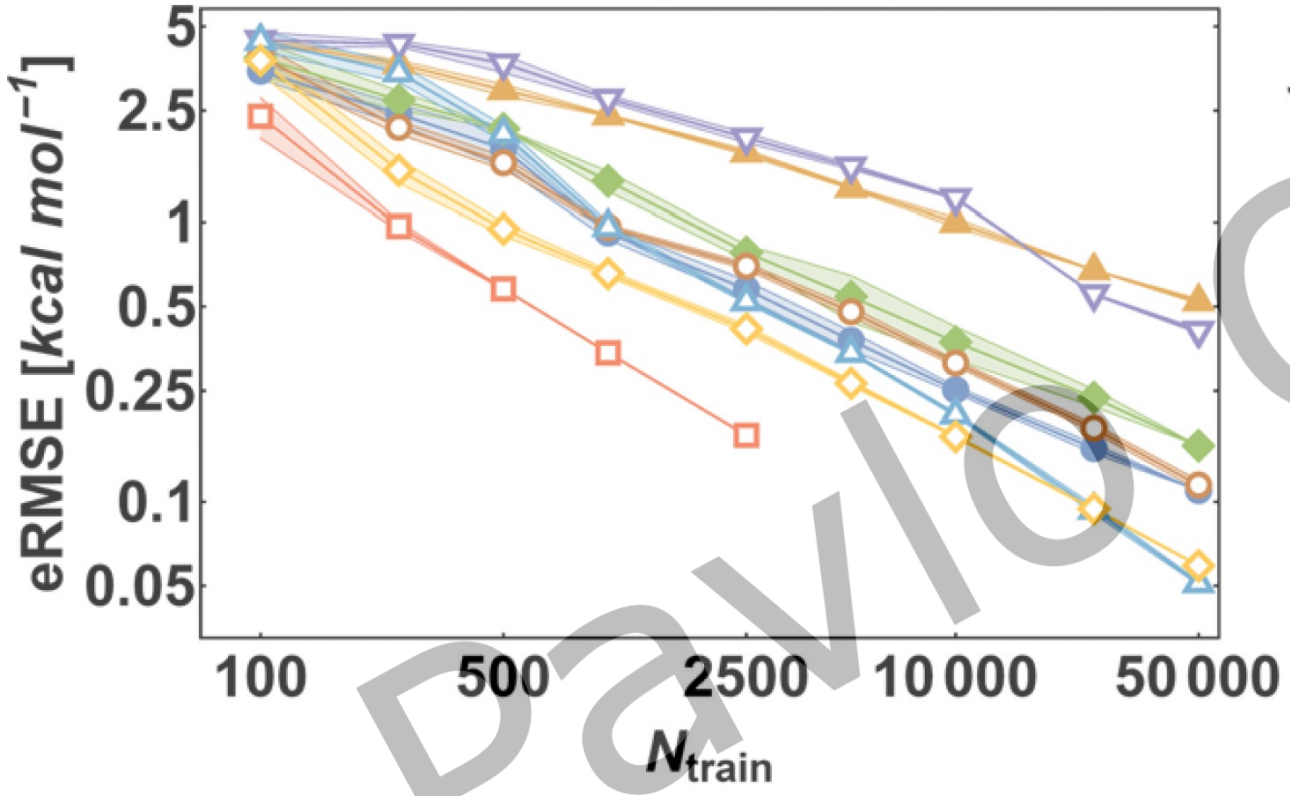
$$\mathbf{x}^{\mathrm{T}} = \left[\cdots \quad r^{\mathrm{ref}}_{a,b\neq a}/r_{a,b\neq a} \quad \cdots \right]$$

Gaussian kernel

$$k(\mathbf{x}_i,\mathbf{x}_j) = \exp\left(-\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2/2\sigma^2\right)$$

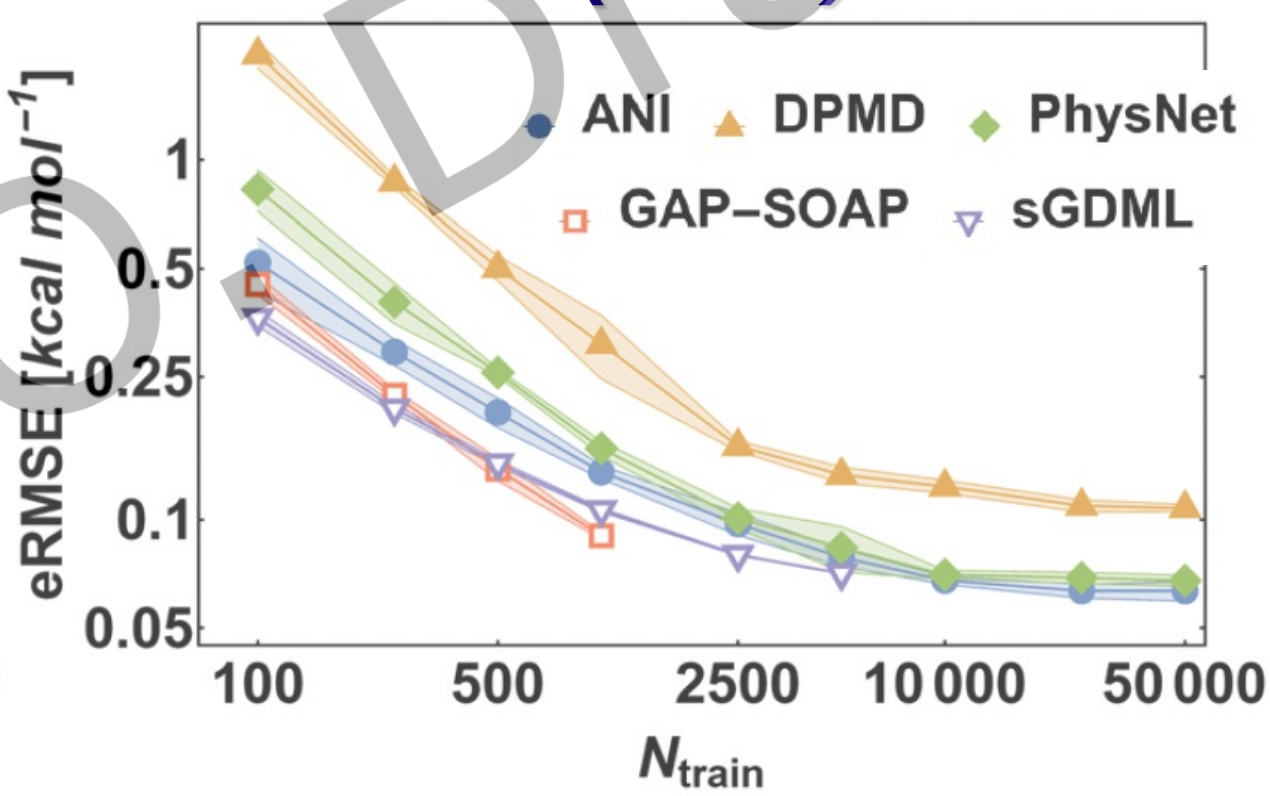
Energies-only

Ethanol (MD17)



Energies+gradients

Ethanol (MD17)



M. Pinheiro Jr, F. Ge, N. Ferré, P. O. Dral, M. Barbatti. *Chem. Sci.* **2021**, *12*, 14396–14413

Example PES6.

Estimate the accuracy of the KREG model for the H_2 molecule for the test set by using the 9:1 splitting of the training set for hyperparameter optimization, i.e., using 10 of the training data for the validation set. To see how stability of the results change compared to 8:2 splitting, you can run this experiment several times and compare the test error and the potential energy curves.

Input file:

```
estAccMLmodel           # Specify the task for MLatom
MLmodelType=KREG        # Specify the model type
MLmodelOut=kreg_opt.unf # Save model in kreg_opt.unf
XYZfile=h2.xyz          # File with XYZ geometries
Yfile=E_FCI_451.dat     # The file with FCI energies
Ntrain=50
Nsubtrain=0.9           # request 90% of the training points to be used for validation
sigma=opt
lgSigmaL=-4             # Lower bound of log2(sigma)
lambda=opt
```

Use the auxiliary files from the previous tasks.

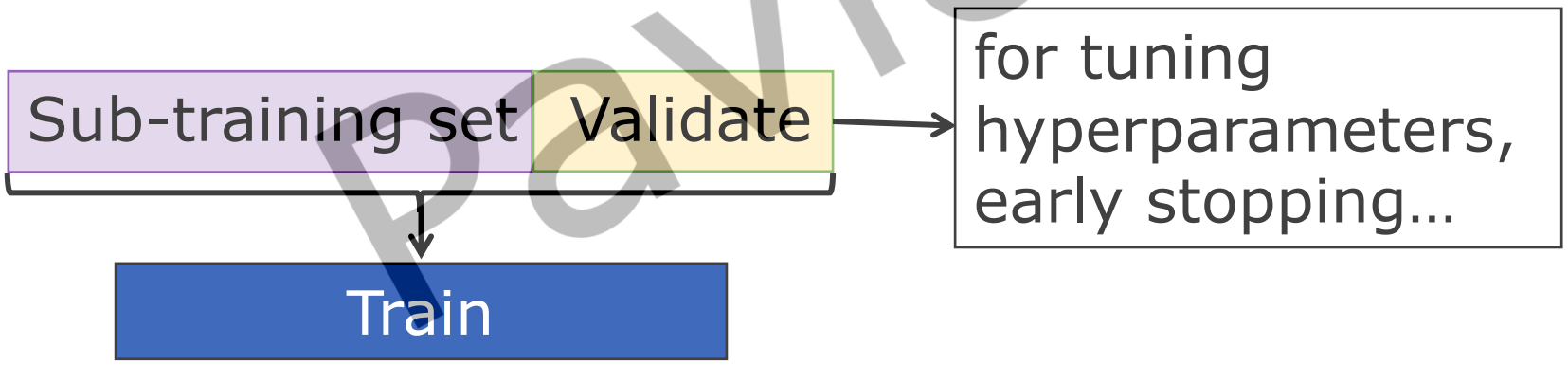
KRR-CM vs KREG

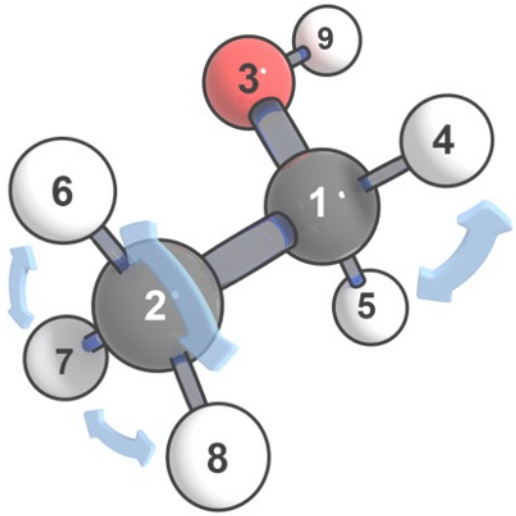
KRR-CM – kernel ridge regression with Gaussian kernel and Coulomb matrix descriptor

Test set RMSEs in kcal/mol

Number of training points	KRR-CM	KREG
100	3.90±0.41	4.45±0.36
2500	0.70±0.02	0.52±0.01

What do we mean by '100 training points?' Is the validation set included?
Let's use the term **'sub-training set'**!





M. Pinheiro Jr, F. Ge, N. Ferré, P. O. Dral, M. Barbatti. *Chem. Sci.* **2021**, *12*, 14396–14413

Example PES7.

Investigate how the performance of the KREG model changes for different number of the training points. Modify the `Ntrain` parameter in the input file below. Particularly for smaller number of training points, there is a wide spread of results, so they should be repeated more times.

Input file:

```
estAccMLmodel           # Specify the task for MLatom
MLmodelType=KREG        # Specify the model type
MLmodelOut=kreg_opt.unf # Save model in kreg_opt.unf
XYZfile=h2.xyz          # File with XYZ geometries
Yfile=E_FCI_451.dat     # The file with FCI energies
Ntrain=50
Nsubtrain=0.9           # request 90% of the training points to be used for validation
sigma=opt
lgSigmaL=-4             # Lower bound of log2(sigma)
lambda=opt
```

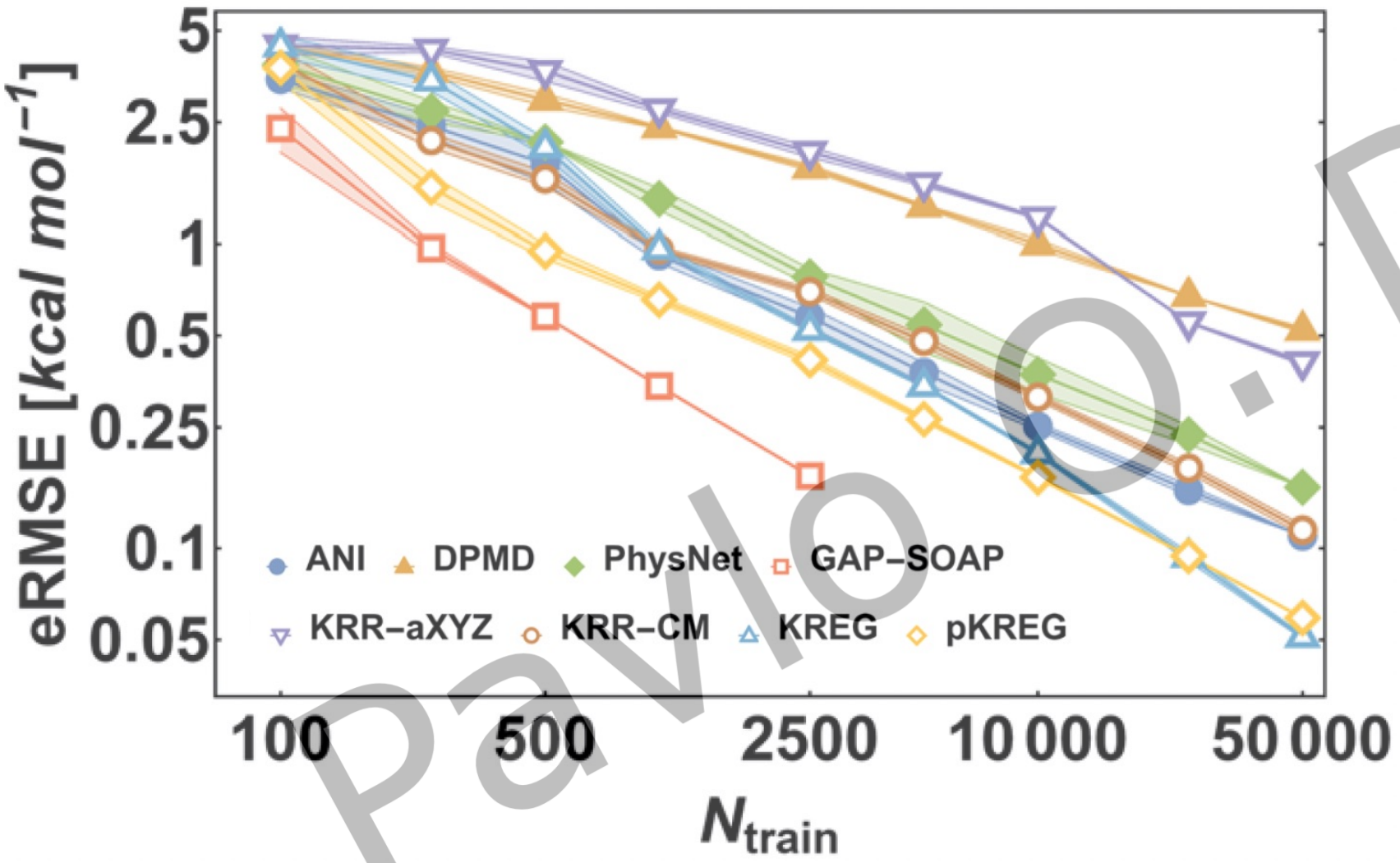
MLatom has an automatic way of generating the dependence of the test error on the size of the training set. The input file can look like:

```
learningCurve
lcNtrains=10,20,50,100,200,300
lcNrepeats=5,4,3,2,1,1
Ntest=100
MLmodelType=KREG        # Specify the model type
MLmodelOut=kreg_opt.unf # Save model in kreg_opt.unf
XYZfile=h2.xyz          # File with XYZ geometries
Yfile=E_FCI_451.dat     # The file with FCI energies
Nsubtrain=0.9
sigma=opt
lgSigmaL=-4             # Lower bound of log2(sigma)
lambda=opt
```

The results of these calculations will be saved in `learningCurve/mlatomf_KREG_en` folder, where the file `lcy.csv` collates the information which you can, e.g., plot in Excel.

Use the auxiliary files from the previous tasks.

Learning curves (energies-only)



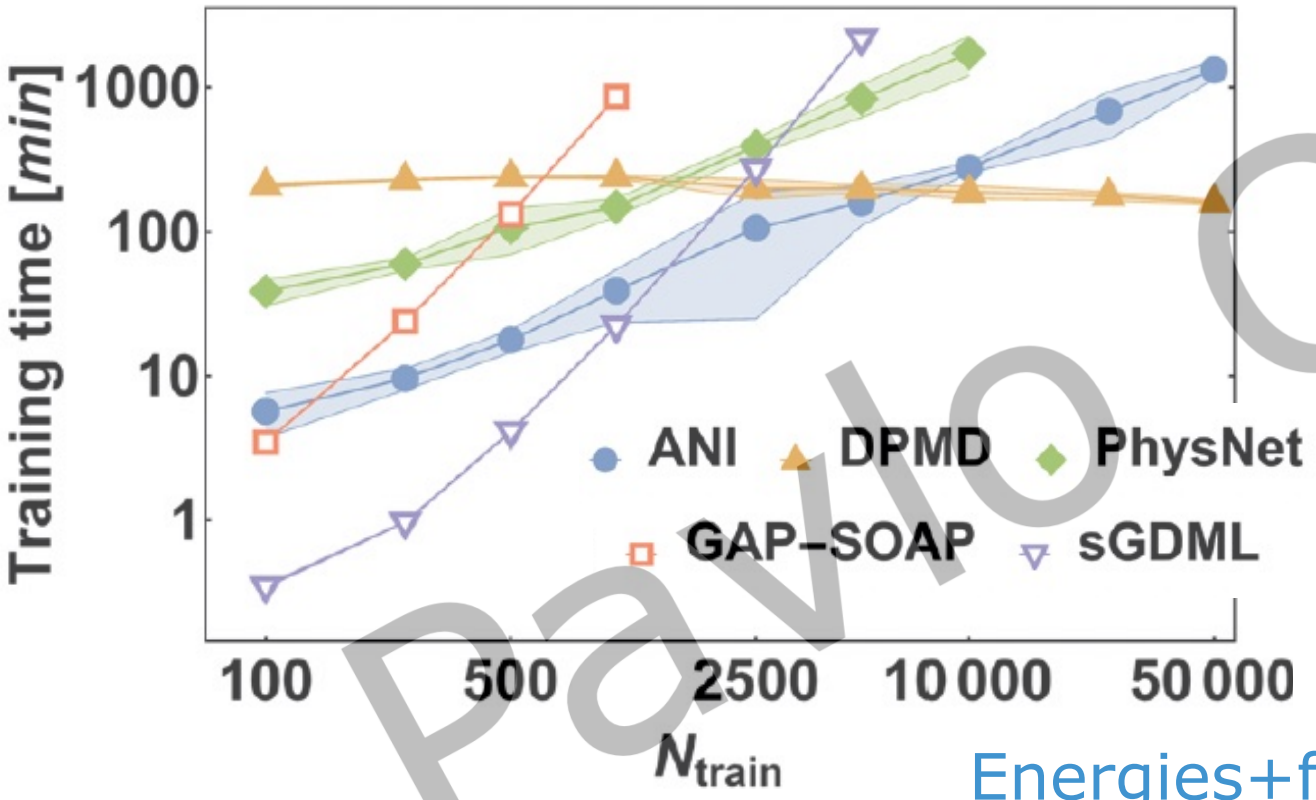
$$\varepsilon = \varepsilon_{\mathrm{a}} + \frac{a}{N_{\mathrm{tr}}^{b}}$$

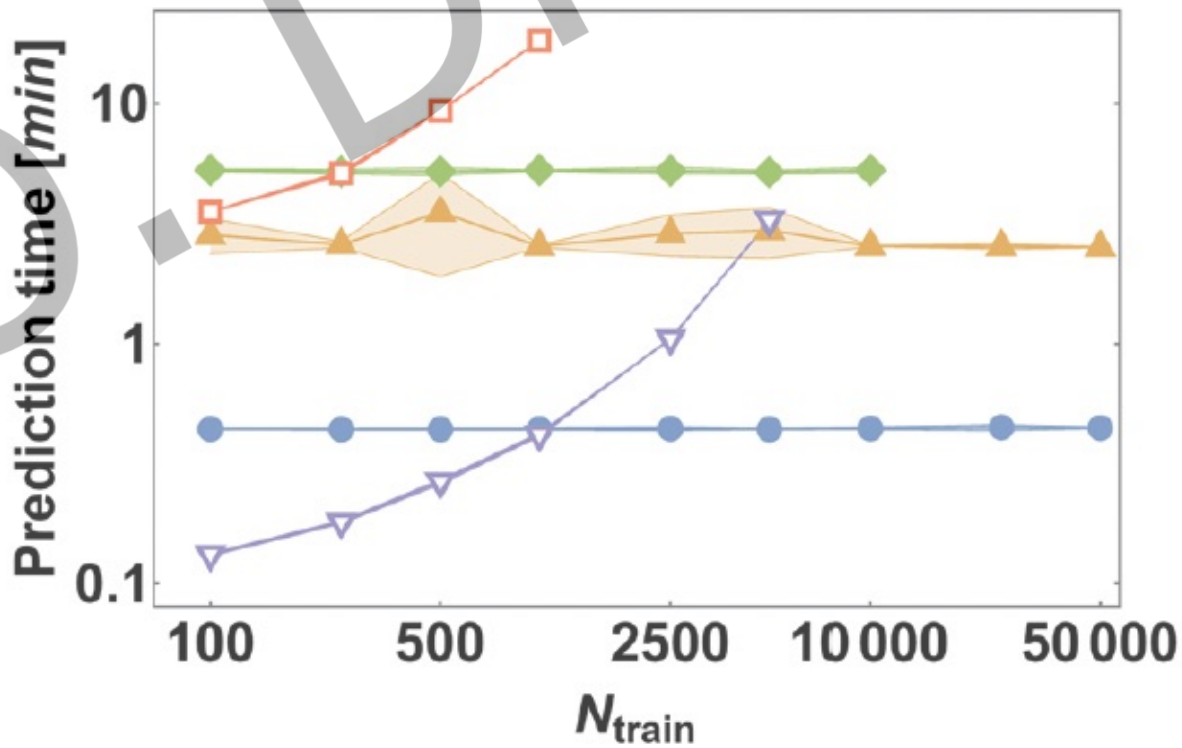
$$\log(\varepsilon) \approx \log(a) - b \log N_{\mathrm{tr}}$$

M. Pinheiro Jr, F. Ge, N. Ferré, P. O. Dral, M. Barbatti. *Chem. Sci.* **2021**, *12*, 14396–14413
P. O. Dral, F. Ge, B.-X. Xue, Y.-F. Hou, M. Pinheiro Jr, J. Huang, M. Barbatti. *Top. Curr. Chem.* **2021,** *379*, 27

Timings!

For small training sets kernel methods (open markers) are often both more accurate and faster for training and prediction than neural networks (filled markers)





Energies+forces

The same hardware!

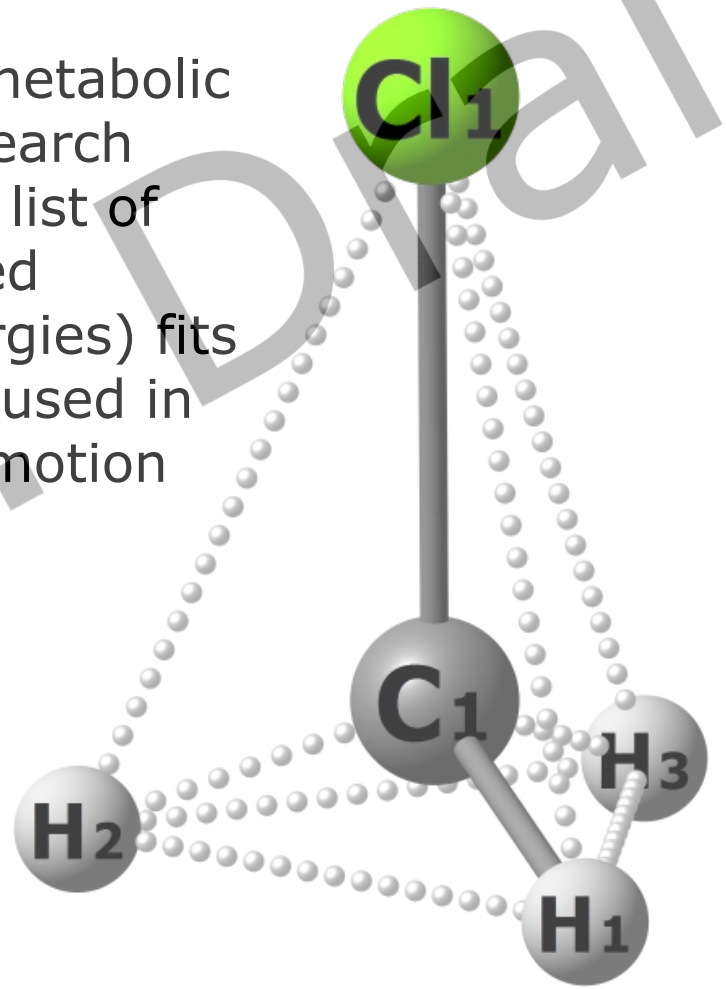
M. Pinheiro Jr, F. Ge, N. Ferré, P. O. Dral, M. Barbatti. *Chem. Sci.* **2021**, *12*, 14396–14413

Spectroscopically accurate PES of CH_3Cl

- Biomarker produced from a secondary metabolic process: useful for extraterrestrial life search
- Highly accurate and comprehensive line list of the rotation-vibration spectrum is needed
- **TROVE** (Theoretical ROVibrational Energies) fits analytic representation of PES, which is used in variational approach to solving nuclear motion problem

$$\left(\sum_{i=1}^{N}\frac{-\hbar^2}{2M_i}\nabla_i^2+V\right)\Psi=E\Psi$$

Fundamental term values are reproduced by TROVE with RMSE of **0.75** cm^{-1}

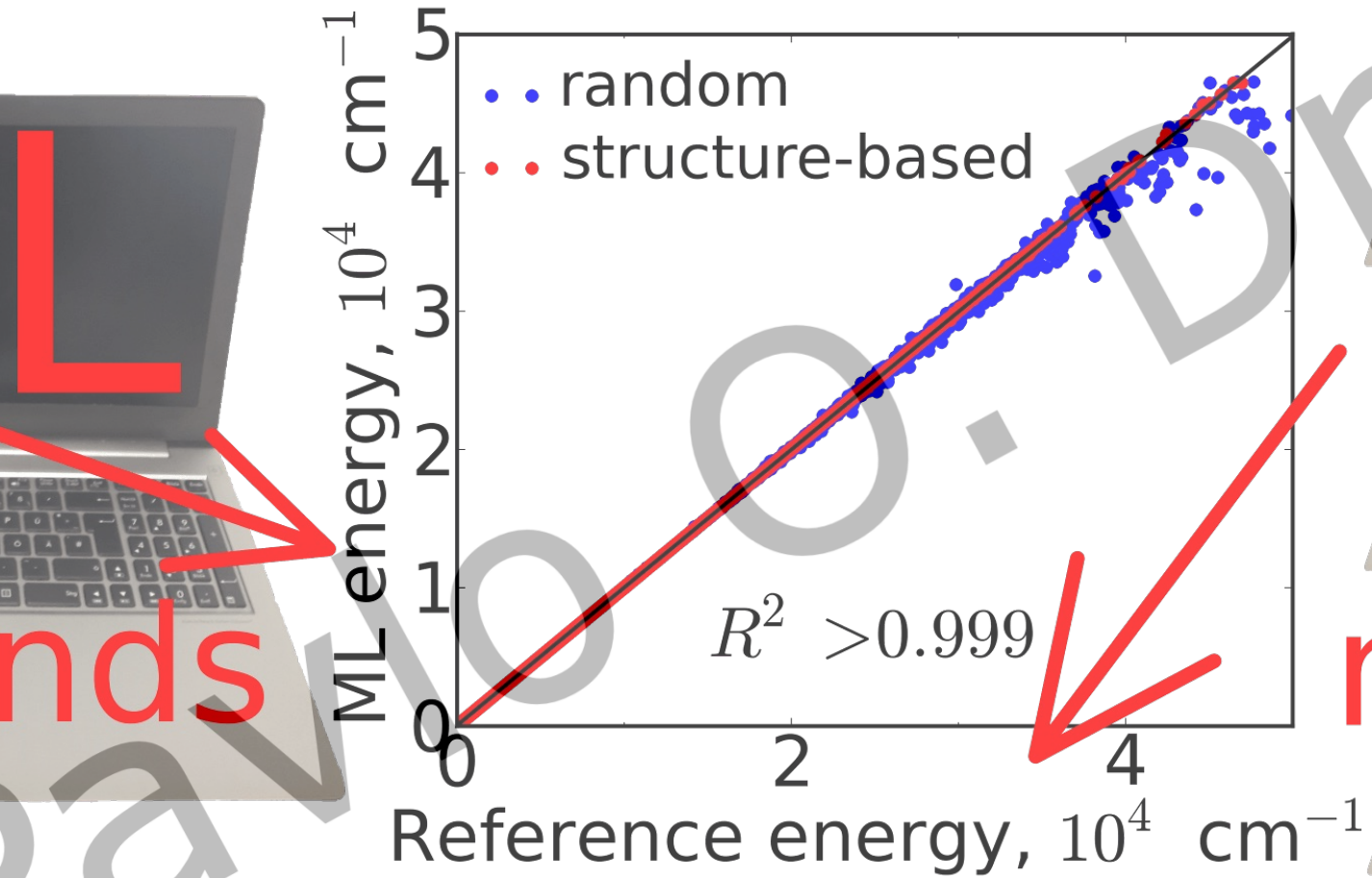


A. Owens, S. N. Yurchenko, A. Yachmenev, J. Tennyson, W. Thiel, *J. Chem. Phys.* **2015**, *142*, 244306
TROVE: S. N. Yurchenko, W. Thiel, P. Jensen, *J. Mol. Spectrosc.* **2007**, *245*, 126

Motivation

ML

seconds



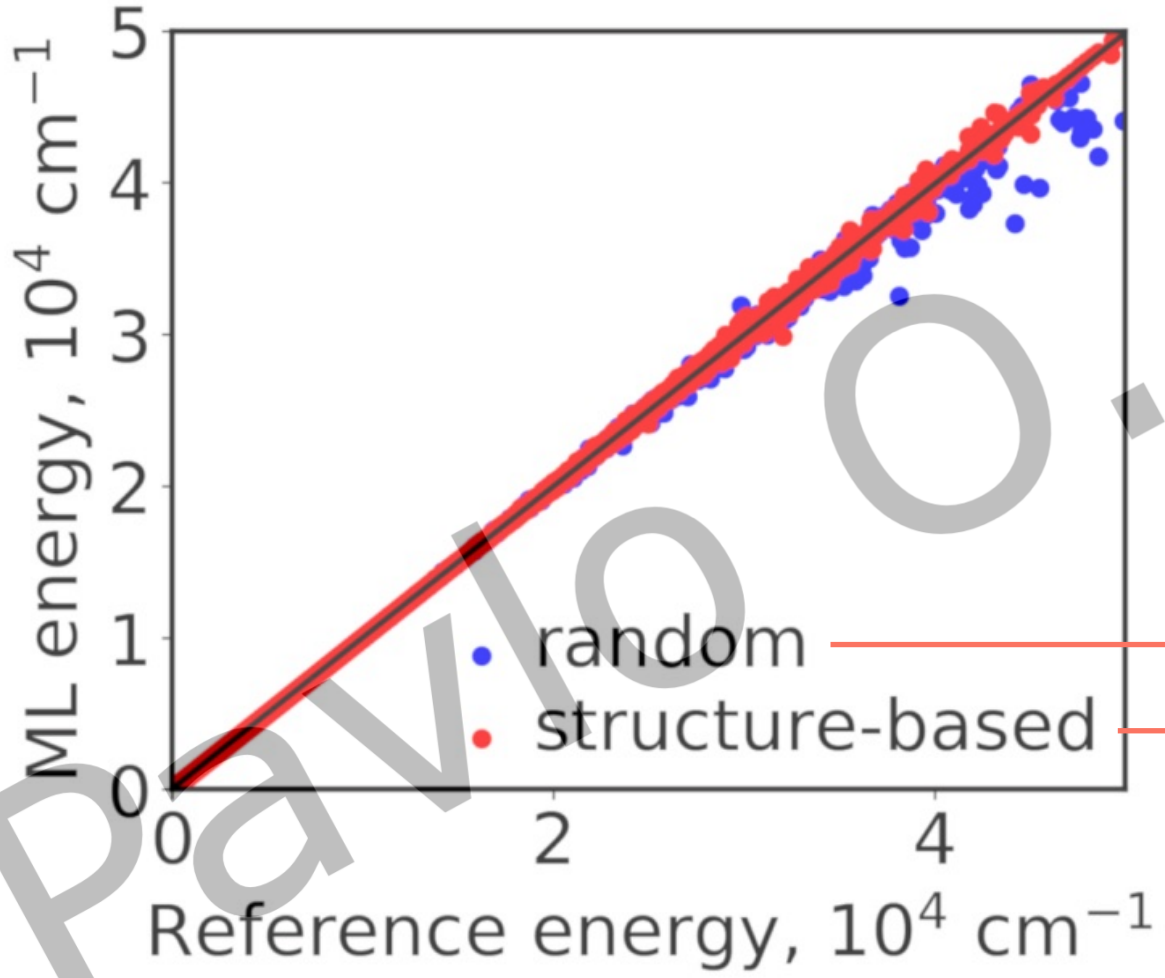
QM

months

P. O. Dral, A. Owens, S. Yurchenko, W. Thiel, *J. Chem. Phys.* **2017**, *146*, 244108

ML Model Performance

Figure from: P. O. Dral, *J. Phys. Chem. Lett.* **2020**, *11*, 2336



Online tutorial:
MLatom.com/AQCtutorial/

Training set

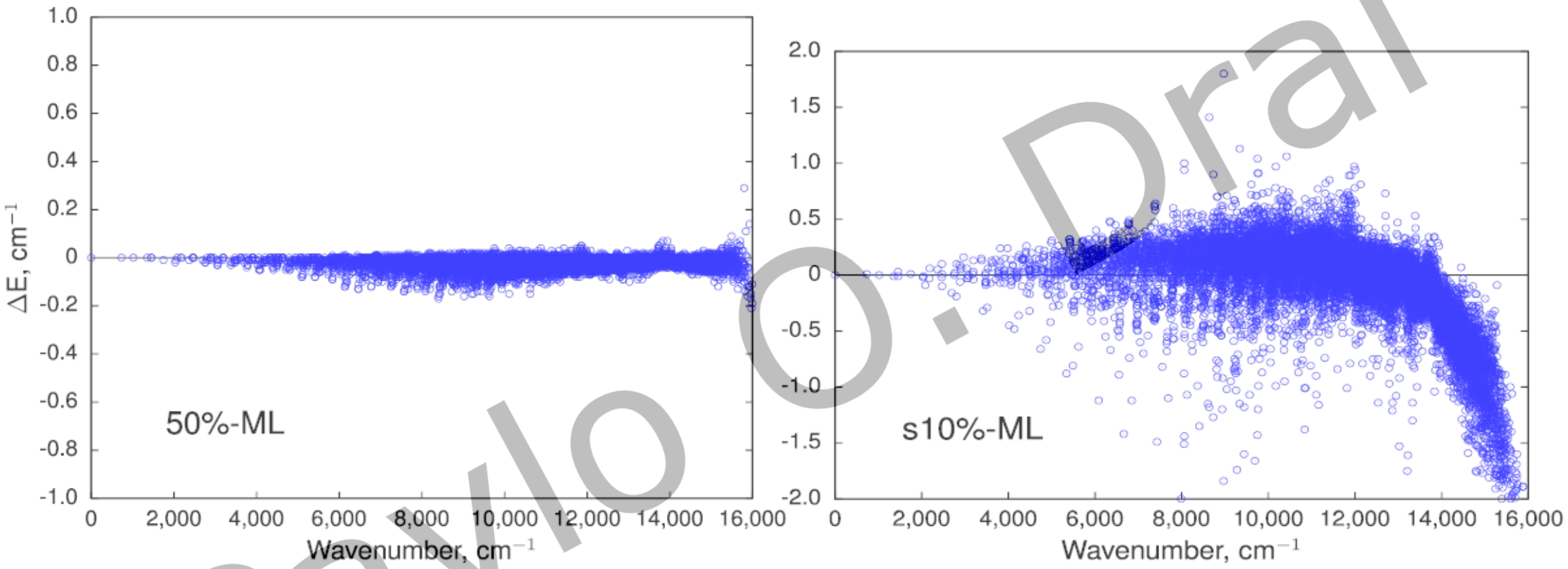
50% of QC data

10% of QC data:
saved 90% of CPU-time!

Weighted RMSE:
3.49 cm^{-1} = 0.01 kcal/mol

P. O. Dral, A. Owens, S. Yurchenko, W. Thiel, *J. Chem. Phys.* **2017**, *146*, 244108

Accuracy of Vibrational Spectra



No of levels	Energy, cm^{-1}	RMSE, cm^{-1}
166	5,000	0.02
3,606	10,000	0.04

No of levels	Energy, cm^{-1}	RMSE, cm^{-1}
166	5,000	0.14
3,606	10,000	0.28

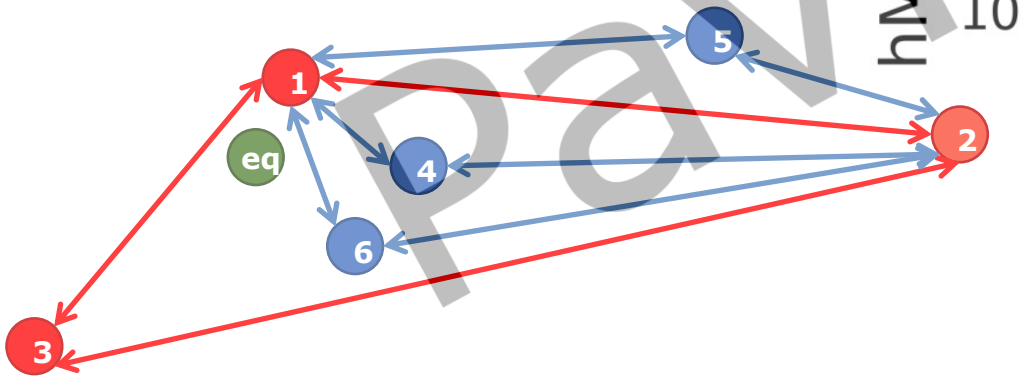
P. O. Dral, A. Owens, S. Yurchenko, W. Thiel, *J. Chem. Phys.* **2017**, *146*, 244108

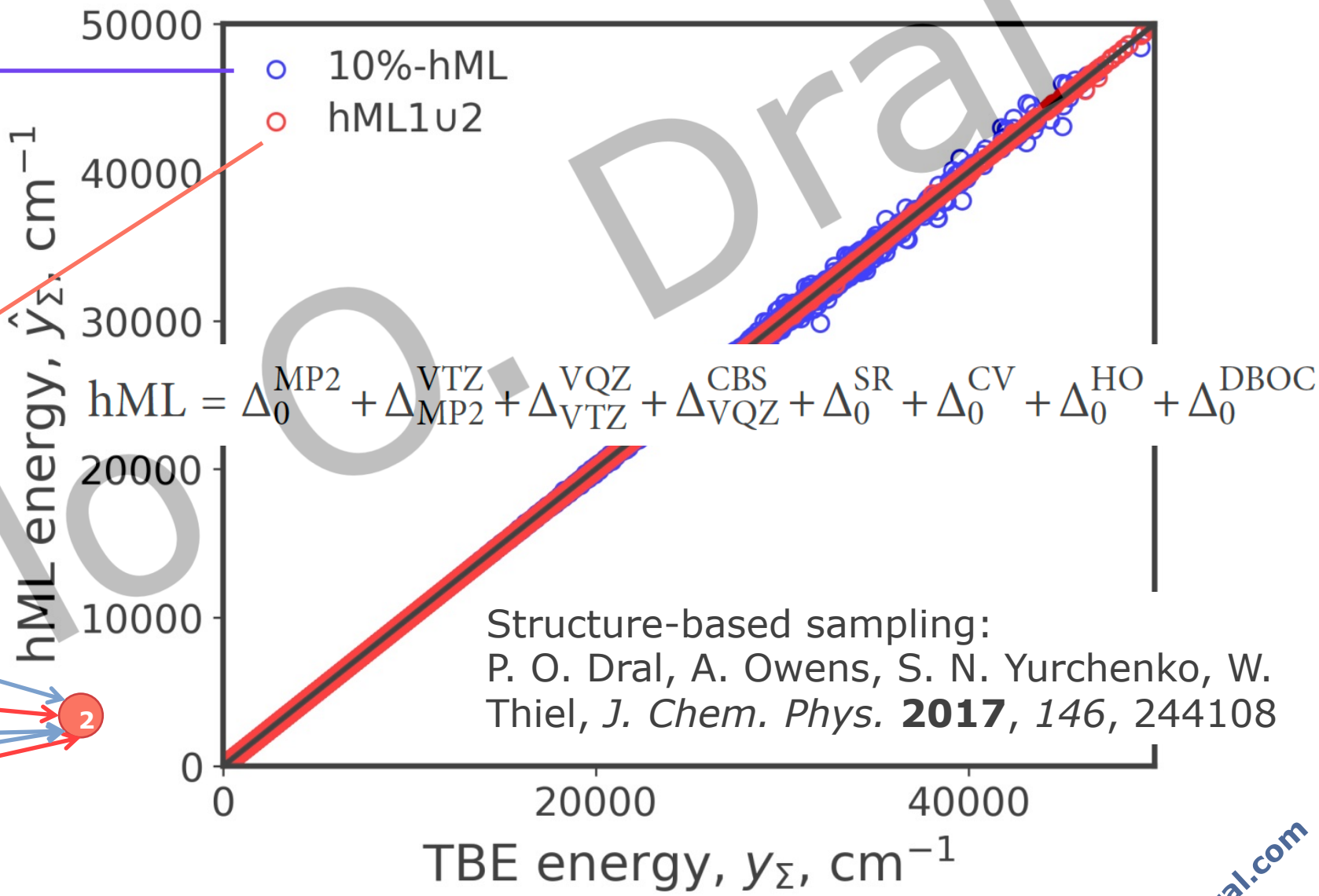
Hierarchical ML for CH_3Cl PES

Pure ML-model trained on **saved 90% of CPU-time**

Weighted RMSE:
3.49 cm⁻¹ = 0.01 kcal/mol

Hierarchical ML models:
saved 99% of CPU-time
Weighted RMSE:
1.12 cm⁻¹ = 0.003 kcal/mol





$$\mathrm{hML} = \Delta_0^{\mathrm{MP2}} + \Delta_{\mathrm{MP2}}^{\mathrm{VTZ}} + \Delta_{\mathrm{VTZ}}^{\mathrm{VQZ}} + \Delta_{\mathrm{VQZ}}^{\mathrm{CBS}} + \Delta_0^{\mathrm{SR}} + \Delta_0^{\mathrm{CV}} + \Delta_0^{\mathrm{HO}} + \Delta_0^{\mathrm{DBOC}}$$

Structure-based sampling:
P. O. Dral, A. Owens, S. N. Yurchenko, W. Thiel, *J. Chem. Phys.* **2017**, *146*, 244108

P. O. Dral, A. Owens, A. Dral, G. Csányi, *J. Chem. Phys.* **2020**, *152*, 204110