XACS
Xiamen Atomistic Computing Suite
XACScloud.com

MLatom.com

Pavlo O. Dral

Xiamen University, P.R. China

Visiting Professor in
Nicolaus Copernicus University, Poland

Lecture: ML potentials I

10 September 2024

Tutorial's Jupyter Notebooks

Please check the link
[and register on XACS cloud – optional]

before lunch break

Questionary

https://github.com/JakubMartinka/karlsruhe2024

Pavlo O. Dral

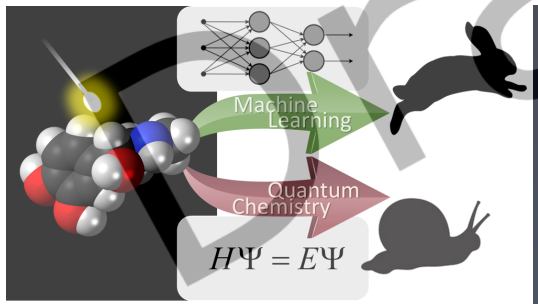
AI in computational chemistry

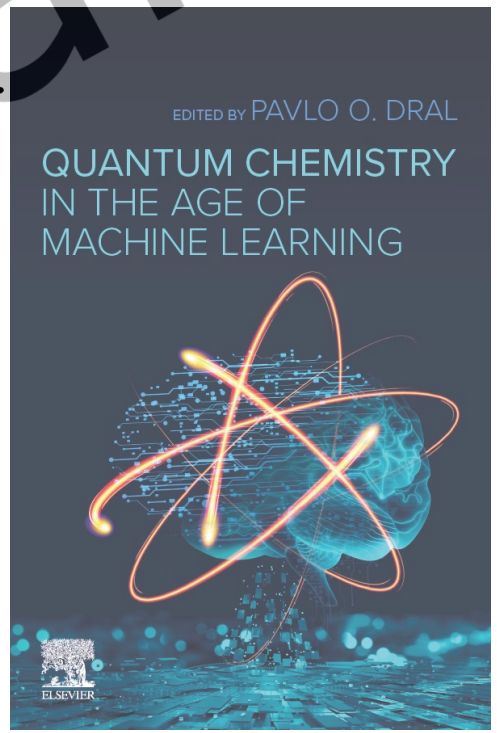
Professor | Outstanding Youth (Overseas)

Email: dral@xmu.edu.cn

Research Areas: artificial intelligence, quantum chemistry, dynamics, excited states, semi-empirical methods

MLatom.com





2024-Present: Nicolaus Copernicus University, Visiting Professor

2021-Present: Xiamen University, Full Professor

2019-2021: Xiamen University, Associate Professor

2013-2019: Max-Planck-Institut für Kohlenforschung, Postdoc

2010-2013: University of Erlangen-Nürnberg, M.Sc. & Ph.D.

2008-2010: University of Erlangen-Nürnberg, M.Sc.

2004-2010: National Technical University of Ukraine "KPI", B.Sc. & M.Sc.

XACS
Xiamen Atomistic Computing Suite

nature communications
CHEMISTRY AND MATERIALS SCIENCES
TOP 25
READ ARTICLES OF 2021
OFFICIAL AUTHOR

Research Interests:

Our research transforms chemical physics simulations by developing novel AI methods and providing software and cloud computing platforms.

✓Selected papers:

AI platform: J. Chem. Theory Comput. 2024, 20, 1193
AI-quantum dynamics: Nat. Commun. 2022, 13, 1930
AI-quantum mechanics: Nat. Commun. 2021, 12, 7022
AI-excited states: Nat. Rev. Chem. 2021, 5, 388
AI force fields: Chem. Sci. 2021, 12: 14396

Group website: dr-dral.com

XACS
Xiamen Atomistic Computing Suite
XACScloud.com

Looking for talented group members!

MLatom.com

PCOSS

国家自然科学基金委员会
National Natural Science Foundation of China

嘉庚创新实验室
TAN KAH KEE INNOVATION LABORATORY

2023

2024

Looking for talented group members!

XACS
Xiamen Atomistic Computing Suite
XACScloud.com

MLatom.com

PCOSS

国家自然科学基金委员会
National Natural Science Foundation of China

嘉庚创新实验室
TAN KAH KEE INNOVATION LABORATORY

2023 – in Warsaw

2024 – in Xiamen

ML potentials reshaping quantum chemistry

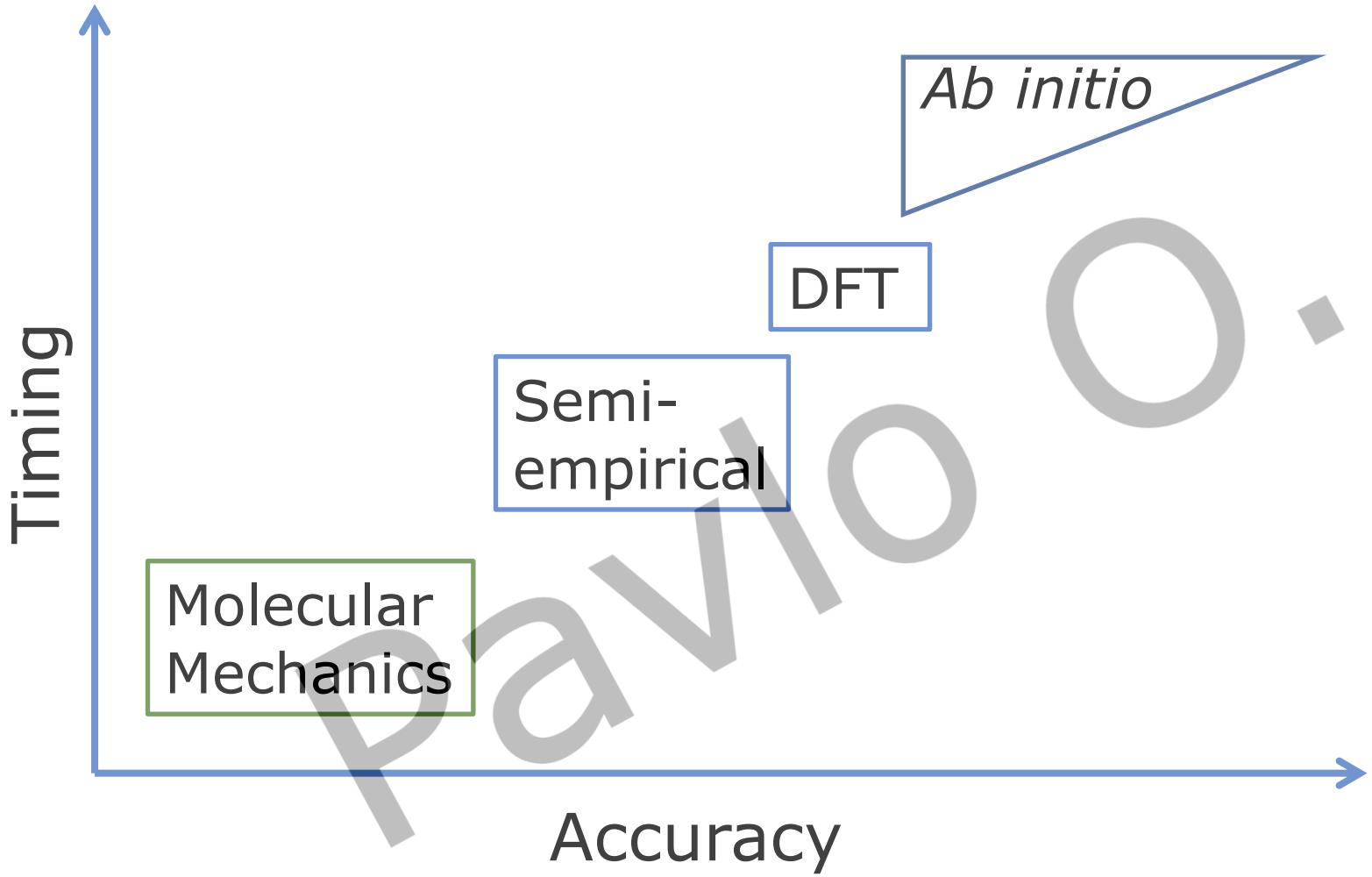


Figure: P. O. Dral, *J. Phys. Chem. Lett.* **2020**, *11*, 2336

ML potentials reshaping quantum chemistry

P. O. Dral, M. Barbatti, *Nat. Rev. Chem.* **2021,** *5*, 388

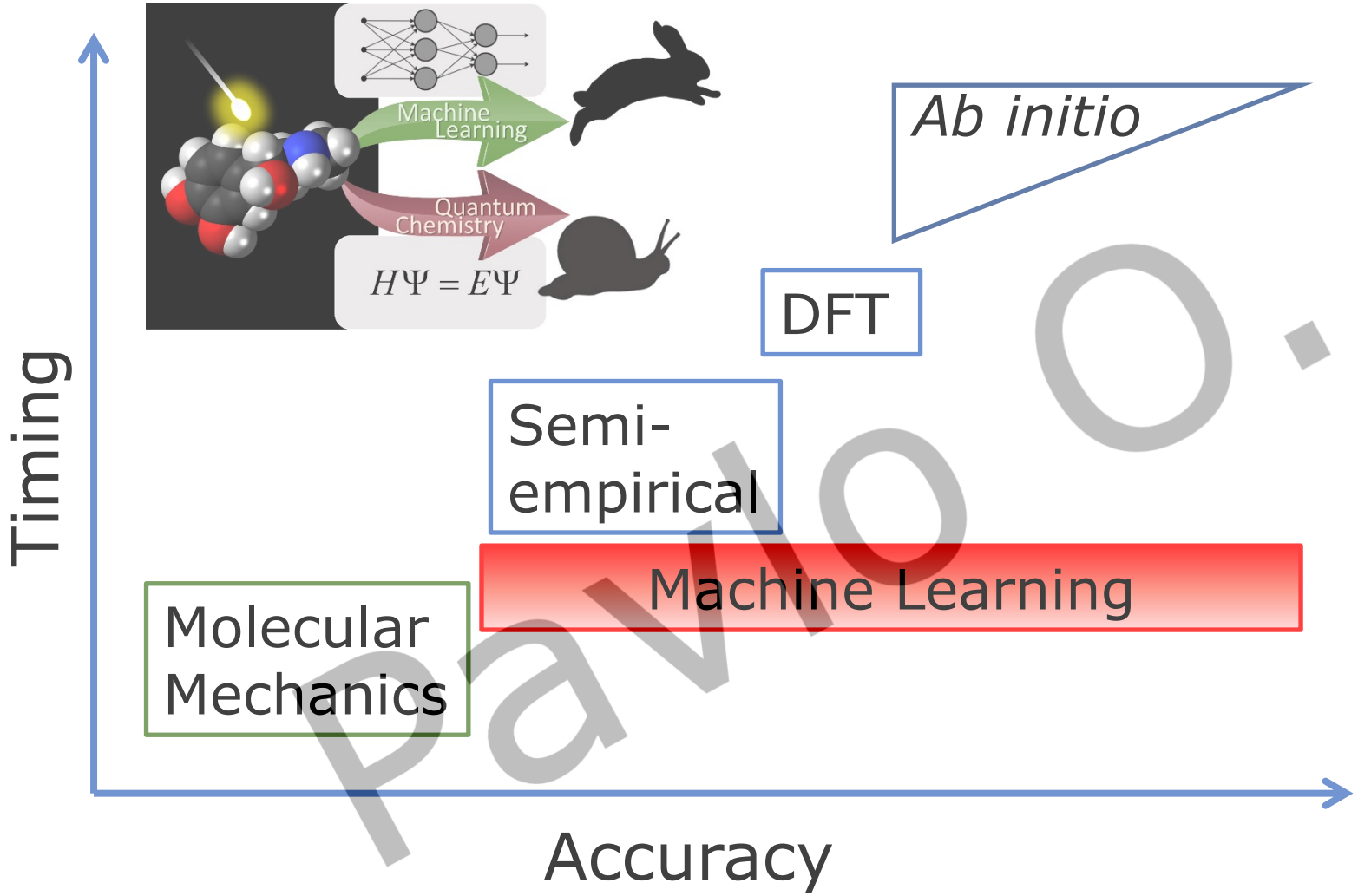
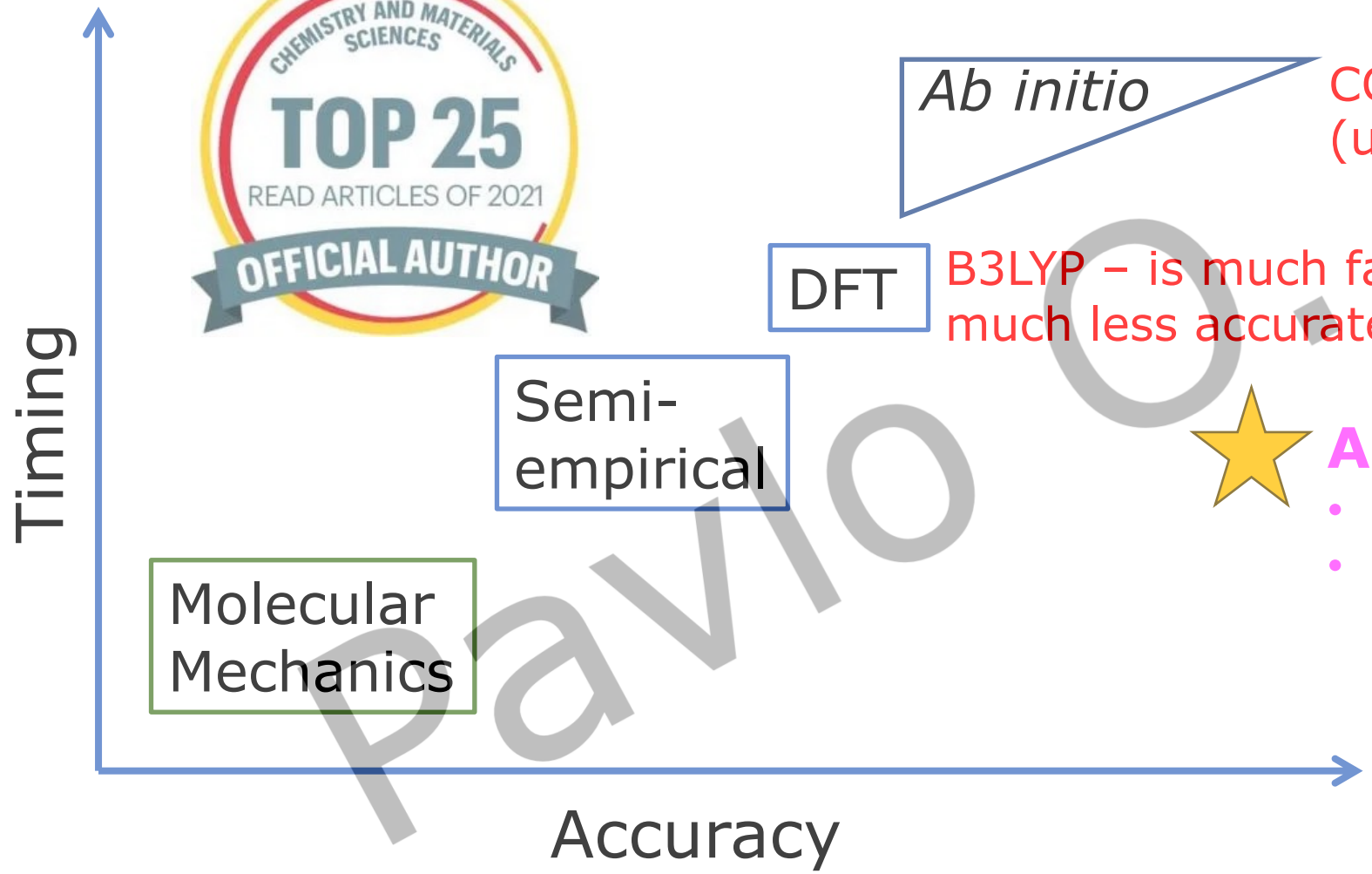


Figure: P. O. Dral, *J. Phys. Chem. Lett.* **2020**, *11*, 2336

Quantum chemistry approximations



CCSD(T) – gold standard, very slow (use it if you can!)

B3LYP – is much faster than the CCSD(T) but is much less accurate

Do not use B3LYP if you can use AIQM1!

AIQM1*

- **Accuracy of CCSD(T)**
- **Orders of magnitude faster than B3LYP**

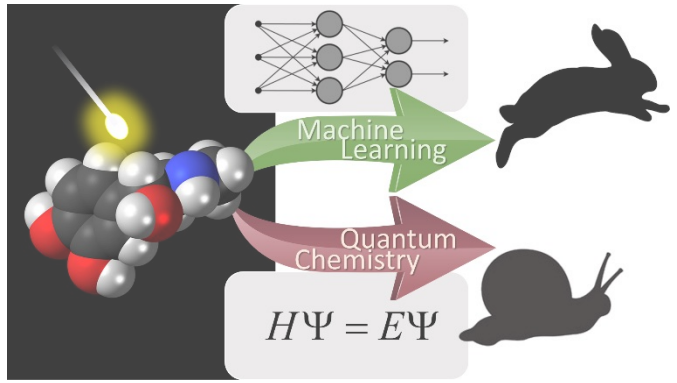
AIQM1: P. Zheng, R. Zubatyuk, W. Wu, O. Isayev, P. O. Dral, *Nat. Commun.* **2021**, *12*, 7022

*CHNO elements only – extensions on the way

P. O. Dral, *J. Phys. Chem. Lett.* **2020**, *11*, 2336; *Chem. Commun.* **2024**, *60*, 3240

ML & Quantum Chemistry

P. O. Dral, M. Barbatti, *Nat. Rev. Chem.* **2021,** *5*, 388



Universal AI-enhanced QM methods

Ab initio / ML-ab initio

DFT / ML-DFT — DM21, CF22D, DENS24…

Semi-empirical / ML-SQC — **AIQM1**

Machine Learning — ANI-1ccx, AIMnet-2, ANI-1xnr, …

Molecular Mechanics

Timing

Accuracy

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

Accessible, versatile, and robust AI-enhanced computational chemistry platform MLatom@XACS:

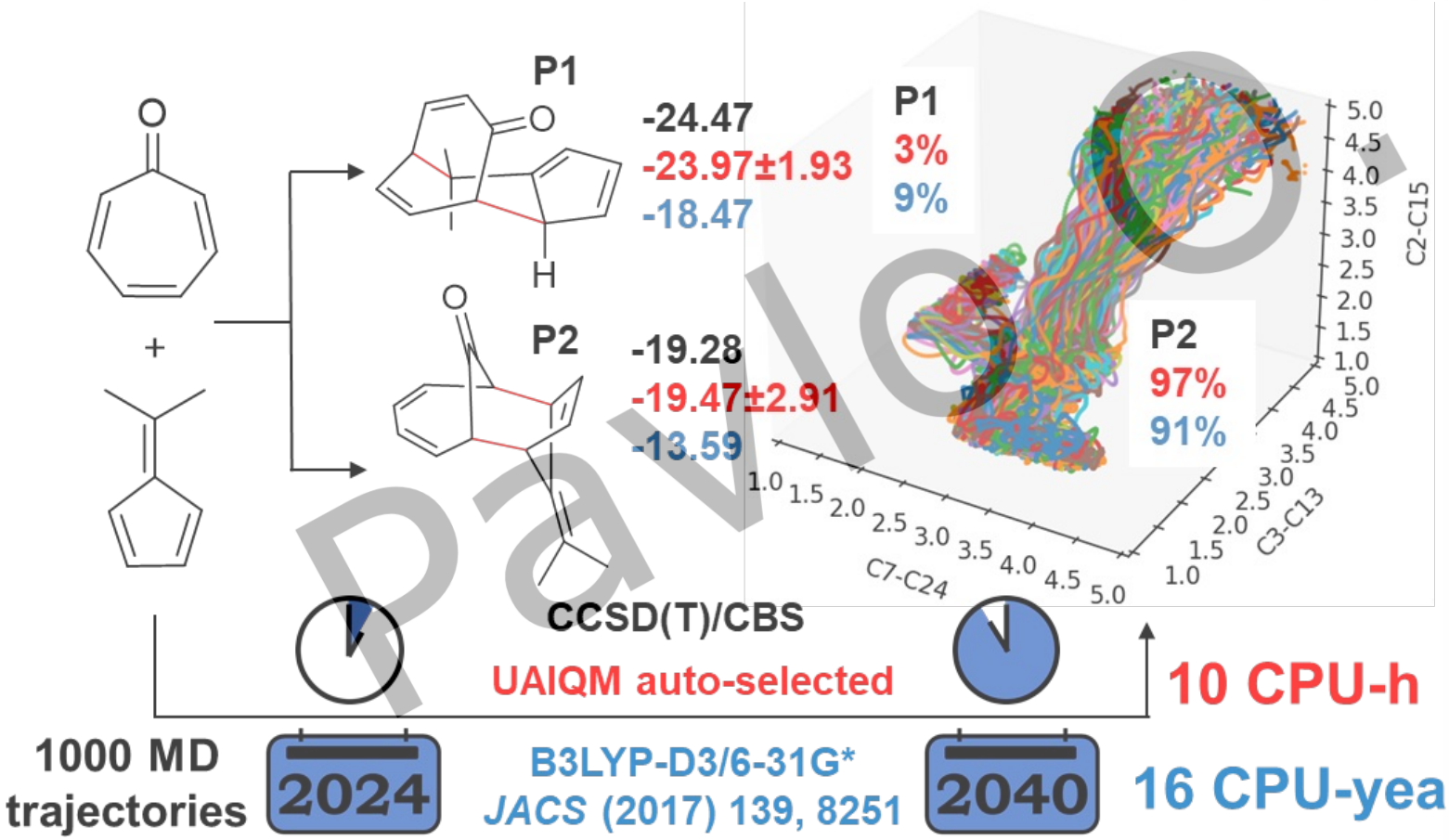
XACS
Xiamen Atomistic Computing Suite
XACScloud.com

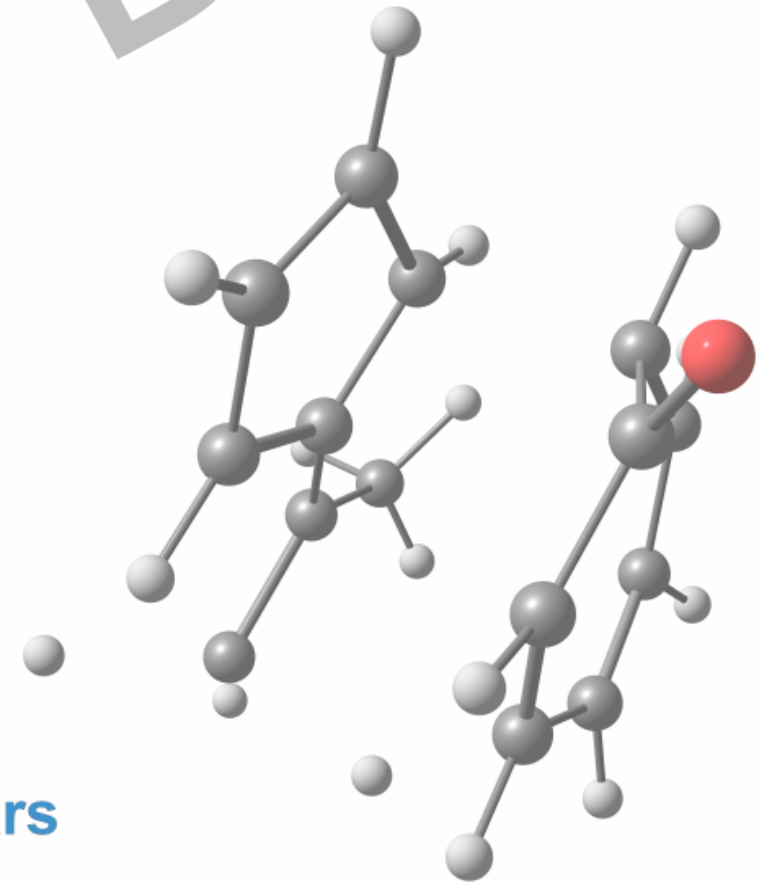
MLatom.com

No reason to do non-ML computational chemistry!

AI methods: **easy to use**, *also* **online**

AI methods are **faster, more accurate, and trustworthy**



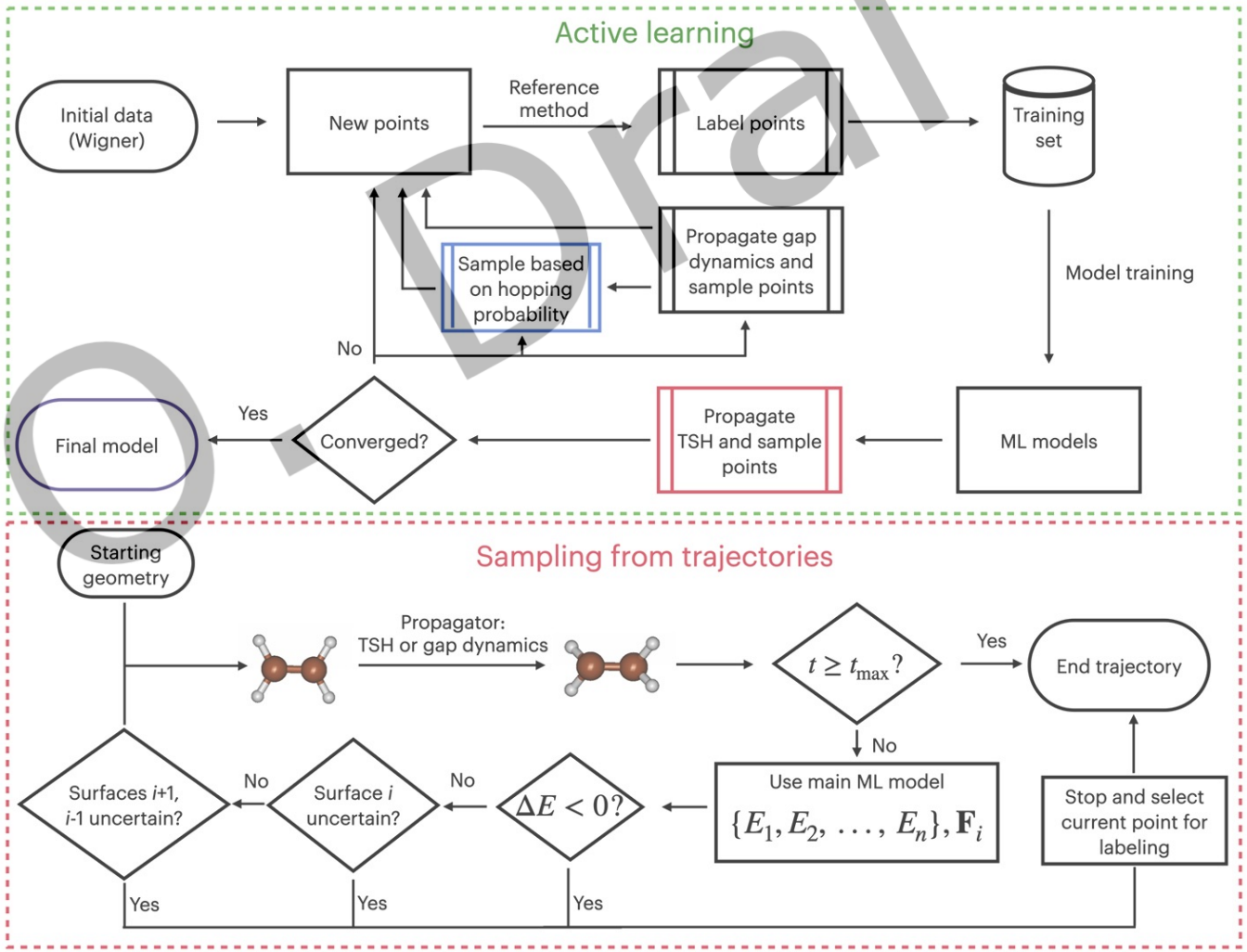


Our latest ML-NAMD results

8 years of research!

Within days, can get precise ML-NAMD results on a single GPU and <16 CPU threads*

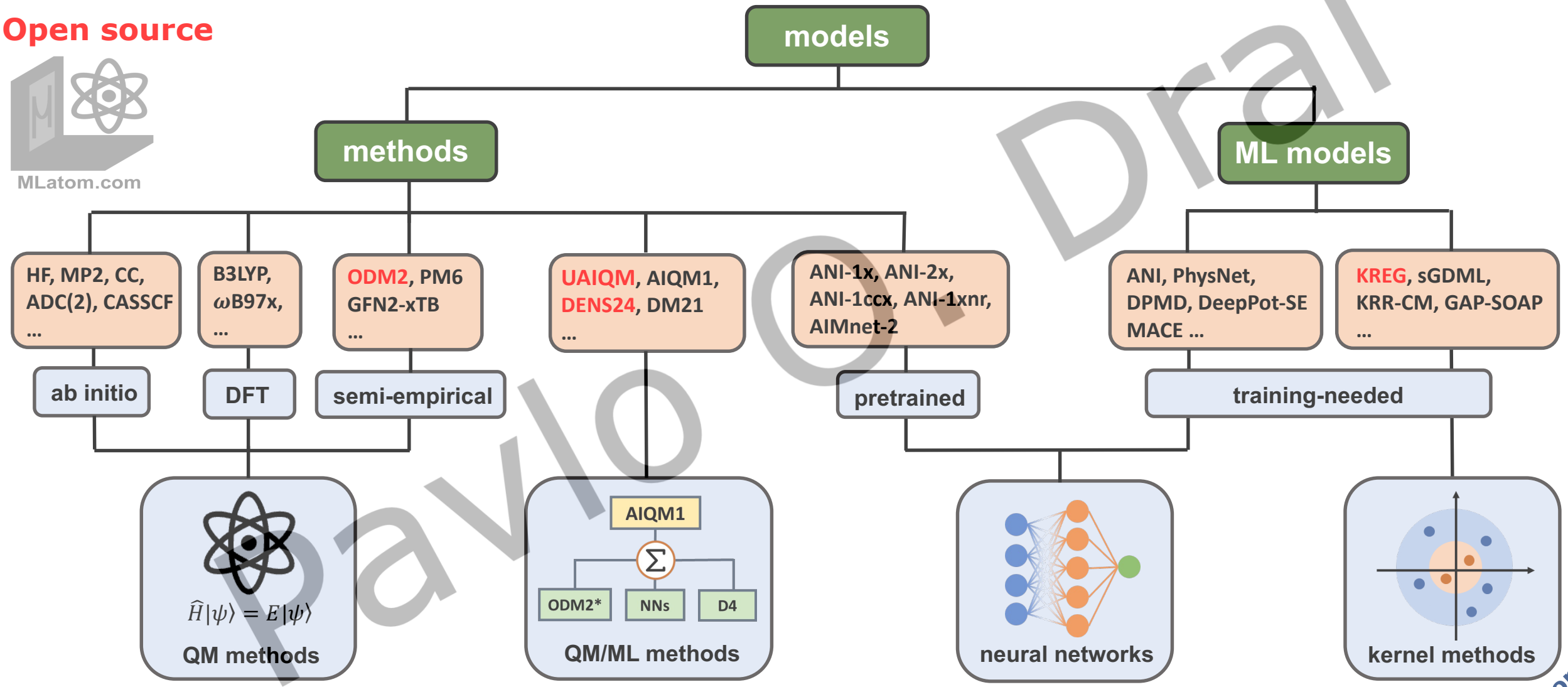
*from scratch for simple molecules



M. Martyka, L. Zhang, F. Ge, Y.-F. Hou, J. Jankowska, M. Barbatti, P. O. Dral. *Charting electronic-state manifolds across molecules with multi-state learning and gap-driven dynamics via efficient and robust active learning*. https://doi.org/10.26434/chemrxiv-2024-dtc1w

AI-enhanced computational chemistry

Open source

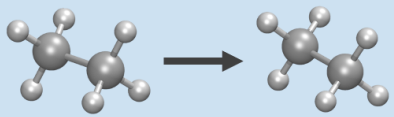


Updated based on: P. O. Dral, F. Ge, Y.-F. Hou, P. Zheng, Y. Chen, M. Barbatti, O. Isayev, C. Wang, B.-X. Xue, M. Pinheiro Jr, Y. Su, Y. Dai, Y. Chen, S. Zhang, L. Zhang, A. Ullah, Q. Zhang, Y. Ou. *JCTC* **2024,** *20*, 1193

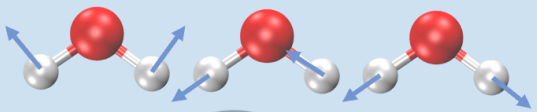
AI-enhanced computational chemistry

Single point calculations — Energies, forces, Hessian matrix…

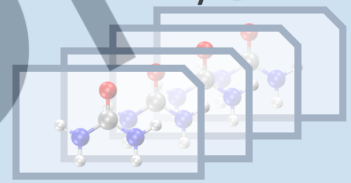
Geometry optimizations



Frequency calculations



Molecular dynamics

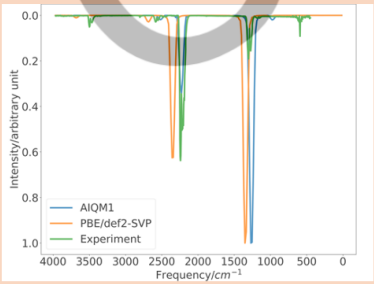


Thermochemistry calculations

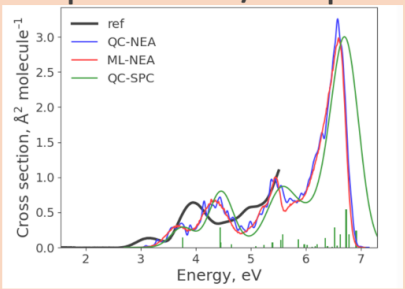
Heat of formation:

$$\Delta H_{\mathrm{at},T} = \left[\sum_A H_T(A)\right] - H_T$$

(Ro)vibrational spectra



One-photon UV/vis spectra



Properties & spectra

Simulations

Quantum dissipative dynamics

Two-photon absorption spectra

P. O. Dral, F. Ge, Y.-F. Hou, P. Zheng, Y. Chen, M. Barbatti, O. Isayev, C. Wang, B.-X. Xue, M. Pinheiro Jr, Y. Su, Y. Dai, Y. Chen, S. Zhang, L. Zhang, A. Ullah, Q. Zhang, Y. Ou. *J. Chem. Theory Comput.* **2024,** *20*, 1193

XACS
Xiamen Atomistic Computing Suite
XACScloud.com

Open source (on github, pip install…); also run on XACScloud.com

MLatom.com

Machine Learning

Simulations

MLatom

Quantum Chemistry

Data

Learning materials

XACS
Xiamen Atomistic Computing Suite
XACScloud.com

MLatom.com

XACS Cloud

Cloud Computing

- Job Submitter
- Terminal
- File Manager
- Job Manager
- Jupyter Lab

Software

- Download

Learning

- Courses

Courses

Testimonials:

"Dr. Dral ofters a fantastic introduction to the concepts around machine learning in chemistry!"

For teachers: Your course can be here!

计算化学与人工智能

计算化学和人工智能迷你课程
作者: Pavlo O.Dral
发布日期: 2024年6月14日

Go

Computational chemistry & AI

Hands-on course on computational chemistry and artificial intelligence (AI) by Pavlo O. Dral

Go

Recommend them further!

Top review from the United States

Patrick Faith

★★★★★ **Singularly Brilliant**

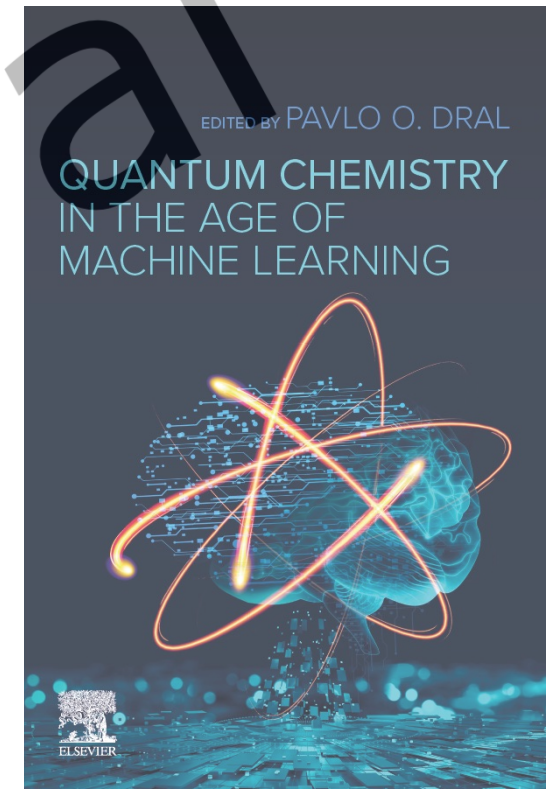
Reviewed in the United States on October 6, 2022

Verified Purchase

I bought this book really with no expectation, it's kind of expensive so was a bit of a stretch buy with really no reviews. Recieved it yesterday and quickly jumped through it, the last book I read that had this level of brilliance was Wheeler's Gravity, but this book has much more of a team spirit reminding me of the early days of quantum mechanics. Only modern book that it reminds me of is Wolframs recent "new kind of ..." but this book is completely practical, team based, and I think a signifacly better path.

I noticed a lot of research lead by the Xiamen area, connected to europe with a bit of usa work. Represents a major shift in in leadership of science, is a book every one should atleast attempt to glance through to "get" what is now going on.

27 chapters
65 authors!



AI is a game changer

P. O. Dral, M. Barbatti, *Nat. Rev. Chem.* **2021,** *5*, 388

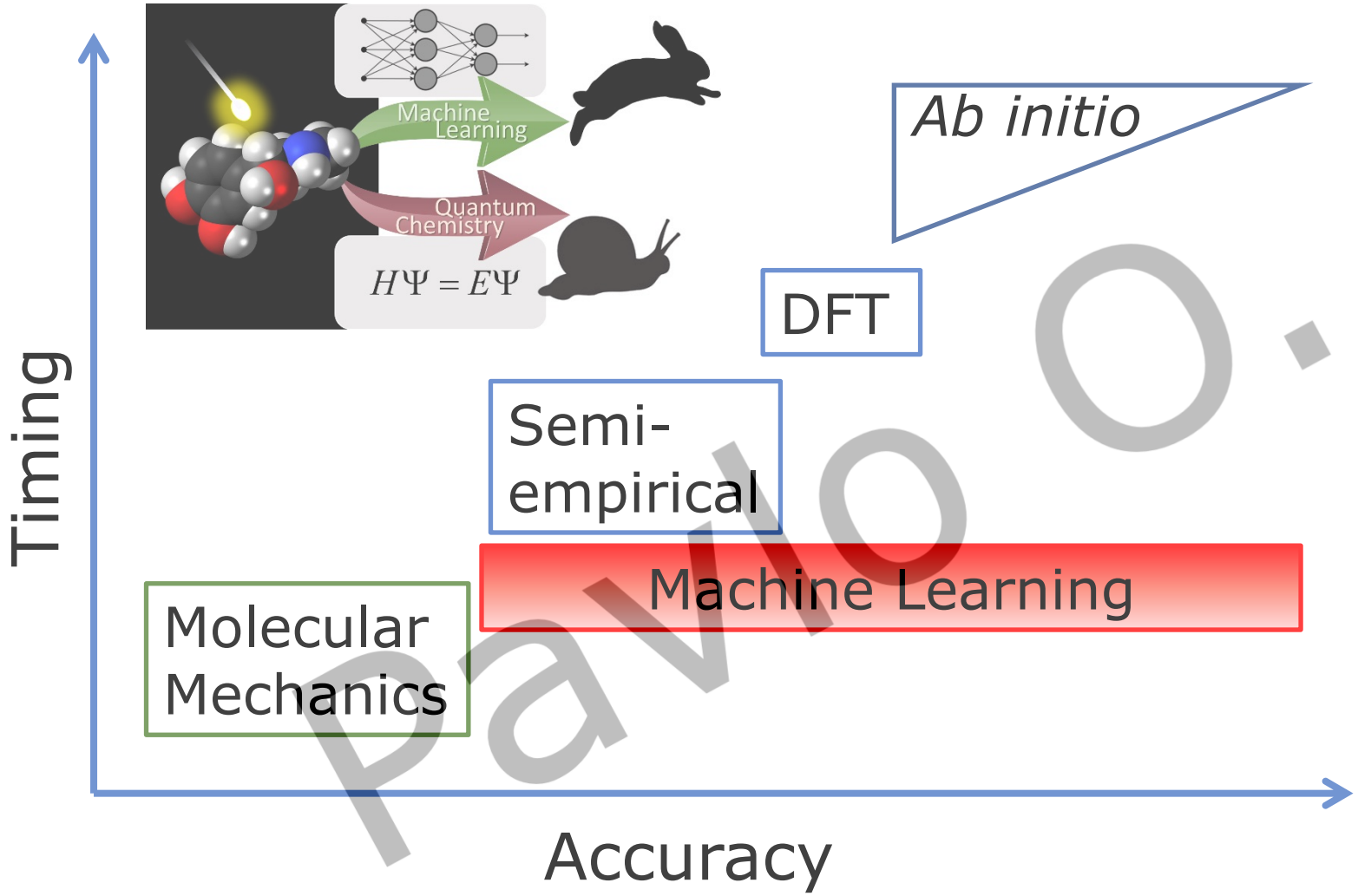


Figure: P. O. Dral, *J. Phys. Chem. Lett.* **2020**, *11*, 2336

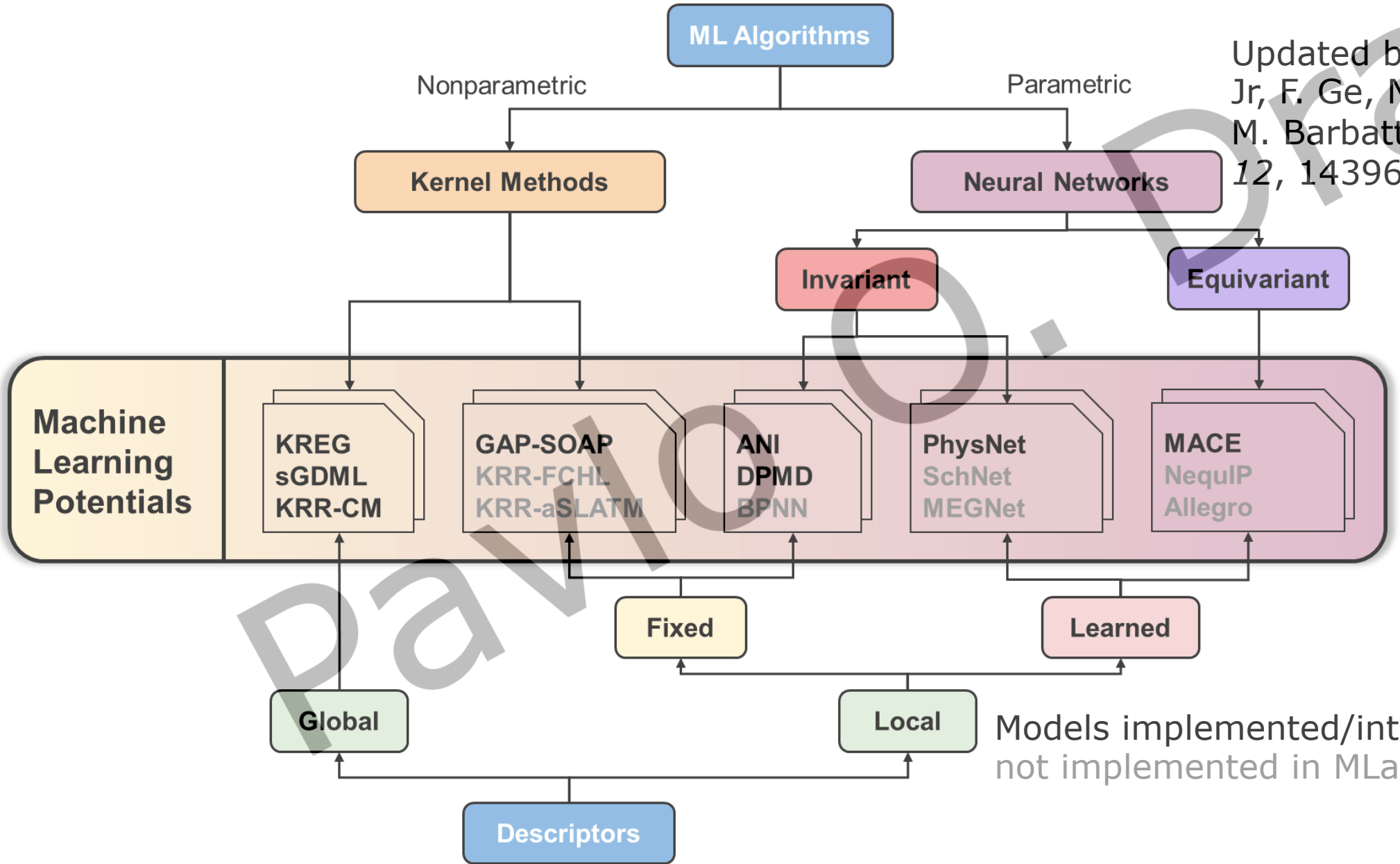
Machine learning potentials

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

$$F_{A,d} = -\frac{\partial E}{\partial x_{A,d}} = -\frac{\partial E^{\text{machine learning potential}}(\mathbf{x})}{\partial x_{A,d}}$$

Zoo of machine learning potentials

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

Supervised Machine Learning

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

Input (x) → *f*(x) → Output (y)

Given collection of known {x,y} find a function *f*(x)

Supervised Machine Learning

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

Input (x) → *f*(x) → Output (y)

↓

Given collection of known {x,y} find a function *f*(x)

↓

Use this function for making new predictions given just {x'}

Supervised Machine Learning

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

Input (x) → *f*(x) → Output (y)

Given collection of known {x,y} find a function *f*(x)

training set train ML model

Use this function for making new predictions given just {x'}

Supervised Machine Learning

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

Input (x) → *f*(x) → Output (y)

Given collection of known {x,y} find a function *f*(x)

training set train ML model

Use this function for making new predictions given just {x'}

- Data
- Choice of x (descriptor)
- Choice of y (labels)
- Fitting function (ML algorithm, ML model)
- Optimization of ML model parameters

Supervised Machine Learning

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

Input (x) → *f*(x) → Output (y)

Given collection of known {x,y} find a function *f*(x)

training set train ML model

Use this function for making new predictions given just {x'}

- Data
- Choice of x (descriptor)
- Choice of y (labels)
- Fitting function (**ML algorithm**, ML model)
- Optimization of ML model parameters

ML algorithms

ML Algorithms

- Various types of neural networks (NN), **deep learning**

- Gaussian processes (GP)
- Kernel ridge regression (KRR)
- Support vector machines (SVMs) & support vector regression (SVR)

- Linear regression!
- Decision trees
- k-Nearest neighbor algorithm
- and many more…

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

Parametric vs nonparametric algorithms

Some of the ML Algorithms

f(x; **parameters**)

Linear regression

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

Number of parameters is fixed: parametric model

Neural networks are also parametric models

Kernel ridge regression (KRR)

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

Number of parameters depends on number of training points: nonparametric model, e.g. KRR

ML Algorithms

- Various types of neural networks (NN), **deep learning**

- Gaussian processes (GP)
- Kernel ridge regression (KRR)
- Support vector machines (SVMs) & support vector regression (SVR)

- Linear regression!
- Decision trees
- k-Nearest neighbor algorithm
- and many more…

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)

Neural networks (NN)

Quantum Chemistry in the Age of Machine Learning. Ed. P. O. Dral. Elsevier: Amsterdam, Netherlands, **2023**.

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)

Neural networks (NN)

Quantum Chemistry in the Age of Machine Learning. Ed. P. O. Dral. Elsevier: Amsterdam, Netherlands, **2023**.

Linear regression

Multiple linear regression

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

$$f(\mathbf{x}_i;\boldsymbol{\beta}) = \sum_{j=1}^{p} \beta_j x_{ij}$$

$$f(\mathbf{x}_i;\boldsymbol{\beta}) = \mathbf{x}_i^T \boldsymbol{\beta}$$

How to find the coefficients β_j?

Linear regression

Multiple linear regression

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

$$f(\mathbf{x}_i;\boldsymbol{\beta}) = \sum_{j=1}^{p} \beta_j x_{ij}$$

$$f(\mathbf{x}_i;\boldsymbol{\beta}) = \mathbf{x}_i^T \boldsymbol{\beta}$$

We can find the coefficients $\boldsymbol{\beta}$ using the method of least squares, where coefficients are fit to get the minimum residual sum of squares (RSS) with respect to the training set with N_{tr} reference values $\mathbf{y}$:

$$\arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{N_{tr}} (f(\mathbf{x}_i;\boldsymbol{\beta}) - y_i)^2$$

Linear regression

$$\arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{N_{tr}} (f(\mathbf{x}_i; \boldsymbol{\beta}) - y_i)^2$$

$$L(\boldsymbol{\beta}) = \sum_{i=1}^{N_{tr}} \left(\mathbf{x}_i^T \boldsymbol{\beta} - y_i\right)^2$$

$$L(\boldsymbol{\beta}) = (\mathbf{X}\boldsymbol{\beta} - \mathbf{y})^T (\mathbf{X}\boldsymbol{\beta} - \mathbf{y})$$

$$\mathbf{X} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{N_{\mathrm{tr}}1} & \cdots & x_{N_{\mathrm{tr}}p} \end{pmatrix}$$

$$\frac{\partial L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = 2\mathbf{X}^T (\mathbf{X}\boldsymbol{\beta} - \mathbf{y}) = \mathbf{0}$$

Linear regression

$$\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^T\mathbf{y}$$

$$\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

Linear regression has an analytical solution!

While it is very advantageous, it assumes that the data follow the linear distribution, which is often not the case

Linear regression

Task 4: Fit linear model $E = aR$ on a training set with 20 points sampled along H_2 dissociation curve (energies E in Hartree at FCI/aug-cc-pV6Z; internuclear distances R in Angstrom)

Calculate R^2 and residual sum of squares (RSS)

Task 5: Fit linear regression with intercept *b*?

Calculate R^2 and residual sum of squares (RSS)

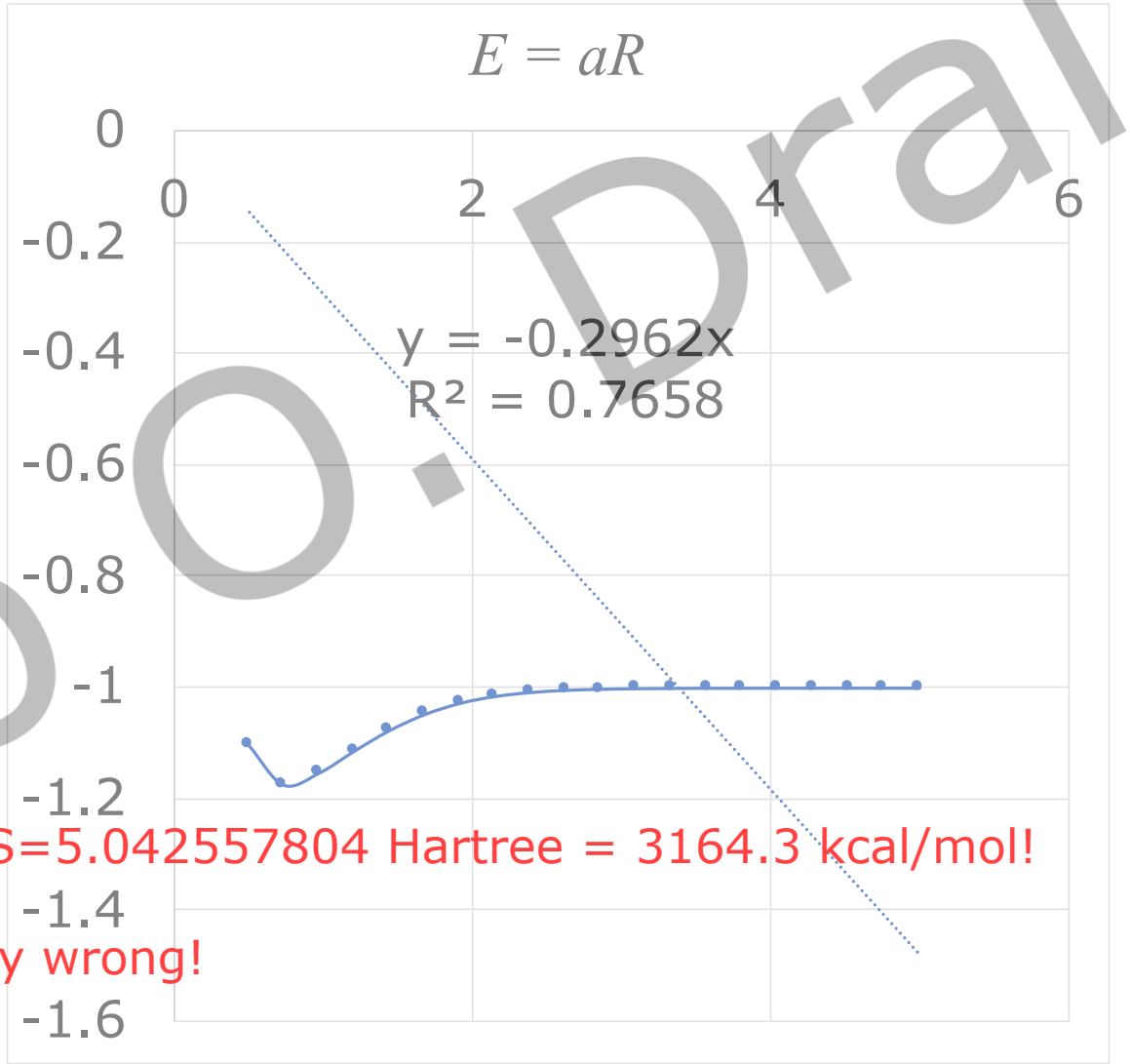
$$E = aR+b$$

$$x_{i1} = R$$

$$\mathrm{RSS} = \sum_{i=1}^{N_{tr}} (f(\mathbf{x}_i; \boldsymbol{\beta}) - y_i)^2$$

Linear regression

H_2 dissociation curve



RSS=5.042557804 Hartree = 3164.3 kcal/mol!

Very wrong!

Linear regression

Task 5: Fit linear regression with intercept *b*?

Calculate R^2 and residual sum of squares (RSS)

$$E = aR + b$$

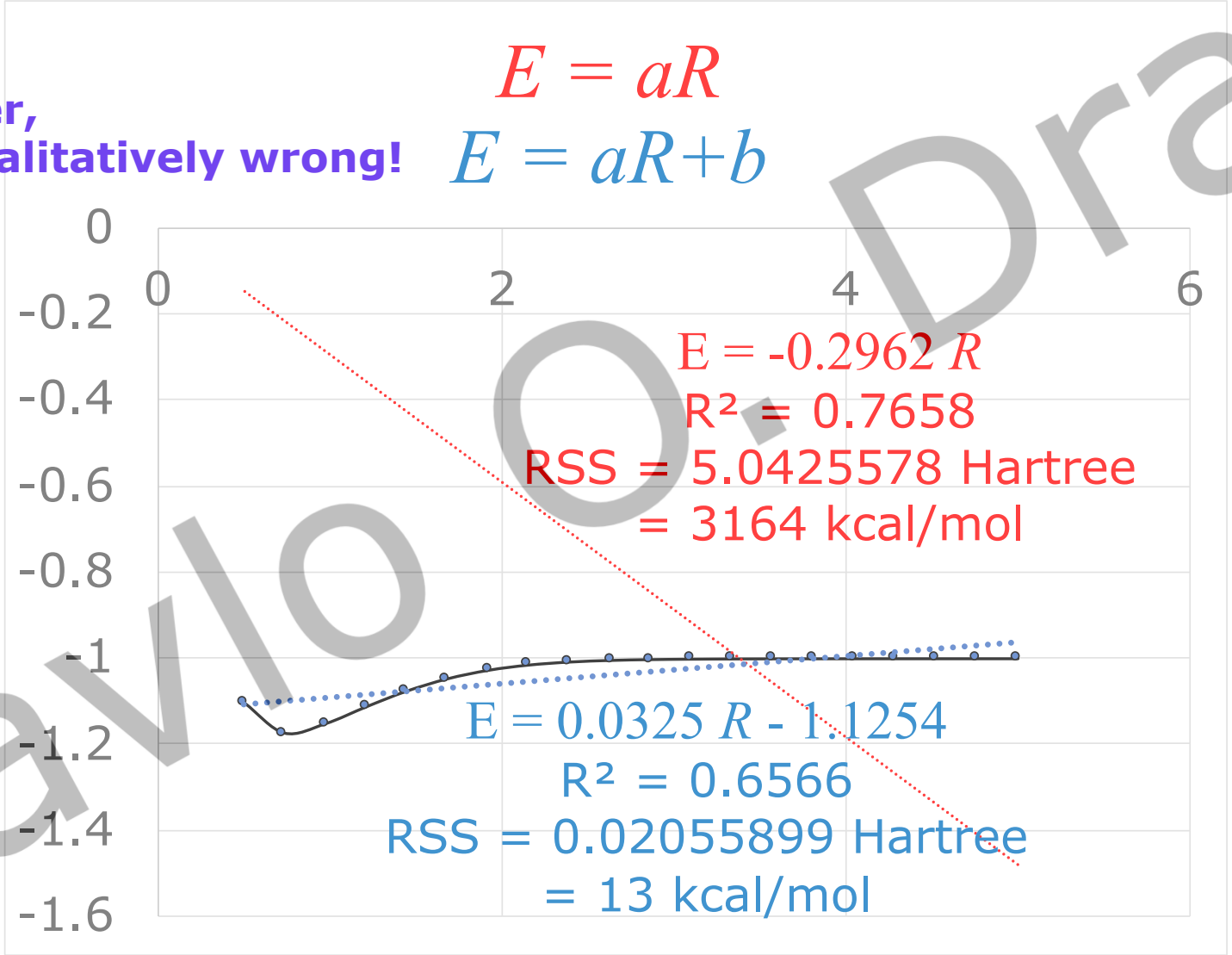
$$\mathrm{RSS} = \sum_{i=1}^{N_{tr}} (f(\mathbf{x}_i; \boldsymbol{\beta}) - y_i)^2$$

Linear regression

Much better,
but still qualitatively wrong!

$$E = aR$$

$$E = aR+b$$



Linear regression

Q: What about linear regression with intercept *b*?

$$E = aR + b$$

It is equivalent to mapping function $R \rightarrow (R, 1)$, $\Phi((R)) = (R, 1)$, where Φ maps from p-dimensional input space into p^d-dimensional feature space
Now we can solve multiple linear regression with two variables

$$f(\mathbf{x}_i; \boldsymbol{\beta}) = \beta_1 x_{i1} + \beta_2 x_{i2} = \beta_1 R + \beta_2 1 = aR + b$$

$$x_{i1} = R_i$$

$$x_{i2} = 1$$

$$\beta_1 = a$$

$$\beta_2 = b$$

Linear regression

Any ideas how to get the dissociation curve shape right?

Linear regression

Q: Any ideas how to get the dissociation curve shape right?

We can use mapping $R \rightarrow (R^{-6}, R^{-12}, 1)$ inspired by Lennard-Jones potential, this allows us to treat data set (*E, R*) nonlinear in input space (*R*) using vectors in feature space $(R^{-6}, R^{-12}, 1)$
Now we can solve multiple linear regression with three variables:

$$f(\mathbf{x}_i; \boldsymbol{\beta}) = \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} = \beta_1 R_i^{-6} + \beta_2 R_i^{-12} + \beta_3 1 = a R_i^{-6} + b R_i^{-12} + c$$

$$x_{i1} = R_i^{-6}$$

$$x_{i2} = R_i^{-12}$$

$$x_{i3} = 1$$

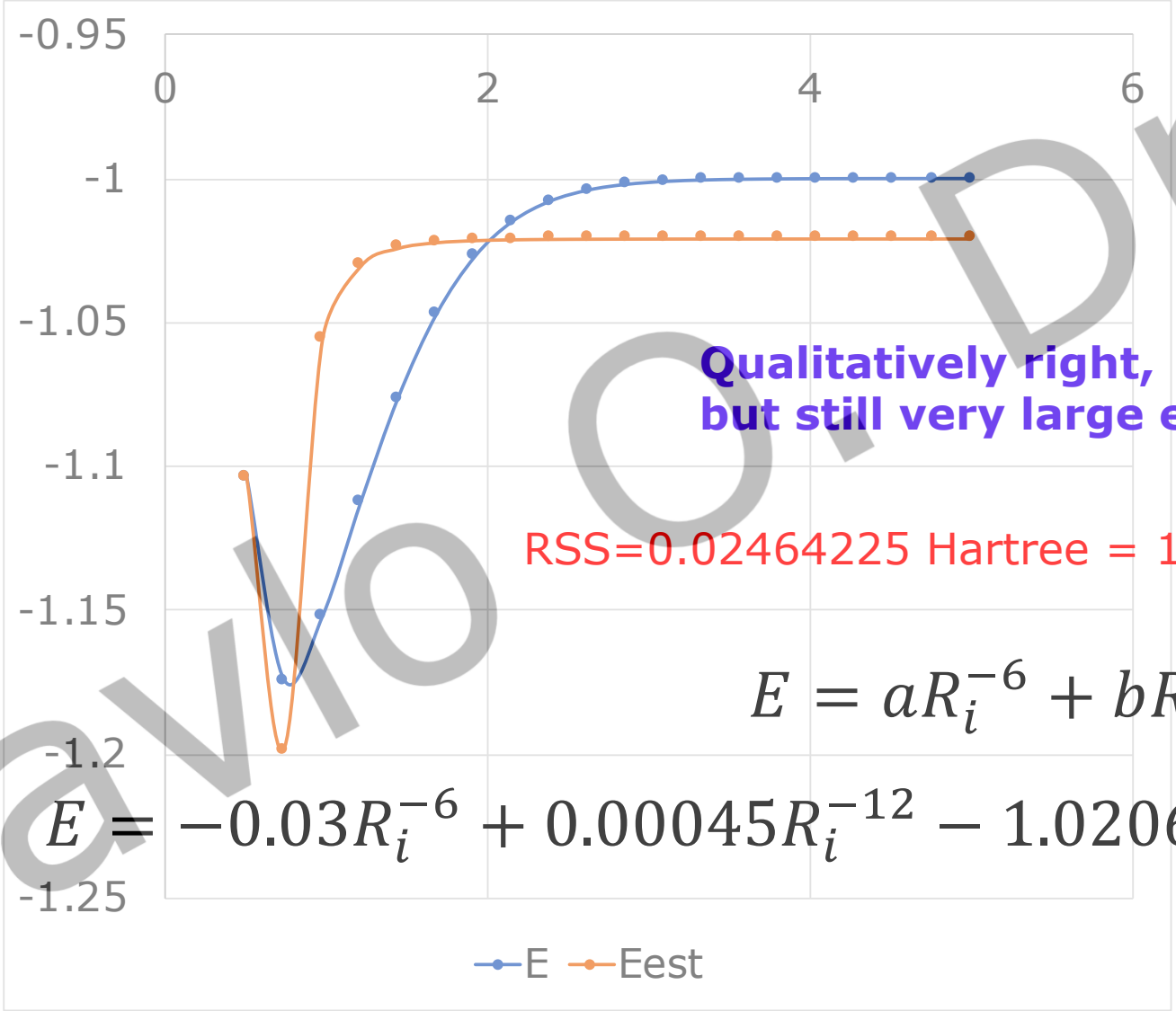
$$\beta_1 = a$$

$$\beta_2 = b$$

$$\beta_3 = c$$

Max Pinheiro Jr, P. O. Dral, Kernel methods.
In *Quantum Chemistry in the Age of Machine Learning*, P. O. Dral, Ed. Elsevier: **2023**.
Paperback ISBN: 9780323900492

Linear regression



Qualitatively right,
but still very large error!

RSS=0.02464225 Hartree = 15 kcal/mol!

$$E = aR_i^{-6} + bR_i^{-12} + c$$

$$E = -0.03R_i^{-6} + 0.00045R_i^{-12} - 1.02060806$$

Linear regression

Can we extend it to more variables and make it more flexible?

Yes! We can go to infinite number of variables!

How?

Using a kernel trick

Linear regression

Let's rewrite the linear regression equation by representing the regression coefficients via a sum over all training points:

$$f(\mathbf{x}';\boldsymbol{\beta}) = \sum_{j=1}^{p} \beta_j x'_j$$

$$\beta_j = \sum_{i=1}^{N_{tr}} \alpha_i x_{ij}$$

$$f(\mathbf{x}') = \sum_{j=1}^{p} \left(\sum_{i=1}^{N_{tr}} \alpha_i x_{ij} \right) x'_j = \sum_{i=1}^{N_{tr}} \alpha_i \sum_{j=1}^{p} x_{ij} x'_j = \sum_{i=1}^{N_{tr}} \alpha_i \mathbf{x}_i^T \mathbf{x}'$$

$$\mathbf{x}_i^T \mathbf{x}' = \langle \mathbf{x}_i, \mathbf{x}' \rangle$$

Dot-product = inner product = scalar product of two vectors

Linear regression

As we have seen before, we can map vectors $\mathbf{x}$ and $\mathbf{x}'$ from p-dimensional input space into p^d-dimensional feature space using mapping function Φ:

$$f(\mathbf{x}') = \sum_{i=1}^{N_{tr}} \alpha_i \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}')$$

In previous examples we new the mapping function and explicit forms of vectors in the feature space. But all we need is a dot-product between vectors in the feature space, not their explicit forms. Such dot-product is called ***kernel*** denoted $k(\mathbf{x}_i, \mathbf{x}')$ and it is calculated in using vectors in the input space (not feature space!):

$$k(\mathbf{x}_i, \mathbf{x}') = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}')$$

The kernel trick is substitution of the calculation of the dot-product using explicit representations of vectors in the feature space by using a kernel function:

$$f(\mathbf{x}') = \sum_{i=1}^{N_{tr}} \alpha_i k(\mathbf{x}_i, \mathbf{x}')$$

Kernel-based machine learning

$$f(\mathbf{x}') = \sum_{i=1}^{N_{tr}} \alpha_i k(\mathbf{x}_i, \mathbf{x}')$$

This is a kernel-based machine learning function.

Kernel trick allows us to use tools of linear regression for data nonlinear in the input space by converting variables into (higher dimensional) feature space.

Q: How to find the regression coefficients α?

Kernel-based machine learning

$$f(\mathbf{x}') = \sum_{i=1}^{N_{tr}} \alpha_i k(\mathbf{x}_i, \mathbf{x}')$$

This is a kernel-based machine learning function.

Kernel trick allows us to use tools of linear regression for data nonlinear in the input space by converting variables into (higher dimensional) feature space.

We can find the coefficients $\boldsymbol{\alpha}$ using the method of least squares, where coefficients are fit to get the minimum residual sum of squares (RSS) with respect to the training set with N_{tr} reference values $\mathbf{y}$:

$$\arg\min_{\boldsymbol{\alpha}} \sum_{i=1}^{N_{tr}} (f(\mathbf{x}_i; \boldsymbol{\alpha}) - y_i)^2$$

Kernel-based machine learning

$$\arg\min_{\boldsymbol{\alpha}} \sum_{i=1}^{N_{tr}} (f(\mathbf{x}_i;\boldsymbol{\alpha}) - y_i)^2$$

$$L(\boldsymbol{\beta}) = \sum_{i=1}^{N_{tr}} (f(\mathbf{x}_i;\boldsymbol{\alpha}) - y_i)^2$$

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

$$L(\boldsymbol{\beta}) = \sum_{i=1}^{N_{tr}} \left(\sum_{j=1}^{N_{tr}} \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) - y_i \right)^2$$

$$L(\boldsymbol{\beta}) = (\mathbf{K}\boldsymbol{\alpha} - \mathbf{y})^T (\mathbf{K}\boldsymbol{\alpha} - \mathbf{y})$$

$$\mathbf{K} = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k\left(\mathbf{x}_1, \mathbf{x}_{N_{\mathrm{tr}}}\right) \\ \vdots & \ddots & \vdots \\ k\left(\mathbf{x}_{N_{\mathrm{tr}}}, \mathbf{x}_1\right) & \cdots & k\left(\mathbf{x}_{N_{\mathrm{tr}}}, \mathbf{x}_{N_{\mathrm{tr}}}\right) \end{pmatrix}$$

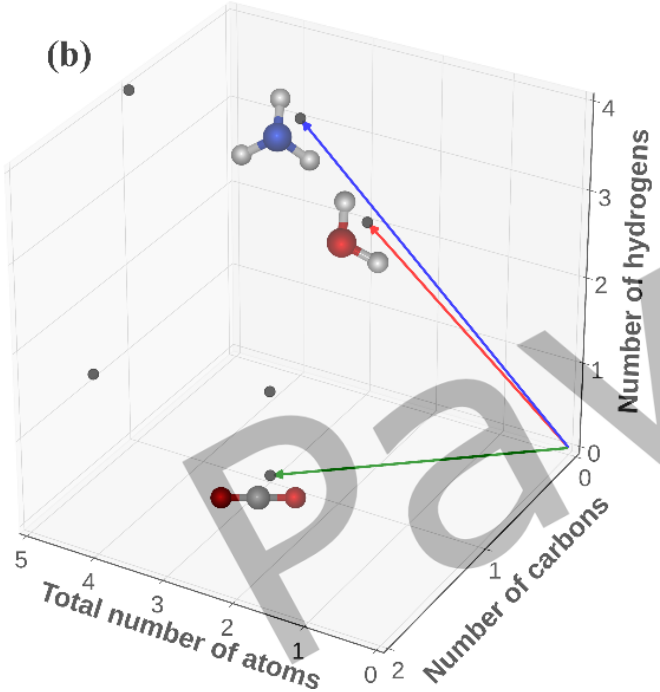
Kernel matrix

Kernel-based machine learning

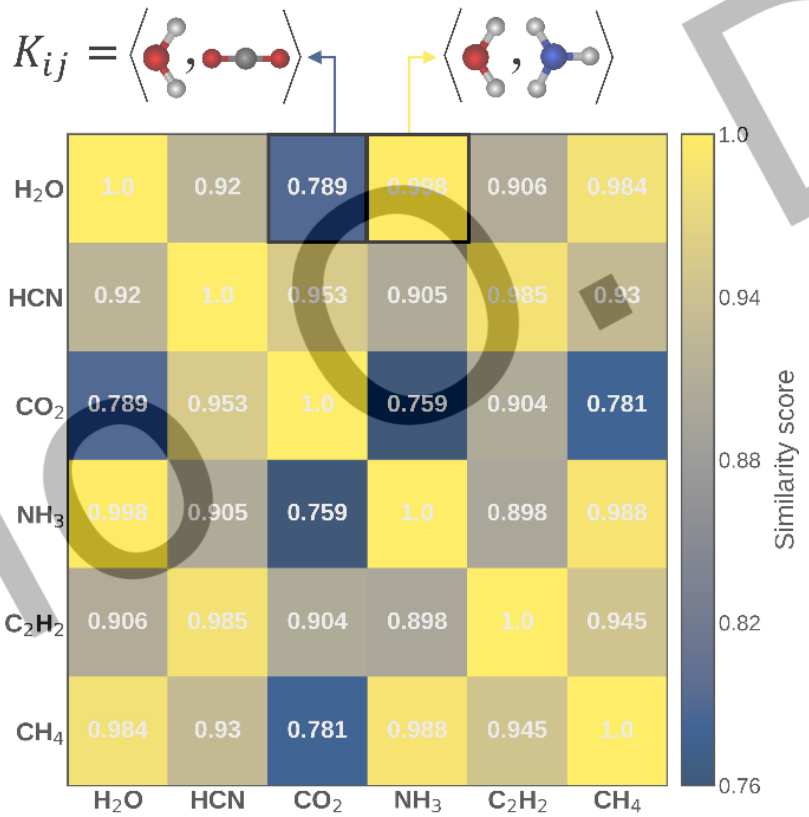
(a)

Molecule	num_atoms	num_carbons	num_hydrogens
H2O	3	0	2
HCN	3	1	1
CO2	3	1	0
NH3	4	0	3
C2H2	4	2	2
CH4	5	1	4

(b)



(c)



Kernel matrix
(matrix measuring similarities between points – here cosine)

$$\cos\theta = \mathbf{a}\cdot\mathbf{b}/(\|\mathbf{a}\|\|\mathbf{b}\|)$$

Max Pinheiro Jr, P. O. Dral, Kernel methods. In *Quantum Chemistry in the Age of Machine Learning*, P. O. Dral, Ed. Elsevier: **2023**, Paperback ISBN: 9780323900492

Kernel-based machine learning

$$\arg\min_{\boldsymbol{\alpha}} \sum_{i=1}^{N_{tr}} (f(\mathbf{x}_i; \boldsymbol{\alpha}) - y_i)^2$$

$$L(\boldsymbol{\beta}) = (\mathbf{K}\boldsymbol{\alpha} - \mathbf{y})^T (\mathbf{K}\boldsymbol{\alpha} - \mathbf{y})$$

$$\frac{\partial L(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} = 2\mathbf{K}^T(\mathbf{K}\boldsymbol{\alpha} - \mathbf{y}) = 2\mathbf{K}(\mathbf{K}\boldsymbol{\alpha} - \mathbf{y}) = 2\mathbf{K}\mathbf{K}\boldsymbol{\alpha} - 2\mathbf{K}\mathbf{y} = \mathbf{0}$$

$$\mathbf{K}\mathbf{K}\boldsymbol{\alpha} = \mathbf{K}\mathbf{y}$$

$$\mathbf{K}^{-1}\mathbf{K}\mathbf{K}\boldsymbol{\alpha} = \mathbf{K}^{-1}\mathbf{K}\mathbf{y}$$

$$\mathbf{K}\boldsymbol{\alpha} = \mathbf{y}$$

$$\boldsymbol{\alpha} = \mathbf{K}^{-1}\mathbf{y}$$

Kernel-based ML vs linear regression

$$f(\mathbf{x}') = \sum_{i=1}^{N_{tr}} \alpha_i k(\mathbf{x}_i, \mathbf{x}')$$

This is a kernel-based machine learning function.

If the kernel function is itself a dot-product:

$$k(\mathbf{x}_i, \mathbf{x}') = \mathbf{x}_i^T \mathbf{x}'$$

The expression becomes equivalent to the linear regression as we have seen above and that is why such a dot-product kernel is also called "linear kernel":

$$f(\mathbf{x}') = \sum_{i=1}^{N_{tr}} \alpha_i k(\mathbf{x}_i, \mathbf{x}') = \sum_{i=1}^{N_{tr}} \alpha_i \mathbf{x}_i^T \mathbf{x}' = \sum_{j=1}^{p} \beta_j x_j'$$

$$\beta_j = \sum_{i=1}^{N_{tr}} \alpha_i x_{ij}$$

Kernel-based machine learning

$$\beta_j = \sum_{i=1}^{N_{tr}} \alpha_i x_{ij}$$

One can get the same β_j for infinite combinations of α_i

Some solutions will have very large α_i with opposite signs trying to compensate each other

$$\beta = \sum_{i=1}^{N_{\mathrm{tr}}=20} \alpha_i x_i = \sum_{i=1}^{N_{\mathrm{tr}}=20} \alpha_i R_i$$

$$\beta = (-0.2962 \cdot 2) \cdot 0.5 + \sum_{i=2}^{N_{\mathrm{tr}}=20} 0 x_i$$

$$\beta = 198476910439 \cdot 0.5 - 19847691043.95924 \cdot 5 + \sum_{i=3}^{N_{\mathrm{tr}}=20} 0 x_i$$

$$f(x;\beta) = \beta x = -0.2962x$$

$$\boldsymbol{\alpha} = \mathbf{K}^{-1}\mathbf{y}$$

Prone to overfitting, numerically unstable
Often, K is not invertible matrix

Shrinkage methods

Ridge regression – belongs to shrinkage methods (useful for feature importance analysis)

$$\arg\min_{\boldsymbol{\beta}} (\mathbf{X}\boldsymbol{\beta} - \mathbf{y})^T (\mathbf{X}\boldsymbol{\beta} - \mathbf{y}) + \lambda \boldsymbol{\beta}^T \boldsymbol{\beta}$$

$$\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}$$

Another example of shrinkage method is the lasso

$$\arg\min_{\boldsymbol{\beta}} (\mathbf{X}\boldsymbol{\beta} - \mathbf{y})^T (\mathbf{X}\boldsymbol{\beta} - \mathbf{y}) + \lambda \sum_{i=1}^{p} |\beta_i|$$

Identity matrix

$$\mathbf{I} = \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix}$$

Kernel ridge regression (KRR)

$$\arg\min_{\boldsymbol{\alpha}} (\mathbf{K}\boldsymbol{\alpha} - \mathbf{y})^T (\mathbf{K}\boldsymbol{\alpha} - \mathbf{y}) + \lambda \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha}$$

$$\boldsymbol{\alpha} = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{y}$$

Coefficient magnitude is forced to shrunk with larger λ in these methods

λ is nonnegative regularization hyperparameter, smoothens function and makes solution numerically more stable.

Kernel-based ML with Gaussian kernel

$$f(\mathbf{x}') = \sum_{i=1}^{N_{\mathrm{tr}}} \alpha_i k(\mathbf{x}_i, \mathbf{x}')$$

This is a kernel-based machine learning function.

One of the popular kernel functions is the Gaussian kernel function:

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

It maps vectors **x** from N_x-dimensional input space into infinite-dimensional feature space.

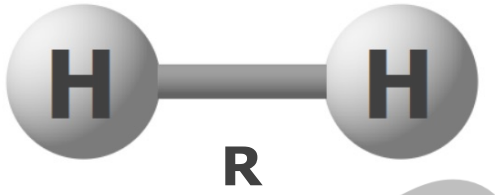
σ is a positive hyperparameter defining the length scale of the Gaussian function.

Model Selection

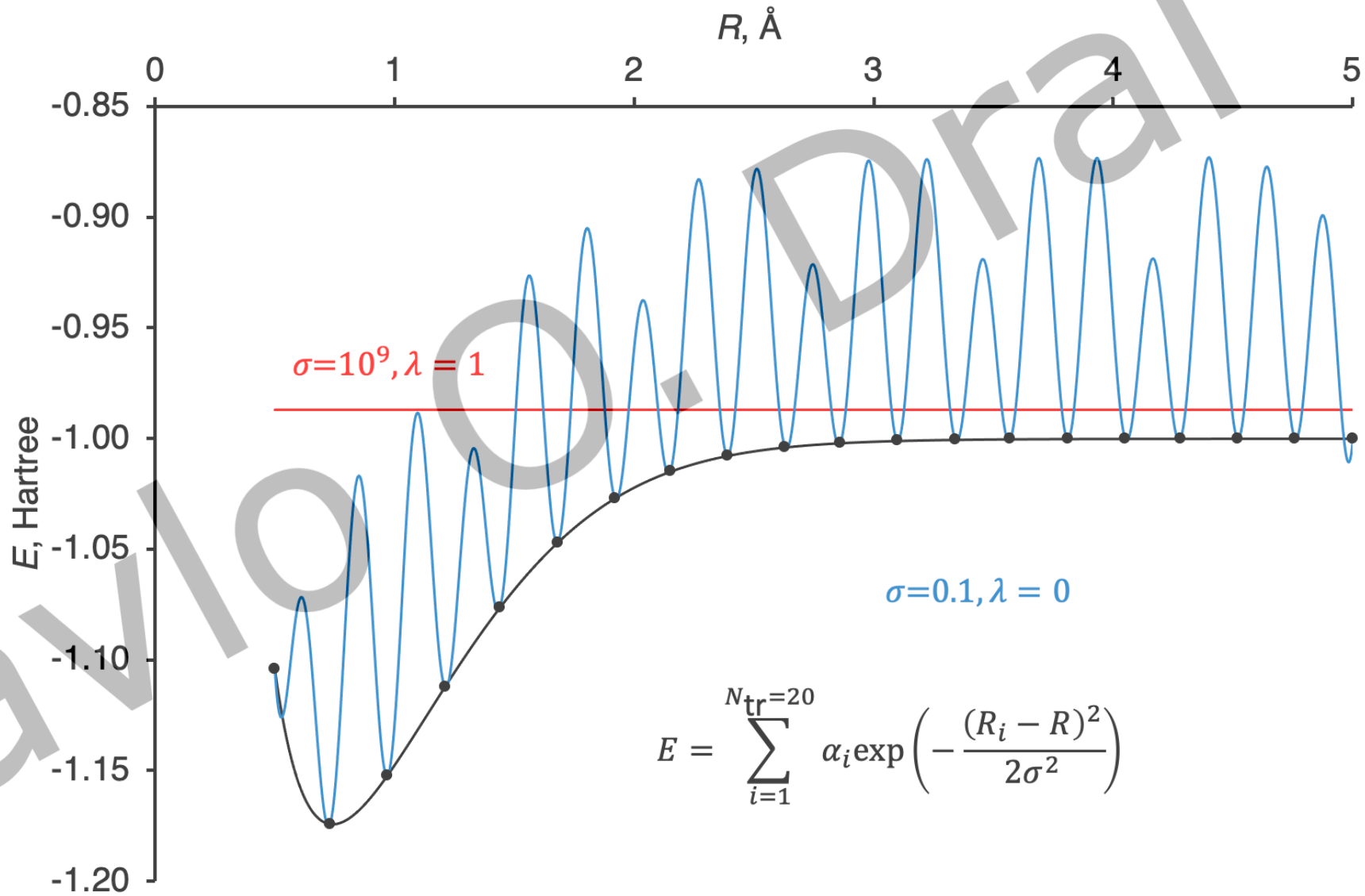
Many ML algorithms can give you practically ideal predictions for their own training set

Underfitting
Low variance
High bias

H_2 dissociation curve



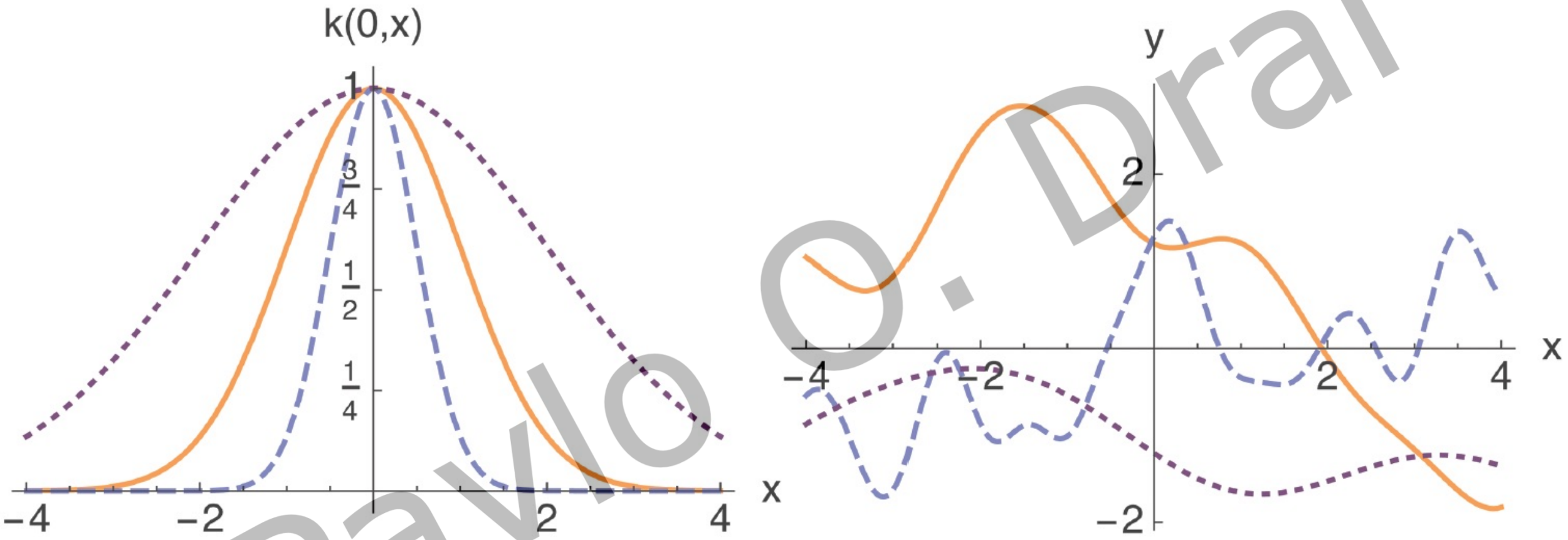
Overfitting
High variance
Low bias



$$E = \sum_{i=1}^{N_{tr}=20} \alpha_i \exp\left(-\frac{(R_i - R)^2}{2\sigma^2}\right)$$

Max Pinheiro Jr, P. O. Dral, Kernel methods. In *Quantum Chemistry in the Age of Machine Learning*, P. O. Dral, Ed. Elsevier: **2023**, Paperback ISBN: 9780323900492

KRR with Gaussian kernel



(b) Gaussian kernel with $\sigma = 0.5, 1, 2$ (dashed, solid, dotted lines).

Figure from: M. Rupp. *Int. J. Quantum Chem.* **2015,** *115*, 1058

KRR with Gaussian kernel: H_2

Take KRR with Gaussian kernel

$$f(\mathbf{x}') = \sum_{i=1}^{N_{tr}} \alpha_i \exp\left(-\frac{1}{2\sigma^2}\sum_{s}^{N_x}\left(x_{i,s} - x'_s\right)^2\right)$$

and consider what happens for very small $\sigma \to 0$:

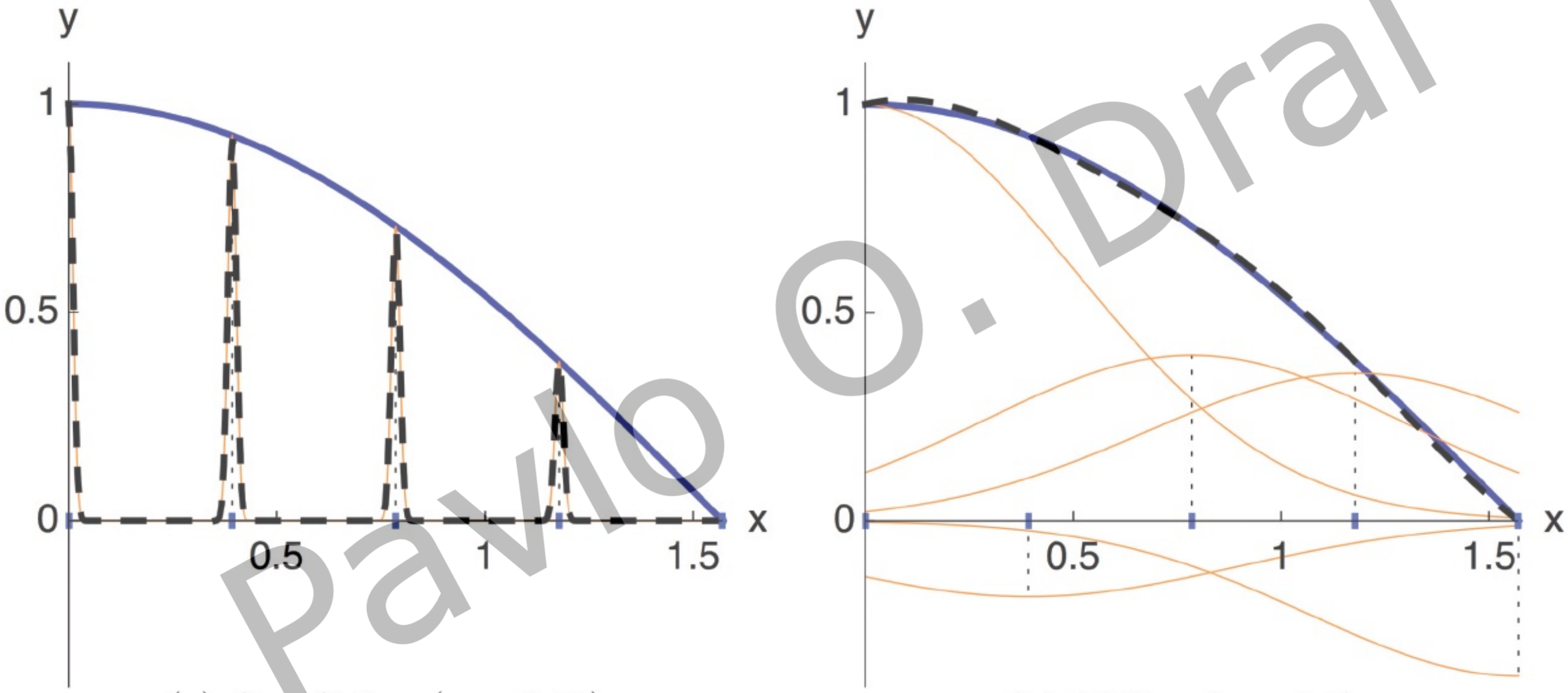
$$f(\mathbf{x}') = \begin{cases} \alpha_i, \text{ for } \mathbf{x}' = \mathbf{x}_i \\ 0, \text{ for } \mathbf{x}' \neq \mathbf{x}_i \end{cases}$$

and consider what happens for very large $\sigma \to \infty$:

$$f(\mathbf{x}') = \sum_{i=1}^{N_{tr}} \alpha_i = const$$

P. O. Dral, Quantum Chemistry Assisted by Machine Learning. In *Advances in Quantum Chemistry: Chemical Physics and Quantum Chemistry Volume 81*, 1st ed.; Brandas, E.; Ruud, K., Eds. Academic Press: 2020; Vol. 81. **Online tutorial:** MLatom.com/AQCtutorial/

KRR with Gaussian kernel

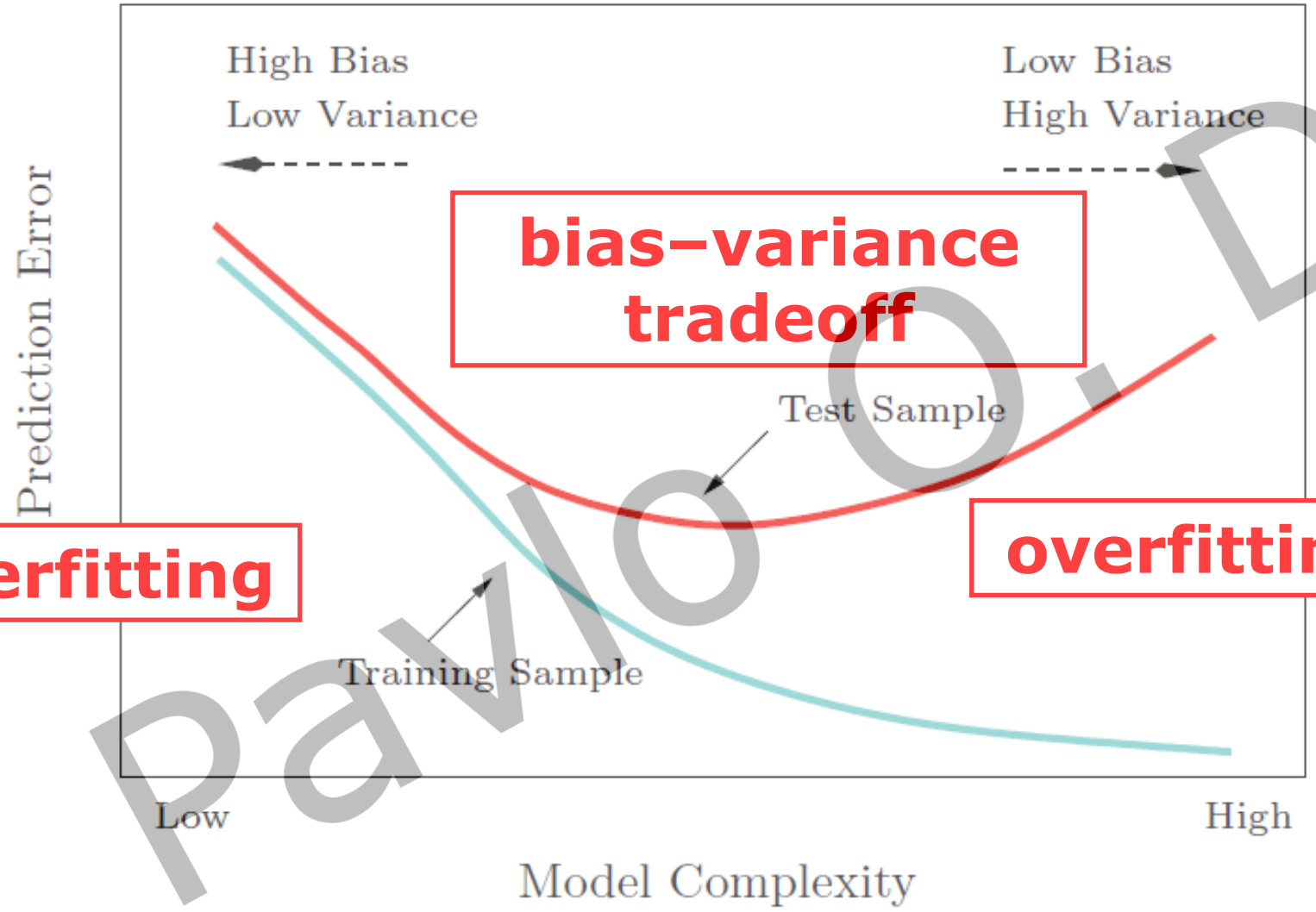


(a) Overfitting ($\sigma = 0.01$)

(b) Fitting ($\sigma = 0.5$)

Figure from: M. Rupp. *Int. J. Quantum Chem.* **2015,** *115*, 1058

Model Selection



bias–variance tradeoff

underfitting

overfitting

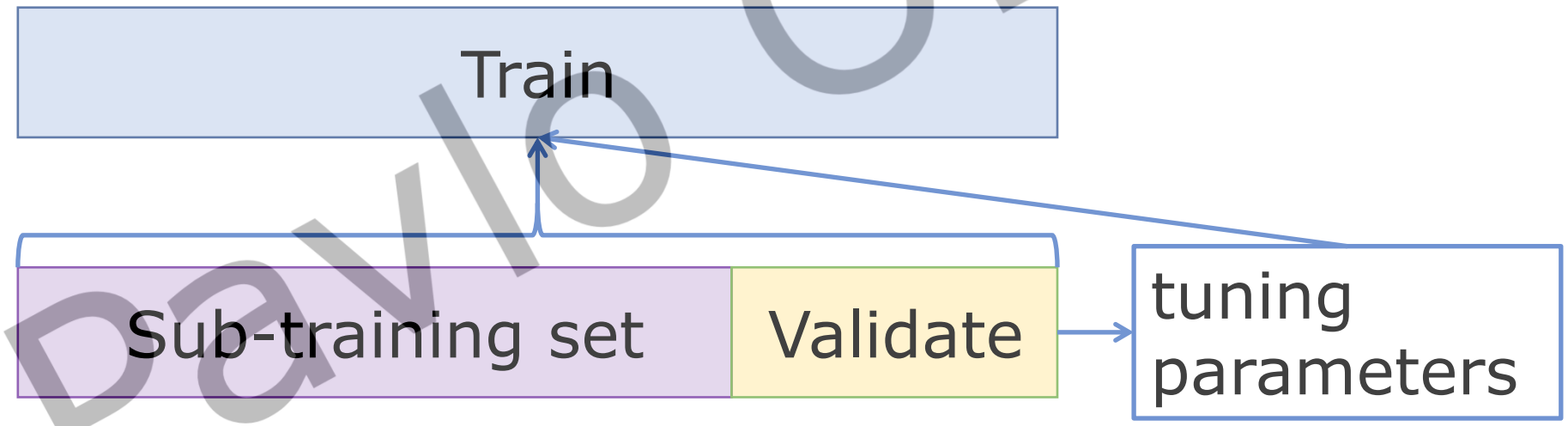
Hastie, Tibshirani, Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed.; Springer-Verlag, **2009**

Model selection (hyperparameter tuning)

Q: How to choose hyperparameters?

Model selection (hyperparameter tuning)

We target minimal error **not** in the training set, but in the validation set for models trained on the sub-training set.



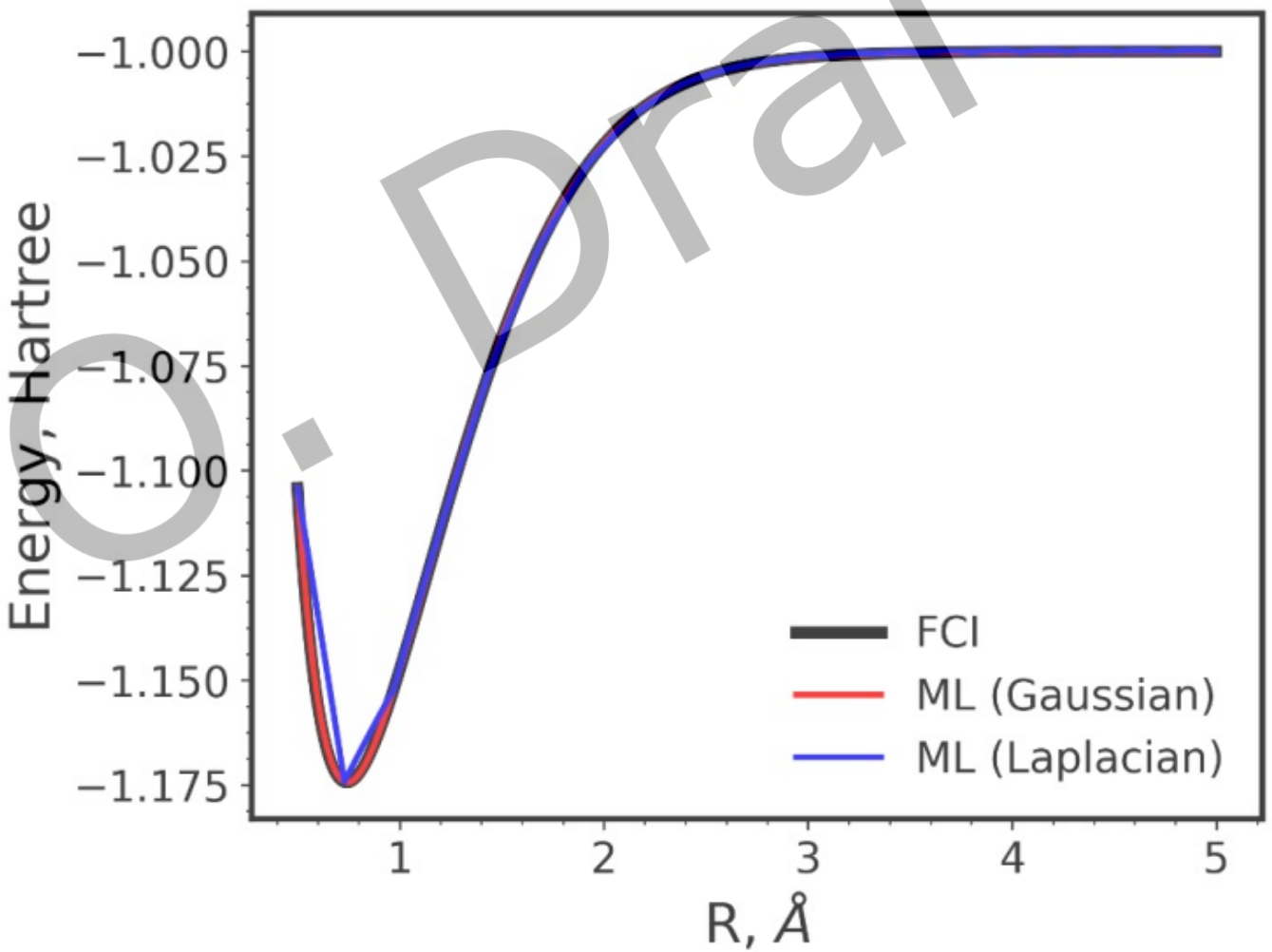
Hastie, Tibshirani, Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed.; Springer-Verlag, **2009**

Kernel-based model with Gaussian kernel

H_2 dissociation curve

Full CI calculations:
more than 30 min
for one value of R.

ML trained on 20 points needs
less than 1 sec.
for hundreds of other points

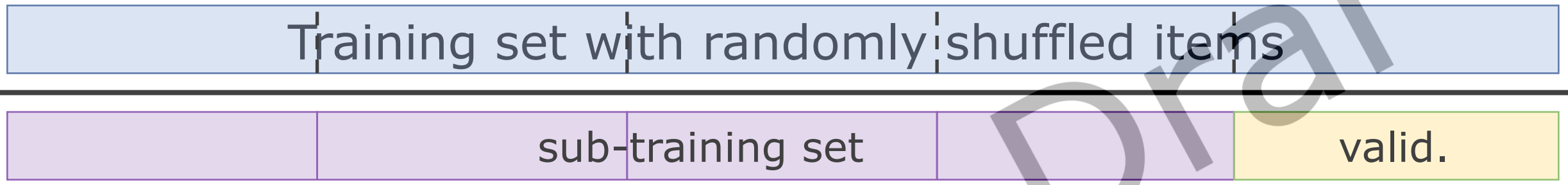


P. O. Dral, Quantum Chemistry Assisted by Machine Learning. In *Advances in Quantum Chemistry: Chemical Physics and Quantum Chemistry Volume 81*, 1st ed.; Brandas, E.; Ruud, K., Eds. Academic Press: 2020; Vol. 81. **Online tutorial:** MLatom.com/AQCtutorial/

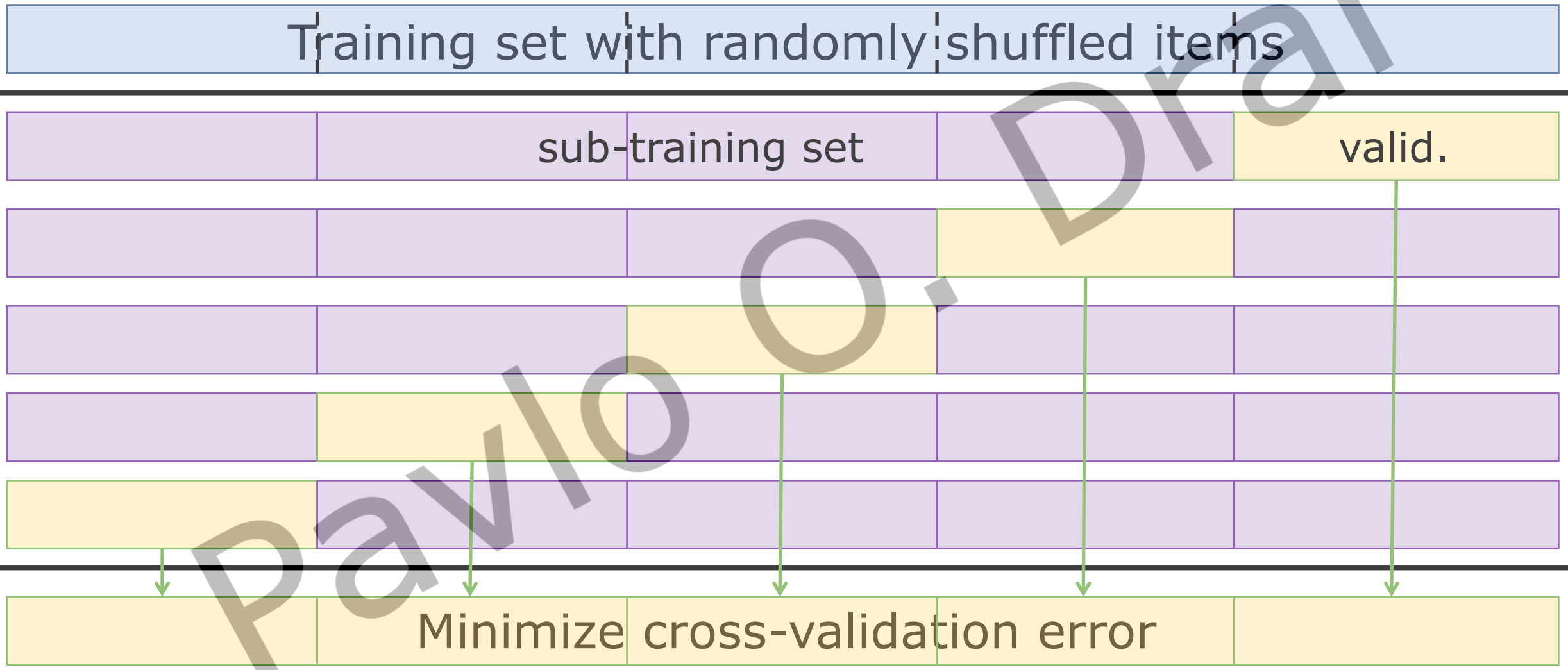
5-fold Cross-validation

Training set with randomly shuffled items

5-fold Cross-validation



5-fold Cross-validation



Model Selection

Random sampling for model selection is not always a good idea

Sometimes, **stratification** is preferable

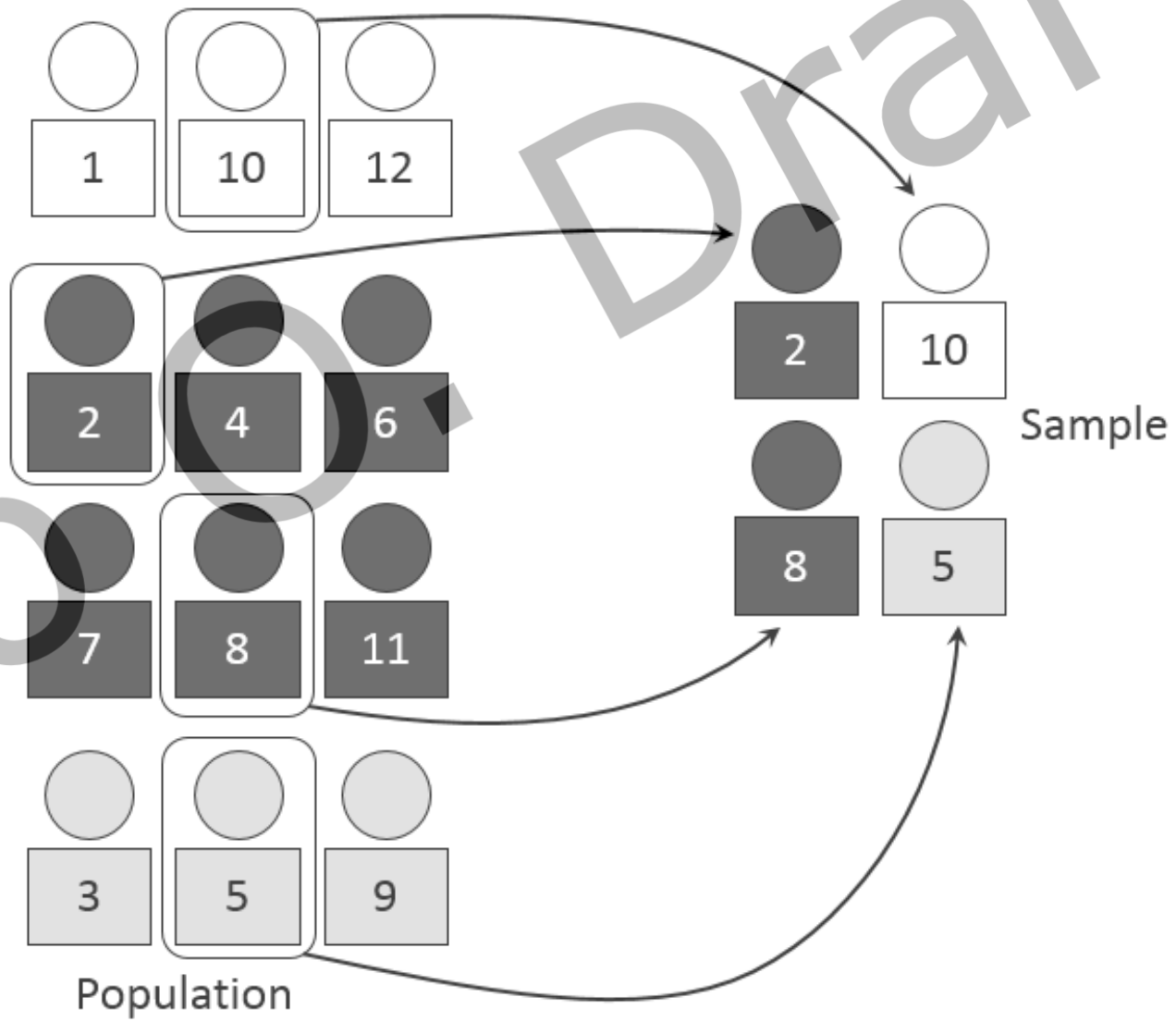


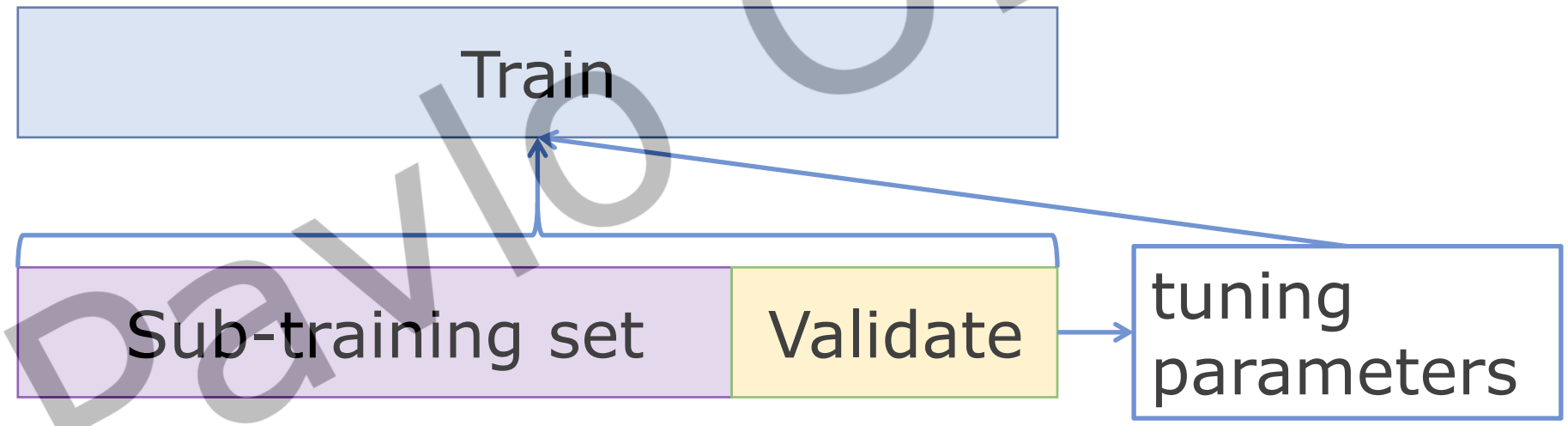
Figure by Dan Kernler [CC BY-SA 4.0], from Wikimedia Commons

Model Evaluation

(estimation of the generalization error)

ML: Error Estimation

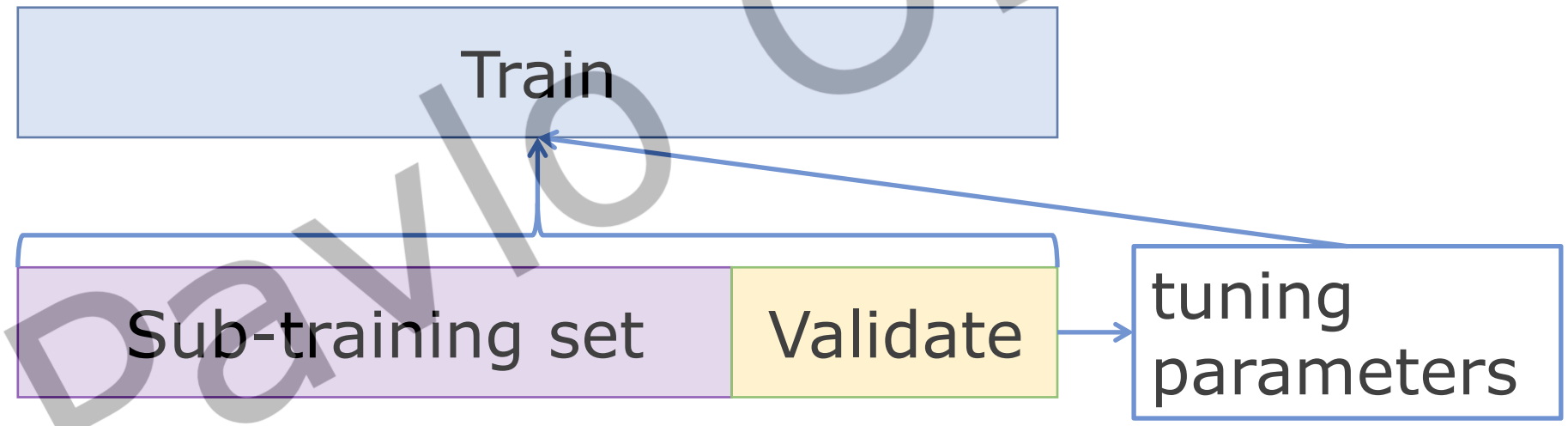
- Often ML error for its own training set is close to zero



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

ML: Error Estimation

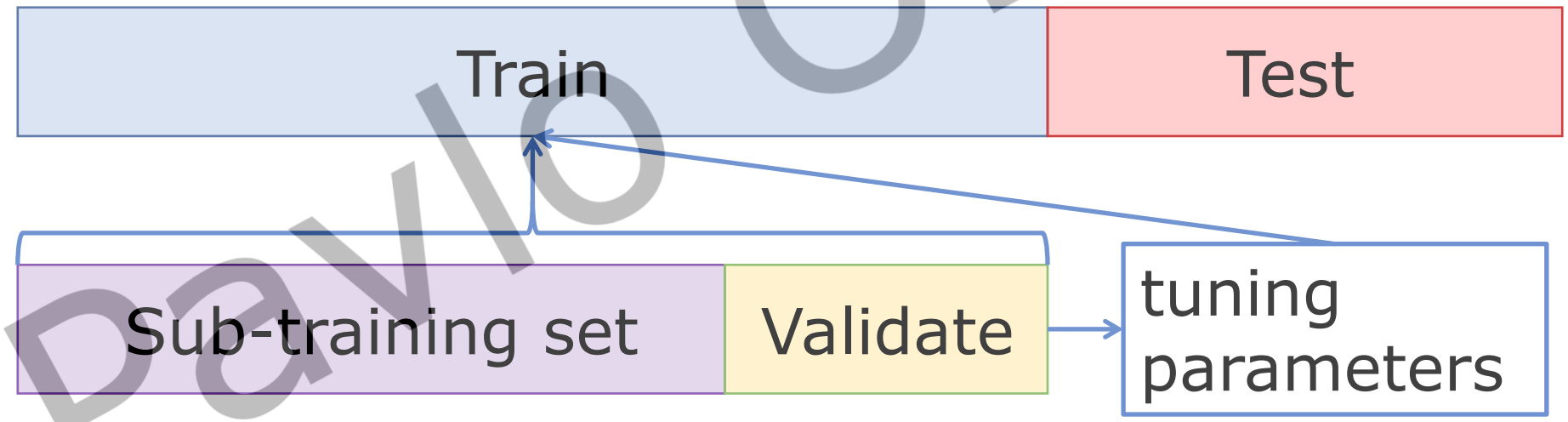
- Often ML error for its own training set is close to zero
- Using errors in the validation set would be also incorrect, because their minimization is a part of the training process



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

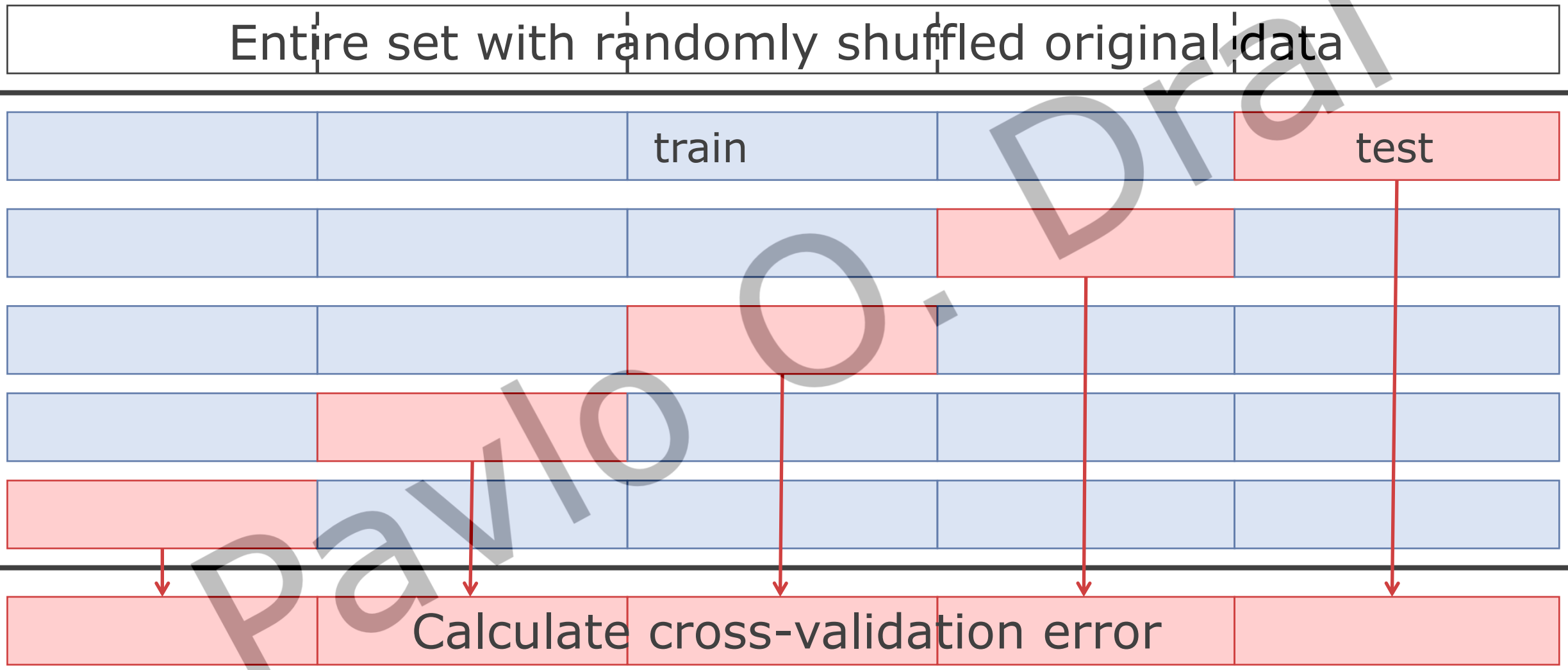
ML: Error Estimation

- 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**

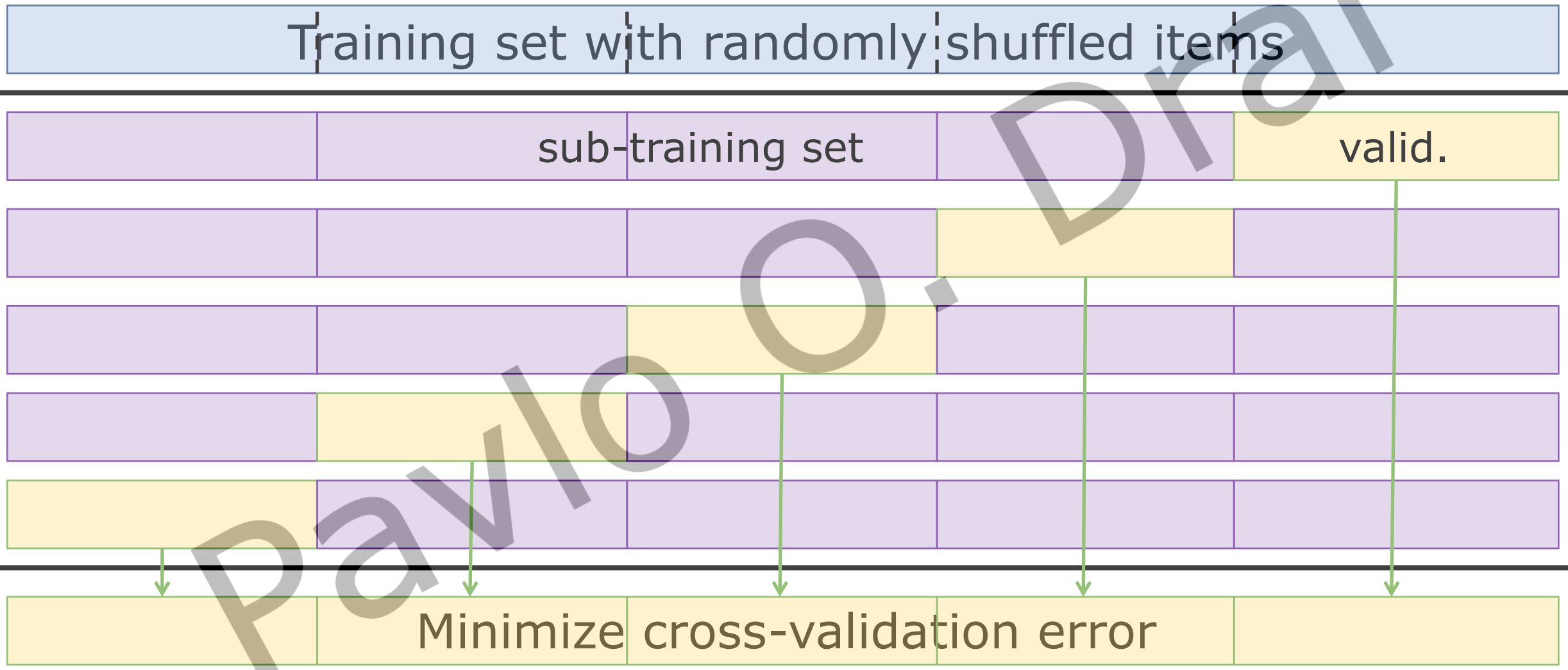


Hastie, Tibshirani, Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed.; Springer-Verlag, **2009**

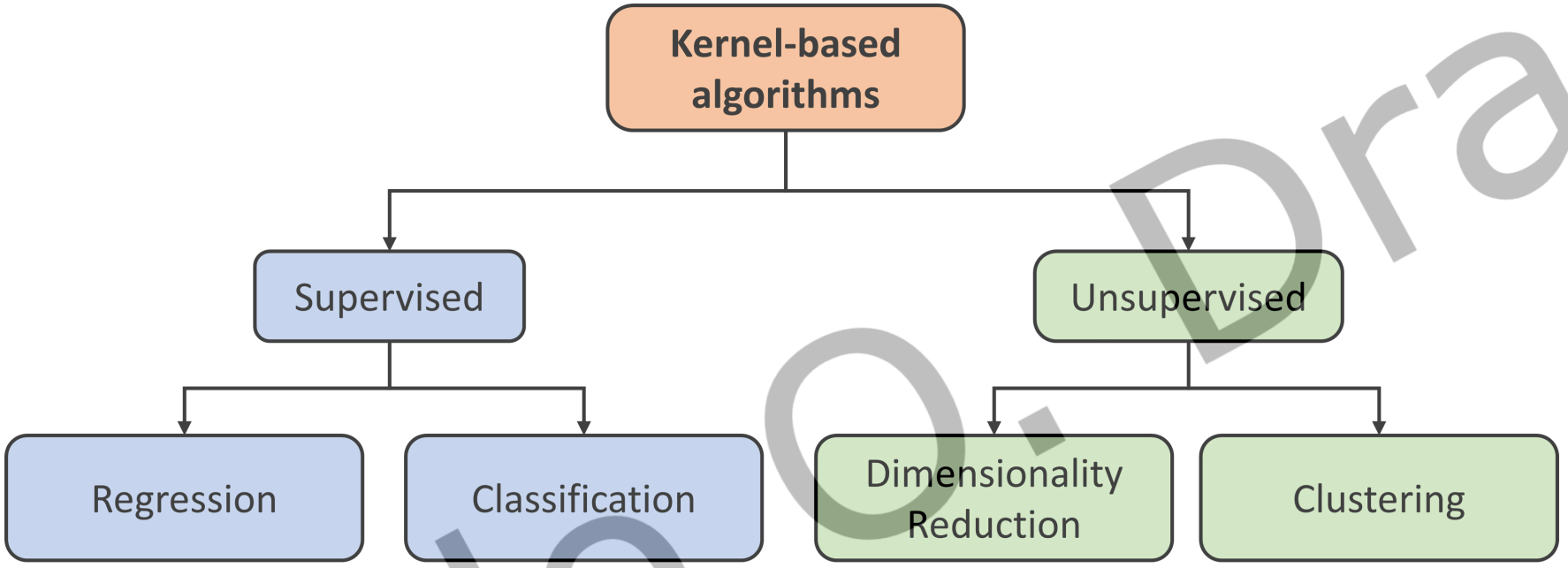
5-fold Cross-validation



5-fold Cross-validation



Family of kernel methods



- Kernel Ridge Regression
- Support Vector Machine (SVM)
- Gaussian Process (GP)

- Kernel Discriminant Analysis
- Kernel KNN
- Kernel logistic regression

- Kernel Principal Component Analysis (KPCA)
- Kernel Canonical Correlation Analysis (KCCA)

- K-means
- Spectral clustering
- kernel self-organizing map

Max Pinheiro Jr, P. O. Dral, Kernel methods. In *Quantum Chemistry in the Age of Machine Learning*, P. O. Dral, Ed. Elsevier: **2023**, Paperback ISBN: 9780323900492

KRR and other kernel methods

- Kernel ridge regression (KRR)

$$f(\mathbf{x}') = \sum_{i=1}^{N_{tr}} \alpha_i k(\mathbf{x}_i, \mathbf{x}')$$

- Gaussian processes (GP, kriging)

$$f(\mathbf{x}') = \sum_{i=1}^{N_{tr}} \alpha_i k(\mathbf{x}_i, \mathbf{x}')$$

Prediction functions are the same for KRR and GP!

- Support vector machines (SVM)

$$f(\mathbf{x}') = \sum_{i=1}^{N_{tr}} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}'),\ 0 < \alpha_i < C$$

Max Pinheiro Jr, P. O. Dral, Kernel methods. In *Quantum Chemistry in the Age of Machine Learning*, P. O. Dral, Ed. Elsevier: **2023**, Paperback ISBN: 9780323900492

KRR and other kernel methods

- Kernel ridge regression gives the same prediction as Gaussian processes:

$$f(\mathbf{x}') = \sum_{i=1}^{N_{tr}} \alpha_i k(\mathbf{x}_i, \mathbf{x}')$$

- Gaussian processes also provide:
  - variance V

$$V(\mathbf{x}') = k(\mathbf{x}', \mathbf{x}') - \mathbf{k}'^T (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{k}'$$

$$\mathbf{k}' = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}') \\ \vdots \\ k\left(\mathbf{x}_{N_{\mathrm{tr}}}, \mathbf{x}'\right) \end{pmatrix}$$

  - Marginal likelihood:

$$\log p(\mathrm{y}|X) = -\frac{1}{2}\mathbf{y}^T \boldsymbol{\alpha} - \frac{1}{2}\log|\mathbf{K} + \lambda \mathbf{I}| - \frac{N_{tr}}{2}\log 2\pi$$

Hyperparameters in kernel function can be found by optimizing log marginal likelihood, for which derivatives are taken, e.g. $\frac{\partial \log p(\mathrm{y}|X, \sigma)}{\partial \sigma}$

Rasmussen, Williams, *Gaussian Processes for Machine Learning*. The MIT Press: Boston, **2006**

Pros & cons of kernel methods

Advantages of kernel methods:

- Nonparametric models, i.e., do not assume a specific behavior of data (compare to parametric model such as linear regression)
- Explicitly incorporate training data, thus very flexible and accurate
- Closed (analytical) solution, i.e. fast training

$$\boldsymbol{\alpha} = (\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{y} \qquad f(\mathbf{x}') = \sum_{i=1}^{N_{tr}} \alpha_i k(\mathbf{x}_i, \mathbf{x}')$$

Disadvantages:

- Slow training for lots of training data (scales as $O(N_{tr}^3)$)
- Requires lots of RAM to store the kernel matrix (scales as $O(N_{tr}^2)$)
- Prediction time slows down with more training data (scales as $O(N_{tr}^1)$)

P. O. Dral, Quantum Chemistry Assisted by Machine Learning. In *Advances in Quantum Chemistry: Chemical Physics and Quantum Chemistry Volume 81*, 1st ed.; Brandas, E.; Ruud, K., Eds. Academic Press: 2020; Vol. 81. **Online tutorial:** MLatom.com/AQCtutorial/

Pros & cons of kernel methods

Table 1 Required memory (RAM) for storing kernel matrix built with increasing number of training points.

Training set size	RAM size
$100 = 10^2$	78 kB
$1000 = 10^3$	7.6 MB
$10{,}000 = 10^4$	0.75 GB
$50{,}000 = 5 \times 10^4$	19 GB
$100{,}000 = 10^5$	75 GB
$500{,}000 = 5 \times 10^5$	1.8 TB
$1000{,}000 = 10^6$	7.3 TB

Table 2 CPU time needed for calculating regression coefficients for increasing number of training points, assuming that it takes 10 s for 10,000 training points.

Training set size	Time
$100 = 10^2$	0.01 milliseconds
$1000 = 10^3$	0.01 s
$10{,}000 = 10^4$	10 s
$50{,}000 = 5 \times 10^4$	21 min
$100{,}000 = 10^5$	2.8 h
$500{,}000 = 5 \times 10^5$	15 days
$1000{,}000 = 10^6$	3.9 months

P. O. Dral, Quantum Chemistry Assisted by Machine Learning. In *Advances in Quantum Chemistry: Chemical Physics and Quantum Chemistry Volume 81*, 1st ed.; Brandas, E.; Ruud, K., Eds. Academic Press: 2020; Vol. 81. **Online tutorial:** MLatom.com/AQCtutorial/

Kernel Ridge Regression

Solutions:

- Reduce the training set by selecting the most relevant points[1,2]
- Sparsification techniques[3]
- Construct high-dimensional kernels as products of one-dimensional kernels[4]

See, for example:
[1] Dral, Owens, Yurchenko, Thiel, *J. Chem. Phys.* **2017**, *146*, 244108
[2] Hu, Xie, Li, Li, Lan, *J. Phys. Chem. Lett.* **2018**, *9*, 2725
[3] Bartók, Csányi, *Int. J. Quantum Chem.* **2015**, *115*, 1051
[4] Unke, Meuwly, *J. Chem. Inf. Model.* **2017**, *57*, 1923

Neural networks

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)

Neural networks (NN)

Quantum Chemistry in the Age of Machine Learning. Ed. P. O. Dral. Elsevier: Amsterdam, Netherlands, **2023**.

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)

Neural networks (NN)

Quantum Chemistry in the Age of Machine Learning. Ed. P. O. Dral. Elsevier: Amsterdam, Netherlands, **2023**.

ML algorithms

Linear regression

$$\hat{y} = f(\mathbf{x}; \mathbf{w}, b) = b + w_1 x_1 + w_2 x_2 + \cdots + w_p x_p = \mathbf{x}^T \mathbf{w} + b$$

Neural networks (NNs): the single hidden layer, feed-forward network

$$\hat{y} = f(\mathbf{x}; \boldsymbol{\alpha}, \mathbf{a}, \mathbf{w}, b) = b + w_1 h_1(\mathbf{x}; \boldsymbol{\alpha}_1, a_1) + \cdots + w_M h_M(\mathbf{x}; \boldsymbol{\alpha}_M, a_M) = \mathbf{h}^T \mathbf{w} + b$$

$$h_m(\mathbf{x}; \boldsymbol{\alpha}_m, a_m) = g\left(a_m + \alpha_{m1} x_1 + \alpha_{m2} x_2 + \cdots + \alpha_{mp} x_p\right) = g(\mathbf{x}^T \boldsymbol{\alpha}_m + a_m)$$

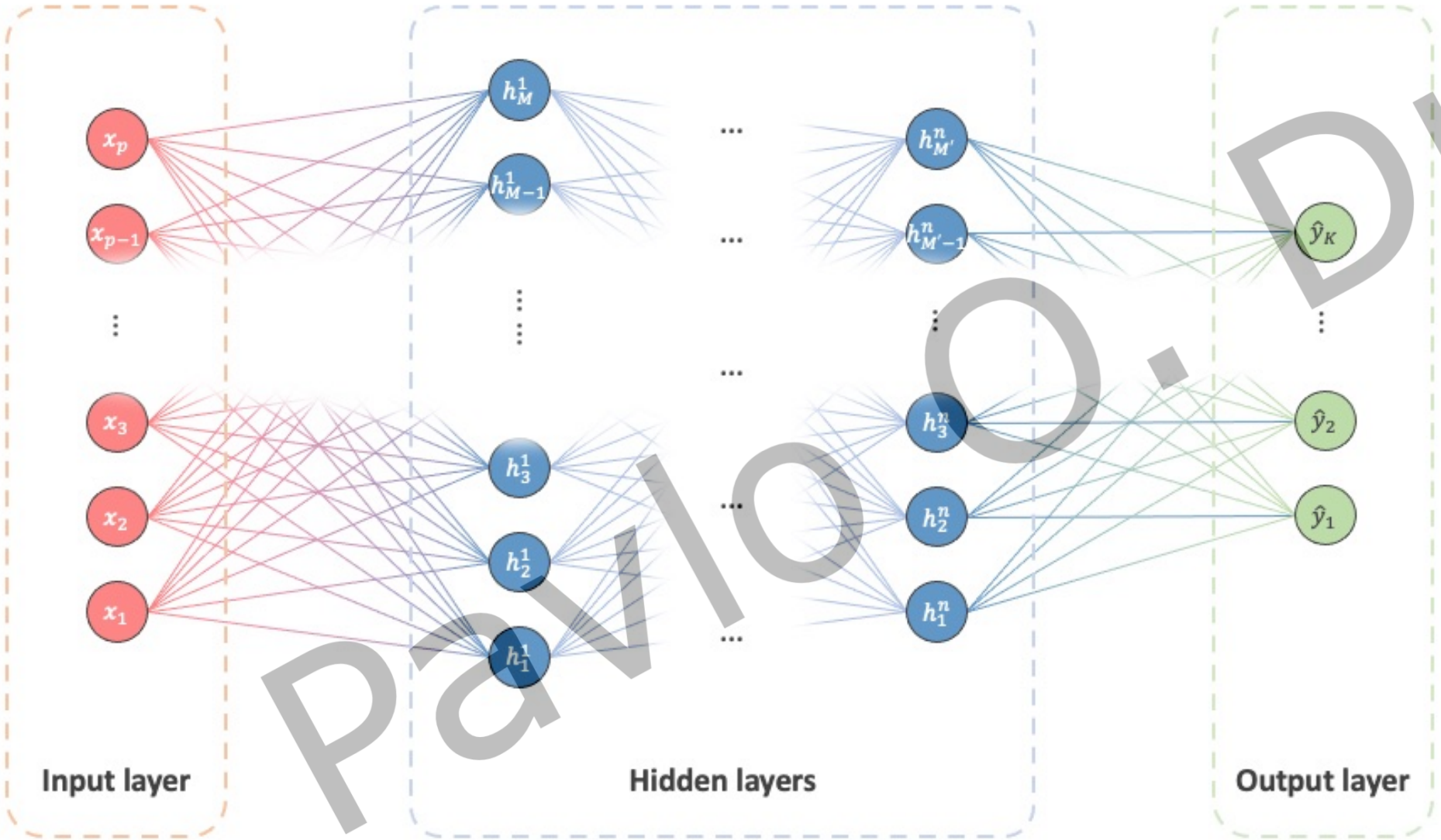
$$\hat{y} = f(\mathbf{x}; \boldsymbol{\alpha}, \mathbf{a}, \mathbf{w}, b) = f^{(2)}(\mathbf{h}; \mathbf{w}, b) = f^{(2)}\left(f^{(1)}(\mathbf{x})\right)$$

Activation functions:

$$g(v) = \exp(-a(v - c)^2)$$ radial basis function (RBF)

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.

Neural network (NN)



NN consists of
- layers
- nodes=units=neurons

The single hidden layer, feed-forward NN

$\mathbf{w}, \boldsymbol{\alpha}$ ... weights

$\mathbf{a}$, b... biases

$$f(\mathbf{x}; \boldsymbol{\alpha}, \mathbf{a}, \mathbf{w}, b) = f^{(2)}(\mathbf{h}; \mathbf{w}, b)$$

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.

ML algorithms

Linear regression

$$\hat{y} = f(\mathbf{x}; \mathbf{w}, b) = b + w_1 x_1 + w_2 x_2 + \cdots + w_p x_p = \mathbf{x}^T \mathbf{w} + b$$

Neural networks (NNs): the single hidden layer, feed-forward network

$$\hat{y} = f(\mathbf{x}; \boldsymbol{\alpha}, \mathbf{a}, \mathbf{w}, b) = b + w_1 h_1(\mathbf{x}; \boldsymbol{\alpha}_1, a_1) + \cdots + w_M h_M(\mathbf{x}; \boldsymbol{\alpha}_M, a_M) = \mathbf{h}^T \mathbf{w} + b$$

$$h_m(\mathbf{x}; \boldsymbol{\alpha}_m, a_m) = g\left(a_m + \alpha_{m1} x_1 + \alpha_{m2} x_2 + \cdots + \alpha_{mp} x_p\right) = g(\mathbf{x}^T \boldsymbol{\alpha}_m + a_m)$$

g is the activation function. If:

- g is the identity function, NN is equivalent to linear regression $g(v) = v$

$$g\left(a_m + \alpha_{m1} x_1 + \alpha_{m2} x_2 + \cdots + \alpha_{mp} x_p\right) = a_m + \alpha_{m1} x_1 + \alpha_{m2} x_2 + \cdots + \alpha_{mp} x_p$$

- Typically, g is used for the nonlinear transformation making the NN flexible

$g(v) = \exp(-a(v - c)^2)$ radial basis function (RBF)

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.

Table 1. Overview of a selection of popular activation functions.

Names	Equation
linear function identity function[2]	$g(v) = v$
rectified linear unit (ReLU) [2]	$g(v) = \max(0, v)$
exponential linear unit (ELU)[5]	$g(v) = \begin{cases} v & \text{if } v \geq 0 \\ a(\exp(v) - 1) & \text{otherwise} \end{cases}$ where a is a parameter
continuously differentiable exponential linear unit (CELU)[6]	$g(v) = \begin{cases} v & \text{if } v \geq 0 \\ a\left(\exp\left(\frac{v}{a}\right) - 1\right) & \text{otherwise} \end{cases}$ where a is a parameter
Gaussian error linear unit (GELU)[7]	$g(v) = v \cdot \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{v}{\sqrt{2}}\right)\right]$ faster approximated versions: $g(v) = 0.5v\left(1 + \tanh\left[\sqrt{\frac{2}{\pi}}(v + 0.044715v^3)\right]\right)$ $g(v) = v \cdot \sigma(1.702v) = v \cdot \frac{1}{1 + \exp(-1.702v)}$

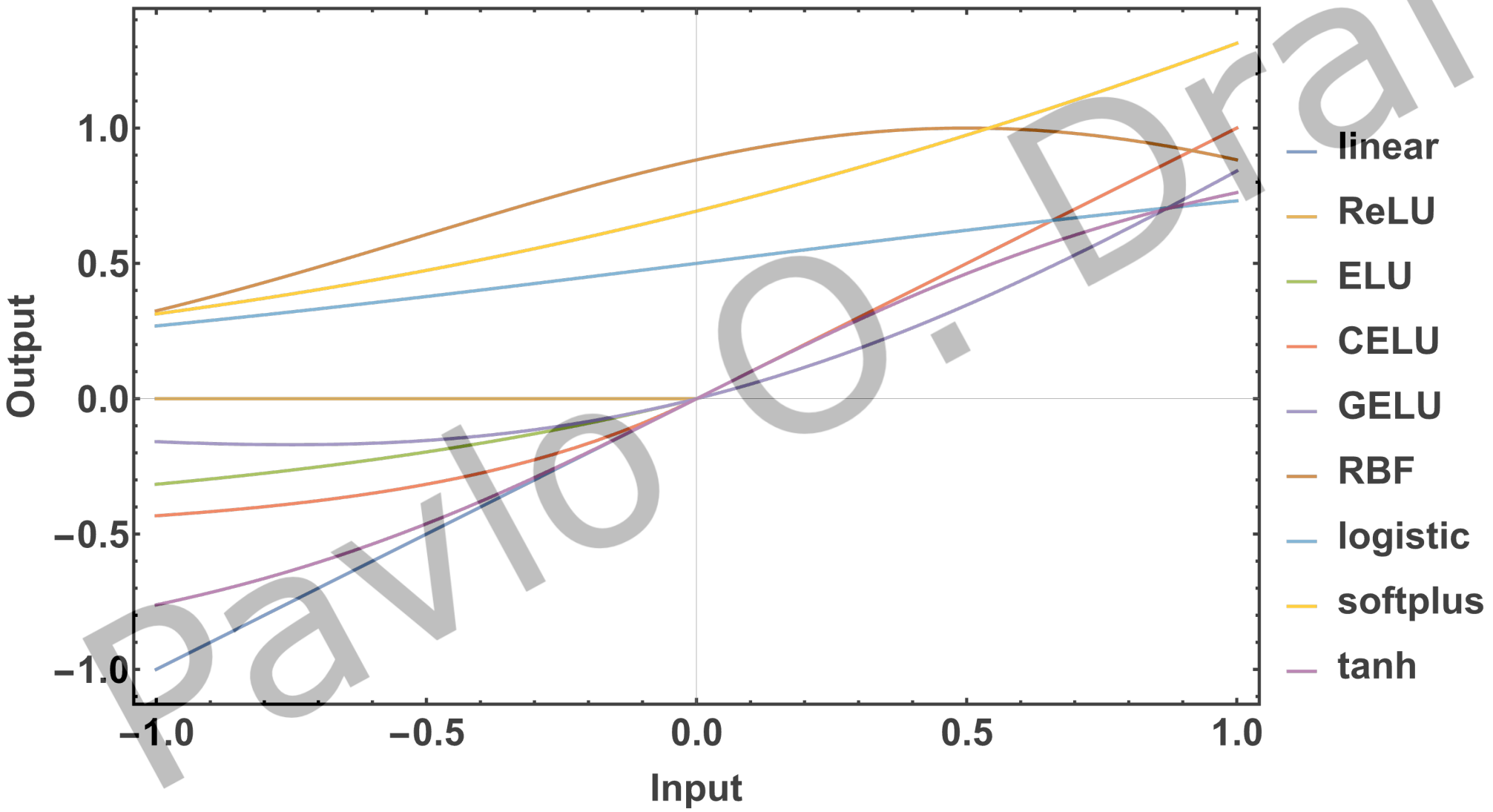
P. O. Dral, A. Kananenka, F. Ge, B.-X. Xue, Neural Networks. In *Quantum Chemistry in the Age of Machine Learning*, 1st ed.; P. O. Dral, Ed. Elsevier: 2023.

Activation functions

radial basis function (RBF)[1-2]	$g(v) = \exp(-a(v-c)^2)$ where a and c are parameters
logistic sigmoid function[1-2]	$g(v) = \sigma(v) = \frac{1}{1+\exp(-v)}$
softplus function[2]	$g(v) = \log(1+\exp(v))$
hyperbolic tangent function[2]	$g(v) = T(v) = \tanh(v)$

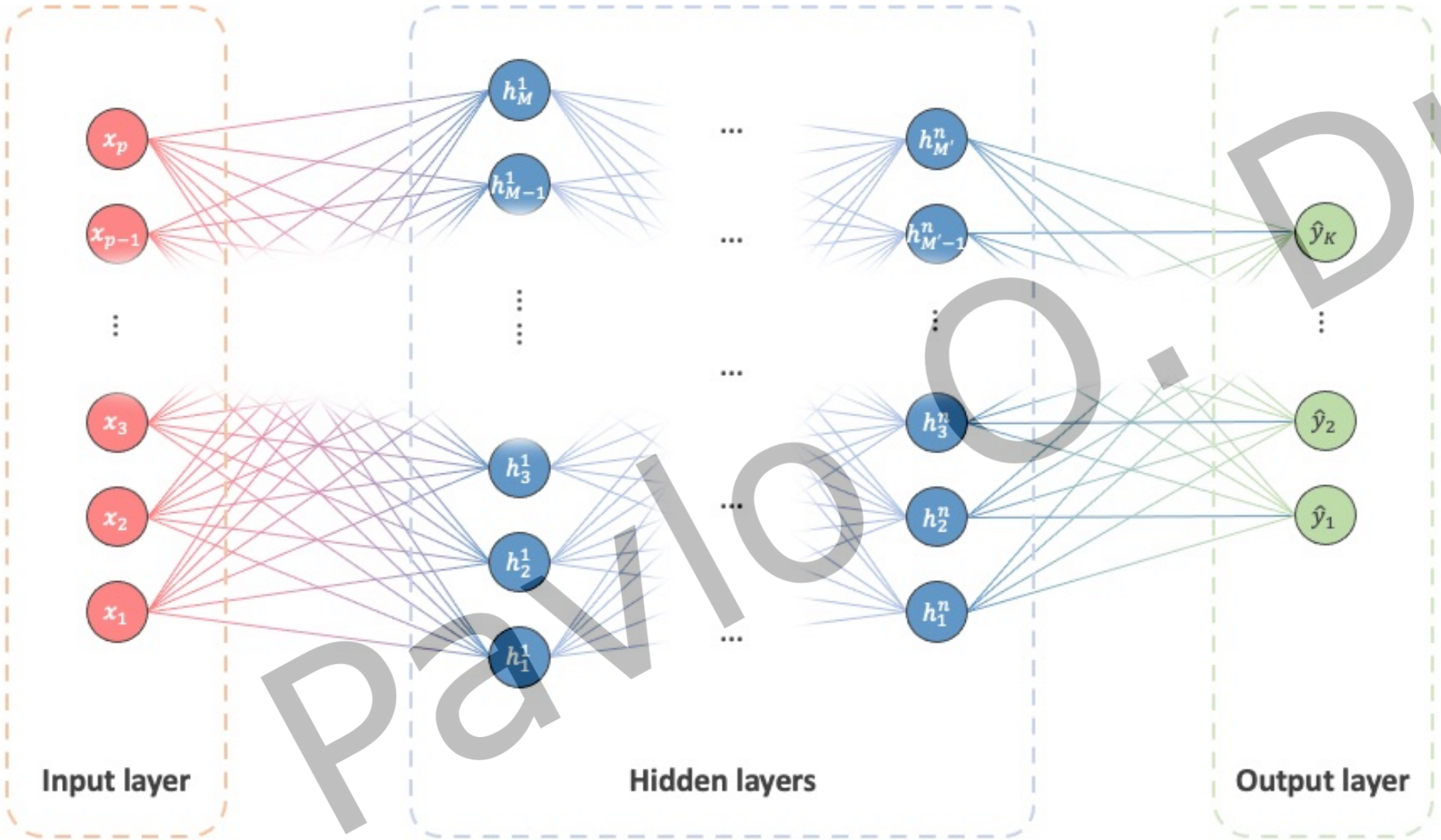
P. O. Dral, A. Kananenka, F. Ge, B.-X. Xue, Neural Networks. In *Quantum Chemistry in the Age of Machine Learning*, 1st ed.; P. O. Dral, Ed. Elsevier: 2023.

Activation functions



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.

Neural network (NN)



NN consists of
- layers
- nodes=units=neurons

The single hidden layer, feed-forward NN

$\mathbf{w}, \boldsymbol{\alpha}$ ... weights

$\mathbf{a}$, b... biases

$$f(\mathbf{x}; \boldsymbol{\alpha}, \mathbf{a}, \mathbf{w}, b) = f^{(2)}(\mathbf{h}; \mathbf{w}, b)$$

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

To train NN means to find its weights $\boldsymbol{\theta}$ usually by solving this minimization task:

$$\arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N_{\mathrm{tr}}} (f(\mathbf{x}_i;\boldsymbol{\theta}) - y_i)^2$$

To avoid overfitting this solution can be regularized using weight decay approach (recall ridge regression and KRR):

$$\arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N_{tr}} (f(\mathbf{x}_i;\boldsymbol{\theta}) - y_i)^2 + \lambda \sum_{j=1}^{N_p} \theta_j^2$$

$$\boldsymbol{\theta} = \mathbf{w}, \boldsymbol{\alpha}, \mathbf{a}, b$$

parameters

$$\hat{y} = f(\mathbf{x};\boldsymbol{\alpha},\mathbf{a},\mathbf{w},b) = b + w_1 h_1(\mathbf{x};\boldsymbol{\alpha}_1,a_1) + \cdots + w_M h_M(\mathbf{x};\boldsymbol{\alpha}_M,a_M) = \mathbf{h}^T\mathbf{w} + b$$

$$h_m(\mathbf{x};\boldsymbol{\alpha}_m,a_m) = g\left(a_m + \alpha_{m1}x_1 + \alpha_{m2}x_2 + \cdots + \alpha_{mp}x_p\right) = g(\mathbf{x}^T\boldsymbol{\alpha}_m + a_m)$$

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

To train NN means to find its weights $\boldsymbol{\theta}$ usually by solving this minimization task:

$$\arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N_{\mathrm{tr}}} (f(\mathbf{x}_i;\boldsymbol{\theta}) - y_i)^2$$

To avoid overfitting this solution can be regularized using weight decay approach (recall ridge regression and KRR):

$$\arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N_{tr}} (f(\mathbf{x}_i;\boldsymbol{\theta}) - y_i)^2 + \lambda \sum_{j=1}^{N_p} \theta_j^2$$

$$\boldsymbol{\theta} = \mathbf{w}, \boldsymbol{\alpha}, \mathbf{a}, b$$

parameters

$$\hat{y} = f(\mathbf{x};\boldsymbol{\alpha},\mathbf{a},\mathbf{w},b) = b + w_1 h_1(\mathbf{x};\boldsymbol{\alpha}_1,a_1) + \cdots + w_M h_M(\mathbf{x};\boldsymbol{\alpha}_M,a_M) = \mathbf{h}^T\mathbf{w} + b$$

$$h_m(\mathbf{x};\boldsymbol{\alpha}_m,a_m) = g\left(a_m + \alpha_{m1}x_1 + \alpha_{m2}x_2 + \cdots + \alpha_{mp}x_p\right) = g(\mathbf{x}^T\boldsymbol{\alpha}_m + a_m)$$

In contrast to linear regression and kernel methods, closed solution is unknown

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

Issues with NNs:

In contrast to linear regression and kernel methods, no closed solution exists

Solutions are unstable and difficult to find.

Computationally expensive optimization problem should be solved and it therefore often can be speed up by using GPUs instead of CPUs.

GPUs are however much more expensive and difficult to get and optimization is still quite slow.

One of the popular approaches for fitting is back-propagation.

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

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

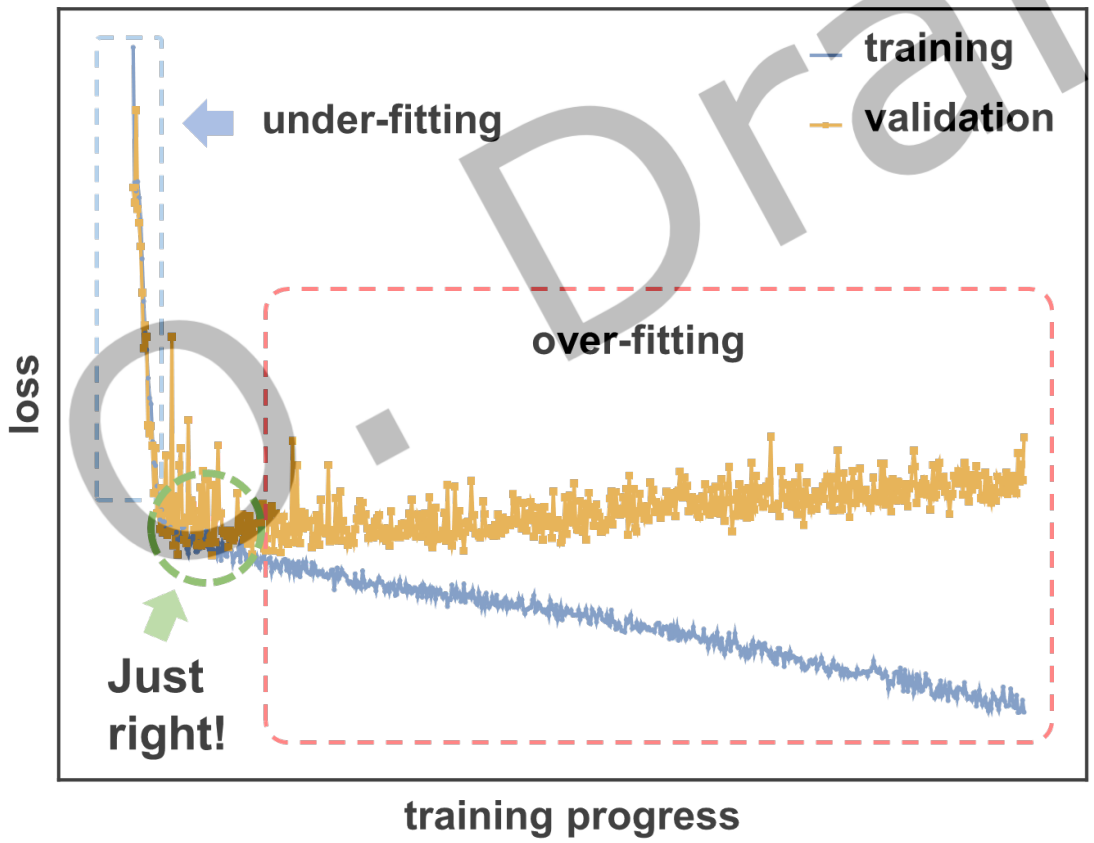
Issues with NNs:

Overfitting, regularization methods to deal with it:

- weight decay

$$\arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N_{tr}} (f(\mathbf{x}_i;\boldsymbol{\theta}) - y_i)^2 + \lambda \sum_{j=1}^{N_p} \theta_j^2$$

- early stopping
- data augmentation



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

Issues with NNs:

- Input values should be scaled, usually standardized to center the inputs and scale them so that their standard deviation is 1 (Z-score normalization)
- It is also important to center reference data
- Number of hidden layers and units should be adjusted often by manual experimentation

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

Issues with NNs:

- Initial guess of weights strongly influences the final parameter values
- Starting with zero values prevents back-propagation algorithm to find better solutions
- Starting with too large values often leads to large generalization errors

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

Issues with NNs:

- Initial guess of weights strongly influences the final parameter values
- Starting with zero values prevents back-propagation algorithm to find better solutions
- Starting with too large values often leads to large generalization errors

Thus, one can get lot of 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 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.

Active learning with several NNs

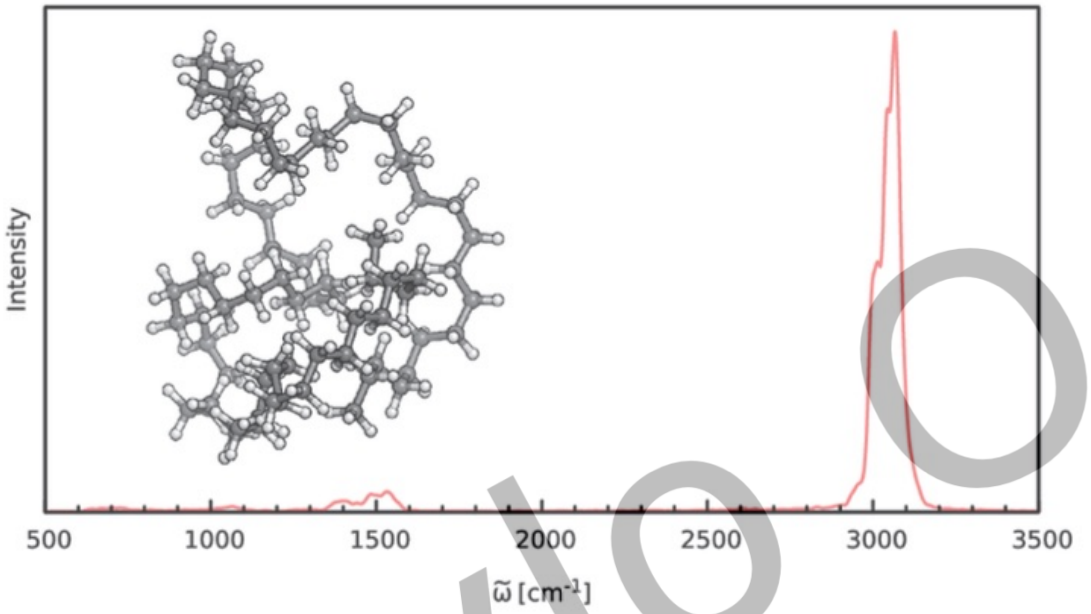
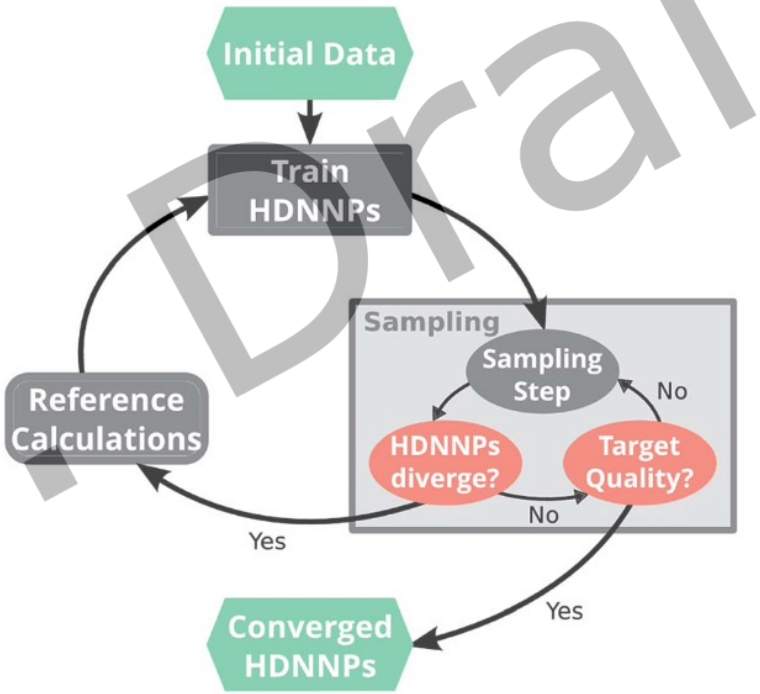


Fig. 6 IR spectrum of the $C_{69}H_{140}$ alkane as predicted by the ML model based on the B2PLYP method.



M. Gastegger, J. Behler, P. Marquetand, *Chem. Sci.* **2017**, *8*, 6924

Deep learning

Deep learning is based on neural networks (NN) with large depth (for feed-forward neural network – more than one hidden unit) in contrast to shallow neural network

Some of other types of neural networks:

- Convolutional networks
- Recurrent neural networks
- Autoencoders

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.

The NN zoo

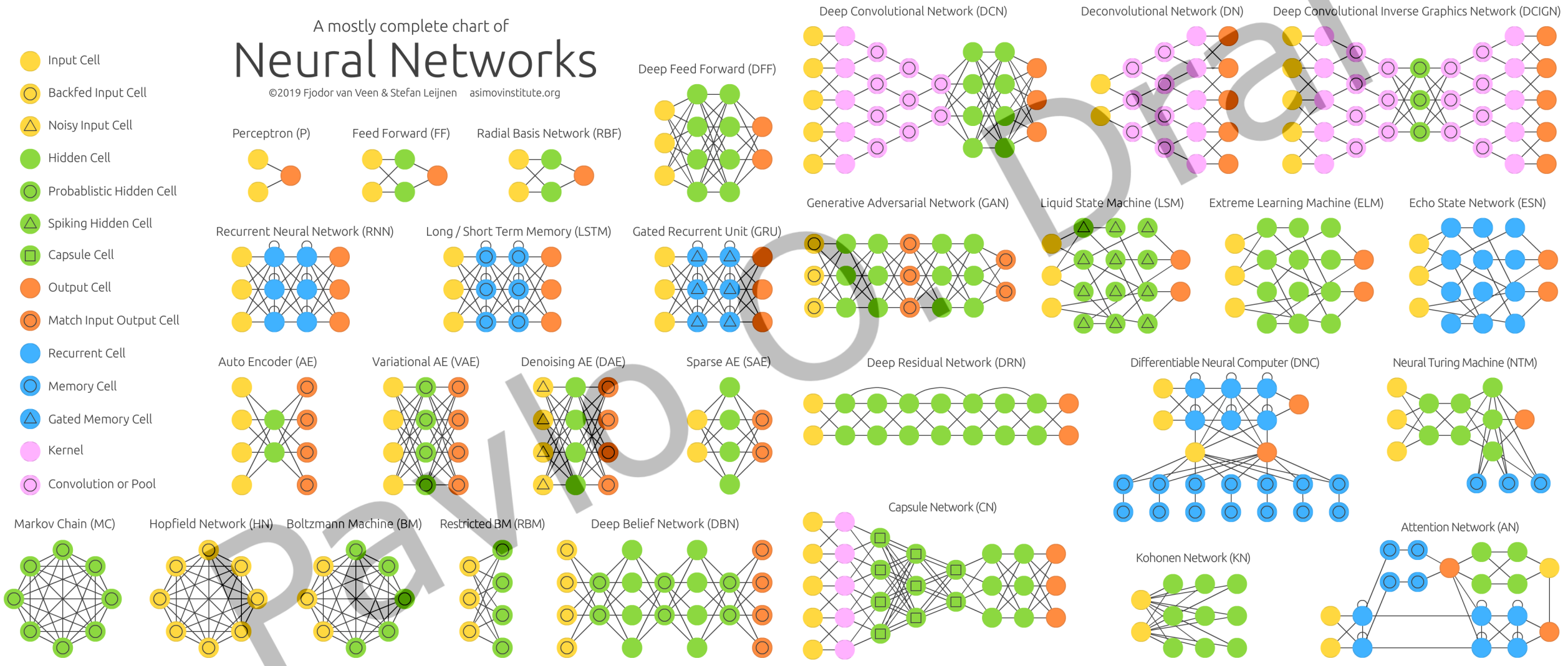


Figure from https://www.asimovinstitute.org/neural-network-zoo/

Parametric vs nonparametric algorithms

Comparison of KMs and NNs

What method to choose?
Kernel methods or neural networks?

Some of the ML Algorithms

f(x; **parameters**)

Linear regression

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

Number of parameters is fixed: parametric model

Neural networks are also parametric models

Kernel ridge regression (KRR)

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

Number of parameters depends on number of training points: nonparametric model, e.g. KRR

Some of the ML Algorithms

All have some advantages and disadvantages, but often provide results with similar accuracy.

In many cases **it is not possible to claim that one fitting method is better than another**. The **choice will depend on experience and taste**.[1]

[1] Manzhos, Dawes, Carrington, *Int. J. Quantum Chem.* **2015**, *115*, 1012

Some of the ML Algorithms

However: You should be aware of the **law of the hammer** and do not try to use a hammer for every problem only because you already have a hammer.

Source: Wikipedia

Categories of machine learning potentials

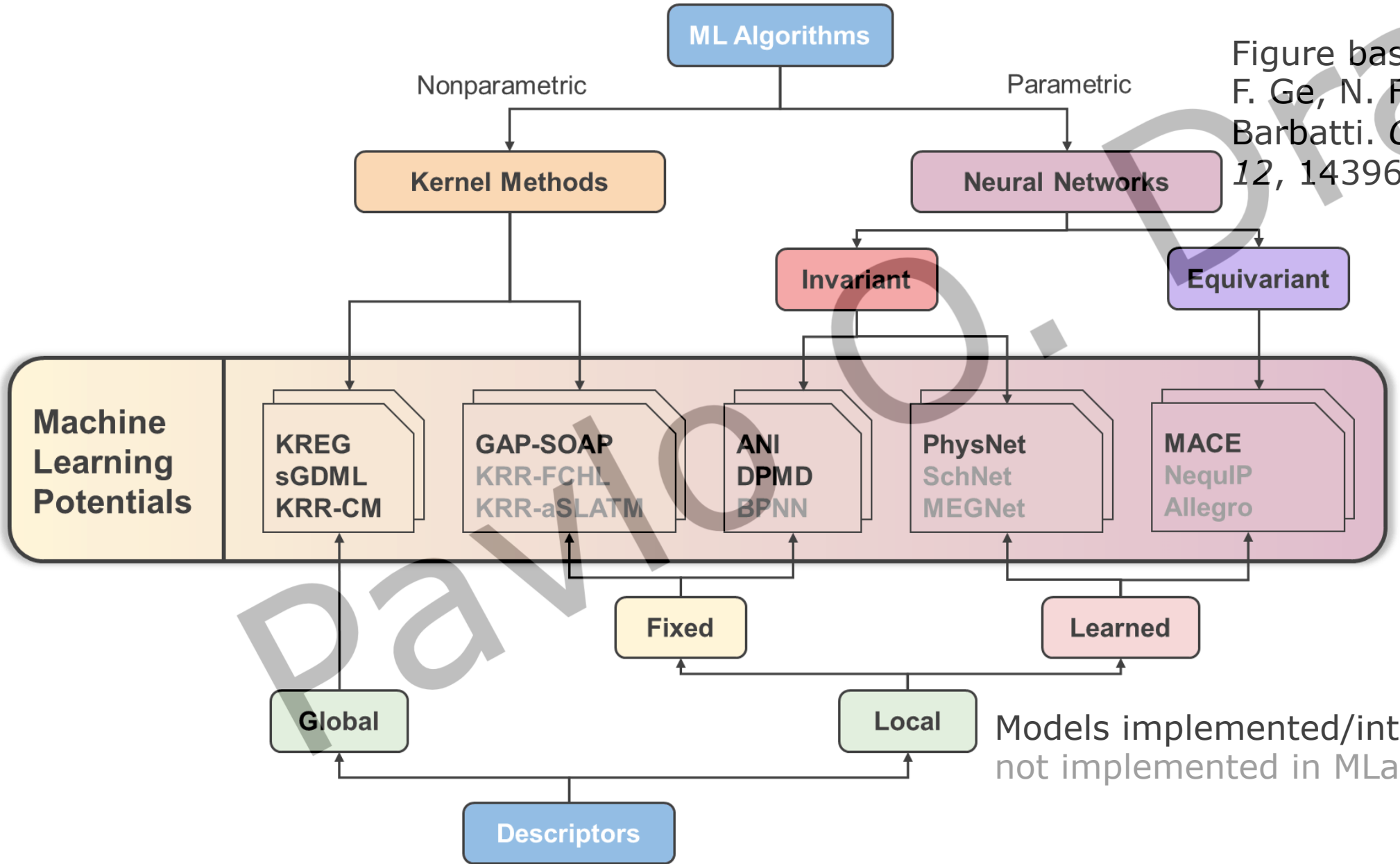


Figure based on M. Pinheiro Jr, F. Ge, N. Ferré, P. O. Dral, M. Barbatti. *Chem. Sci.* **2021**, *12*, 14396–14413

Models implemented/interfaced in MLatom
not implemented in MLatom

Supervised Machine Learning

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

Input (x) → *f*(x) → Output (y)

Given collection of known {x,y} find a function *f*(x)

training set train ML model

Use this function for making new predictions given just {x'}

- Data
- **Choice of x (descriptor)**
- Choice of y (labels)
- Fitting function (ML algorithm, ML model)
- Optimization of ML model parameters

The KREG model

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

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

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

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

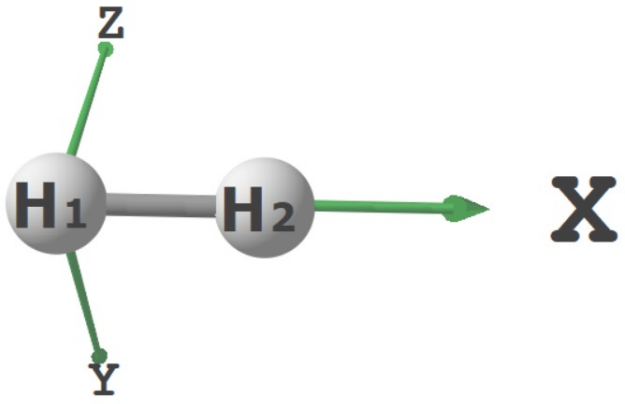
λ is the regularization parameter ensuring transferability

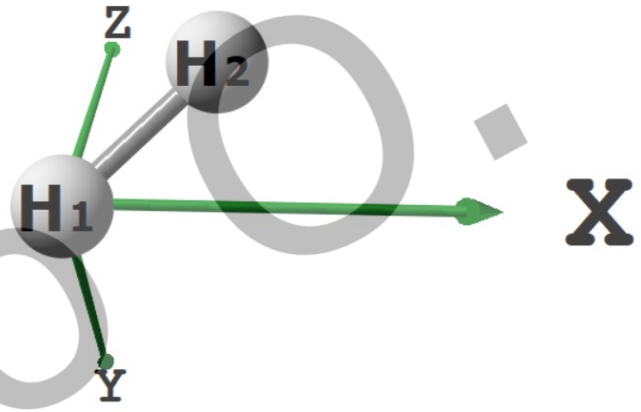
$$(\mathbf{K} + \lambda \mathbf{I})\boldsymbol{\alpha} = \mathbf{y}$$

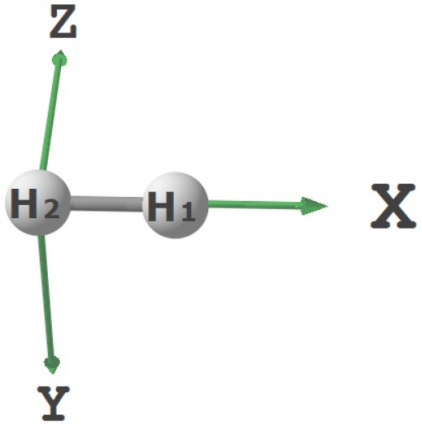
P. O. Dral, A. Owens, S. Yurchenko, W. Thiel, *J. Chem. Phys.* **2017**, *146*, 244108

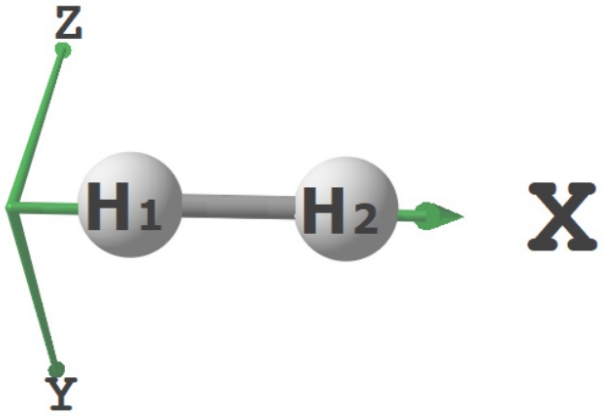
Requirements to MLPs

- rotationally and translationally invariant (or **equivariant**!)
- same-element atom index invariant





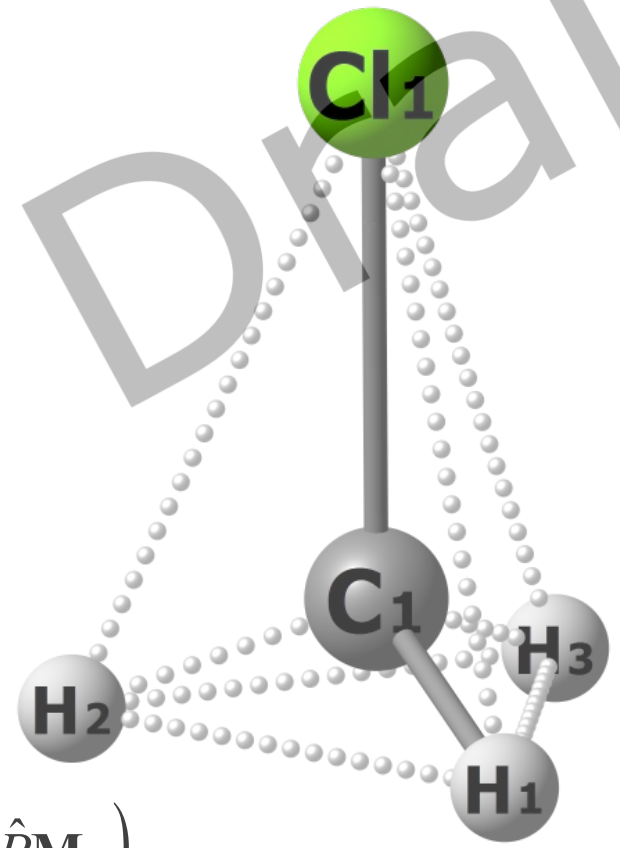




Molecular Descriptor for CH_3Cl

$$\mathbf{M} = \begin{pmatrix} \frac{r^{eq}_{C-Cl}}{r_{C-Cl}} \\ \frac{r^{eq}_{C-H1}}{r_{C-H1}} \\ \frac{r^{eq}_{Cl-H1}}{r_{Cl-H1}} \\ \frac{r^{eq}_{H1-H2}}{r_{H1-H2}} \\ \dots \end{pmatrix}$$

- Sort atoms, e.g. hydrogens can be sorted by their nuclear repulsion
- Normalized permutationally invariant kernel



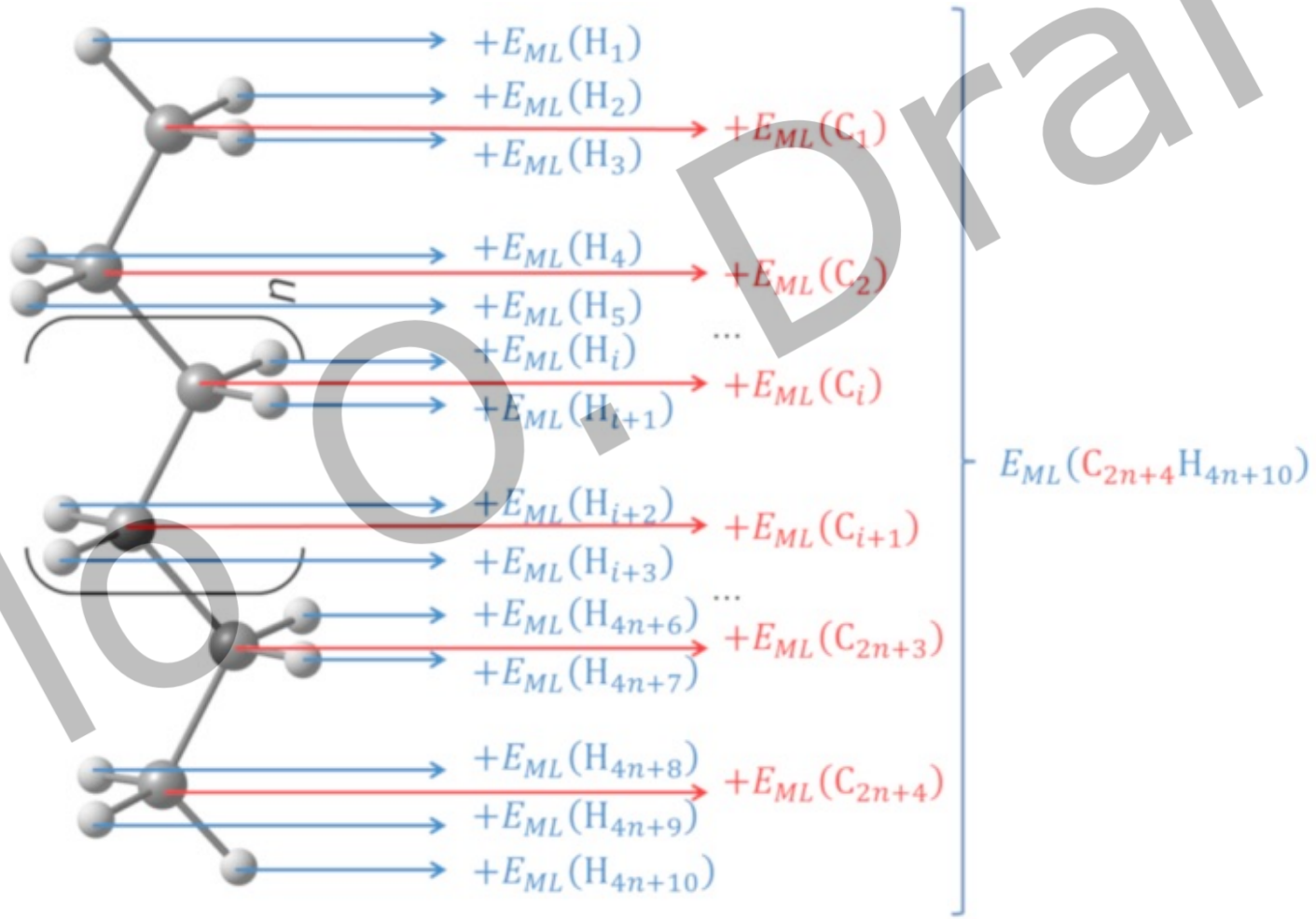
$$\overline{K}(\mathbf{M}_i, \mathbf{M}_j) = \frac{\sum_{\hat{P}}^{N_{perm}} K(\mathbf{M}_i, \hat{P}\mathbf{M}_j)}{\sqrt{\sum_{\hat{P}}^{N_{perm}} K(\mathbf{M}_i, \hat{P}\mathbf{M}_i)}\sqrt{\sum_{\hat{P}}^{N_{perm}} K(\mathbf{M}_j, \hat{P}\mathbf{M}_j)}} \quad [2]$$

[1] Dral, Owens, Yurchenko, Thiel, *J. Chem. Phys.* **2017**, *146*, 244108
[2] Bartók, Csányi, *Int. J. Quantum Chem.* **2015**, *115*, 1051

Locality in quantum chemistry

Behler–Parrinello
GAP-SOAP
ANI
PhysNet
SchNet
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

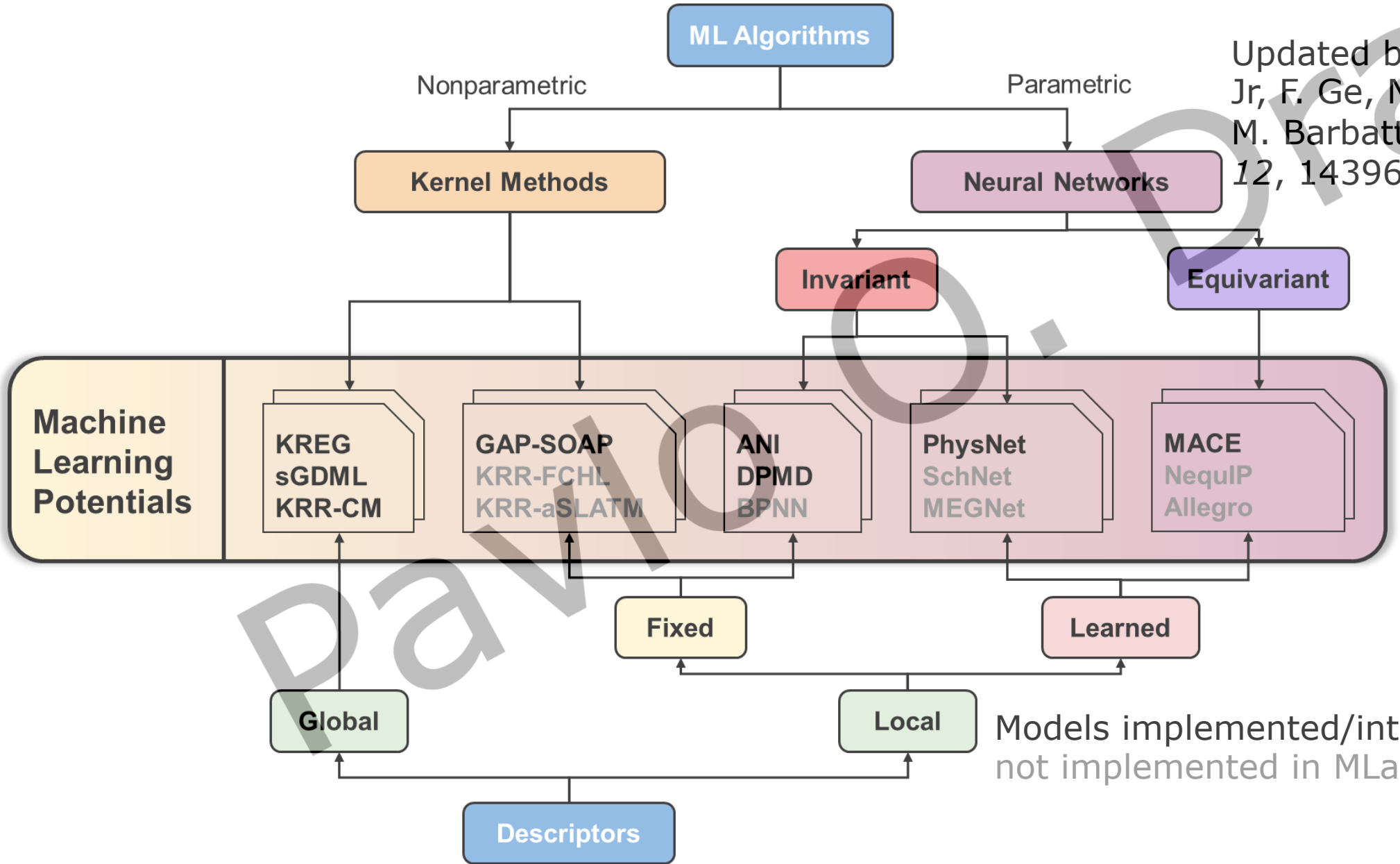
$$\mathbf{x}_i^0 = \mathbf{e}_{Z_i}$$

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


ROYAL SOCIETY OF CHEMISTRY

Certificate of appreciation
Highly cited article

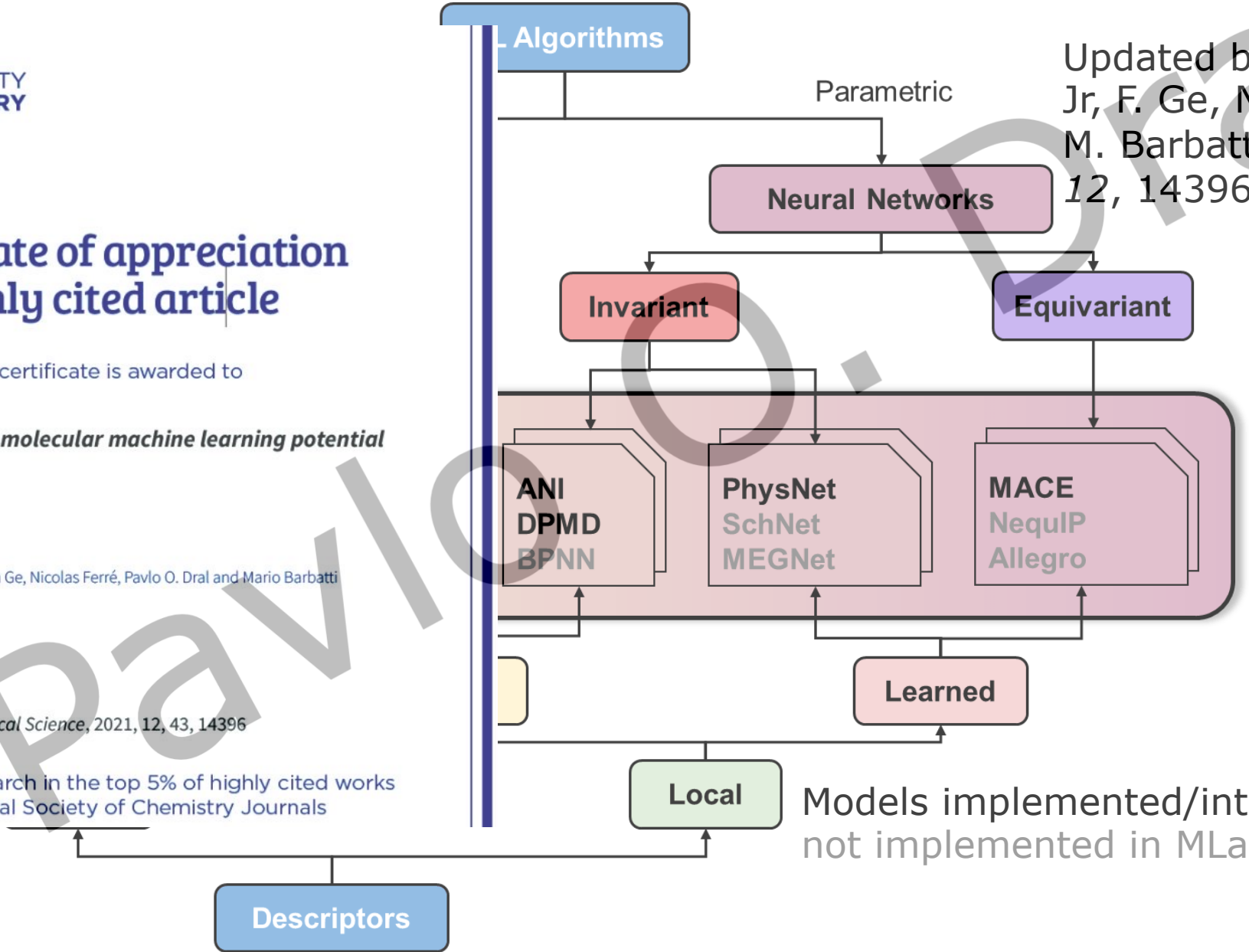
This certificate is awarded to

Choosing the right molecular machine learning potential

Max Pinheiro, Fuchun Ge, Nicolas Ferré, Pavlo O. Dral and Mario Barbatti

Chemical Science, 2021, 12, 43, 14396

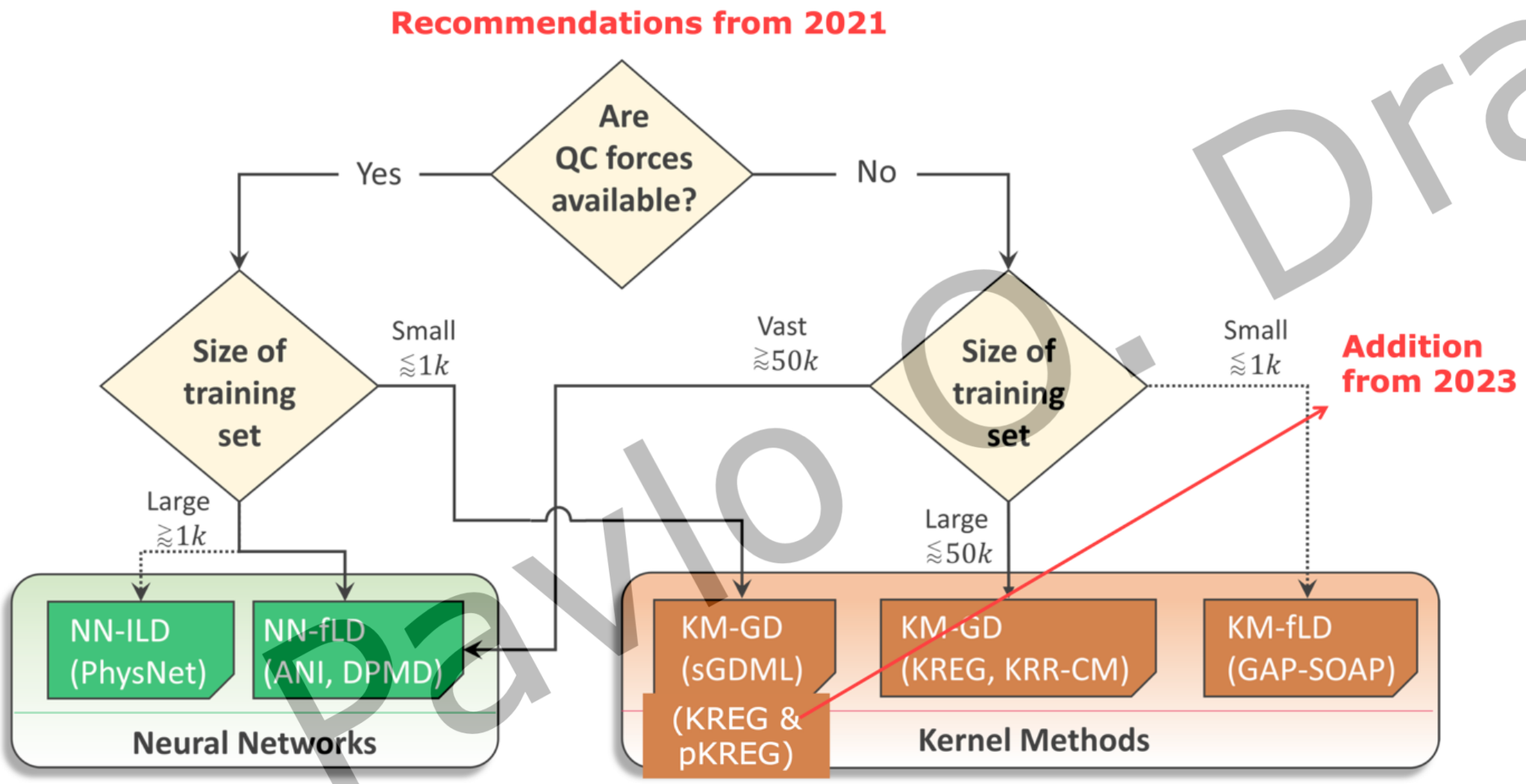
For publishing research in the top 5% of highly cited works from the Royal Society of Chemistry Journals



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



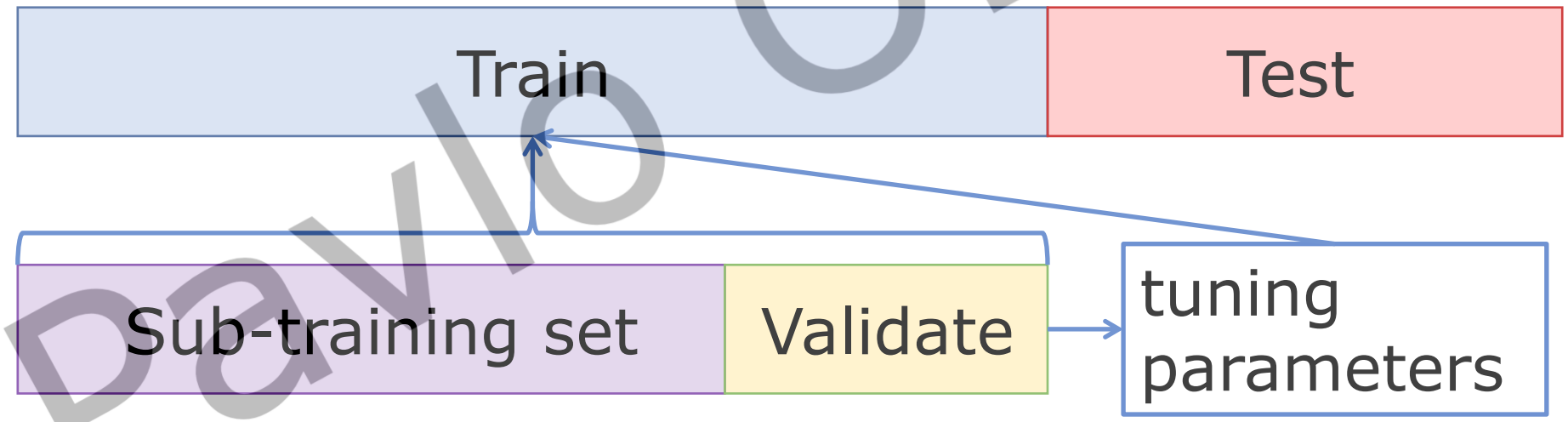
KREG & pKREG: Y.-F. Hou, F. Ge, P. O. Dral. *J. Chem. Theory Comput.* **2023**, *19*, 2369–2379

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

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**

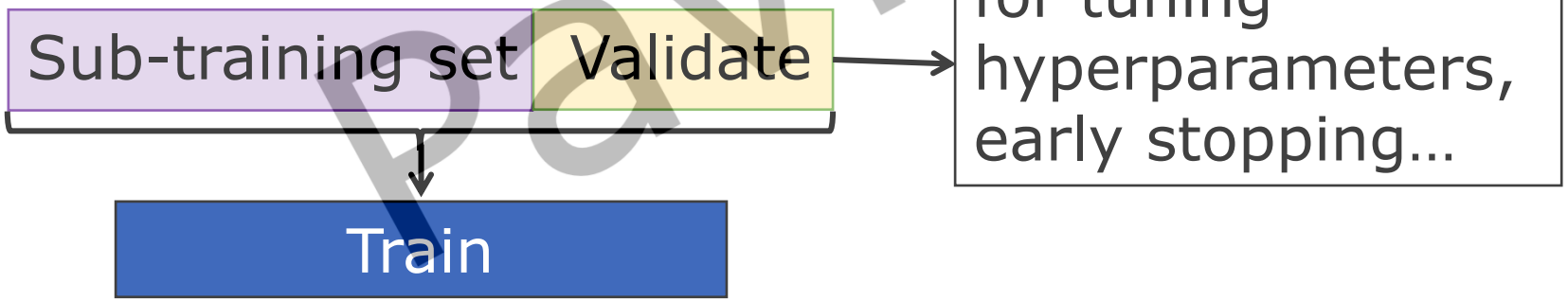
KRR-CM vs KREG

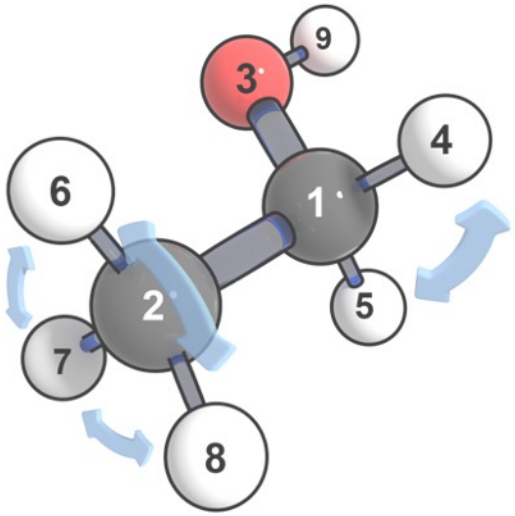
KRR-CM – kernel ridge regression with Gaussian kernel and Coulomb matrix descriptor

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

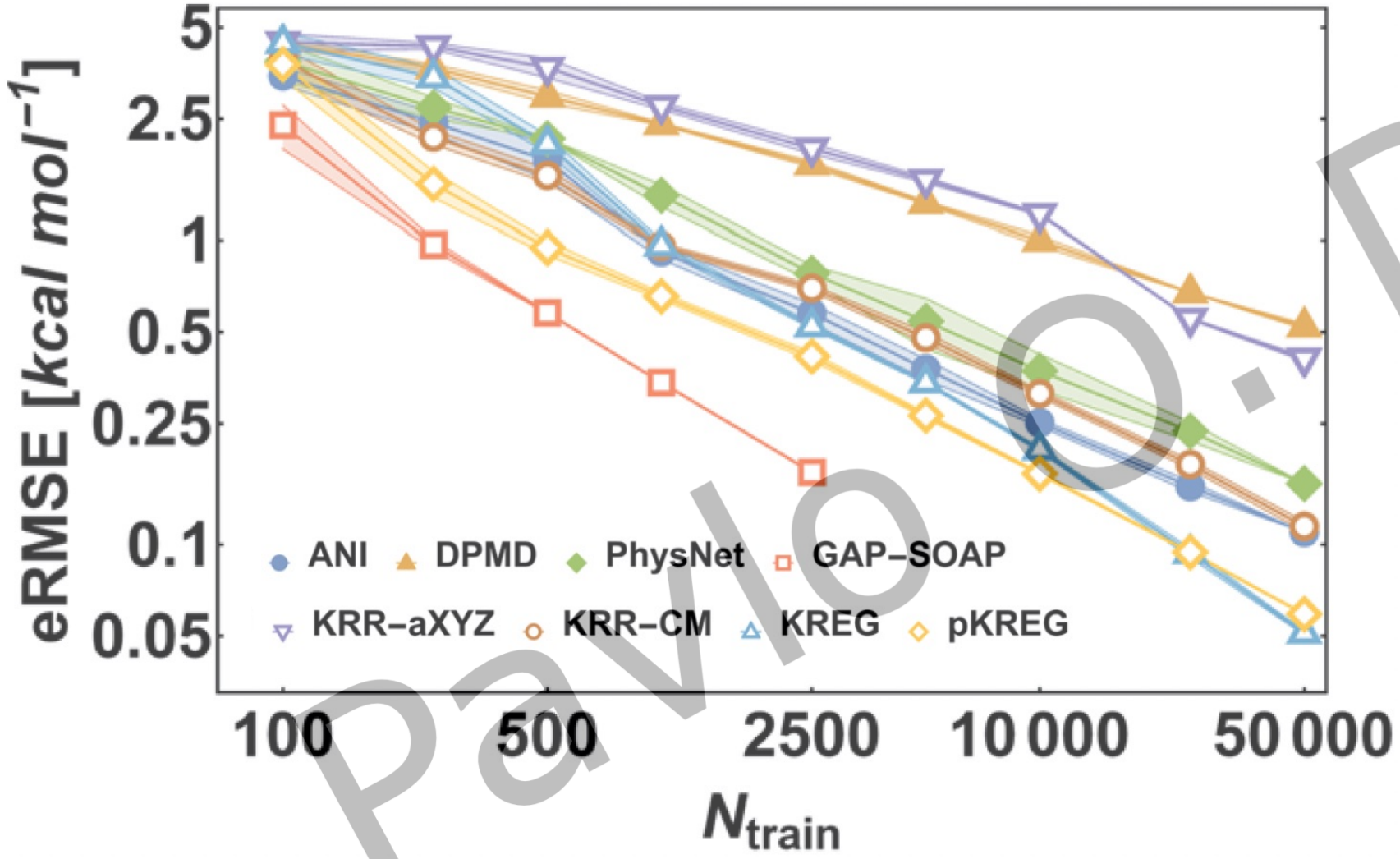
What do we mean by '100 training points?' Is the validation set included?
Let's use the term **'sub-training set'**!





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

Learning curves (energies-only)



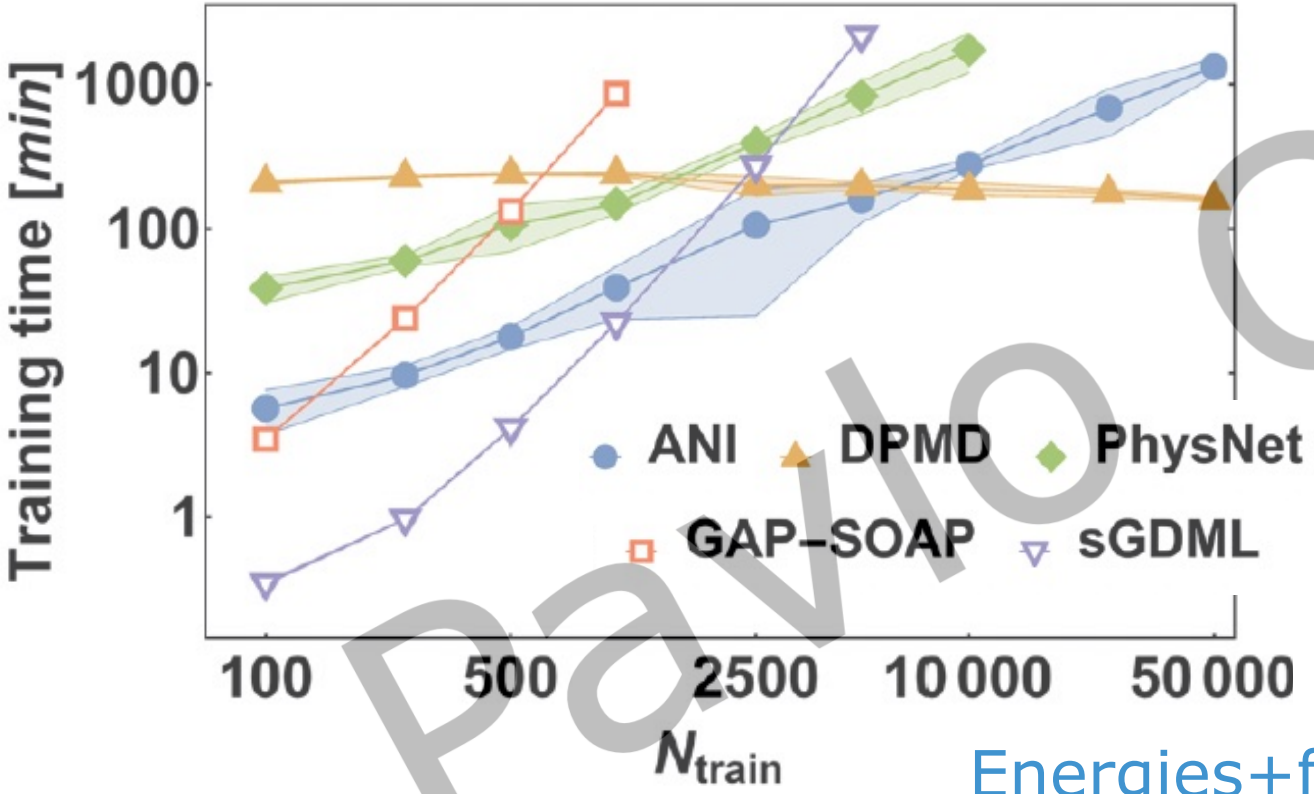
$$\varepsilon = \varepsilon_a + \frac{a}{N_{tr}^{b}}$$

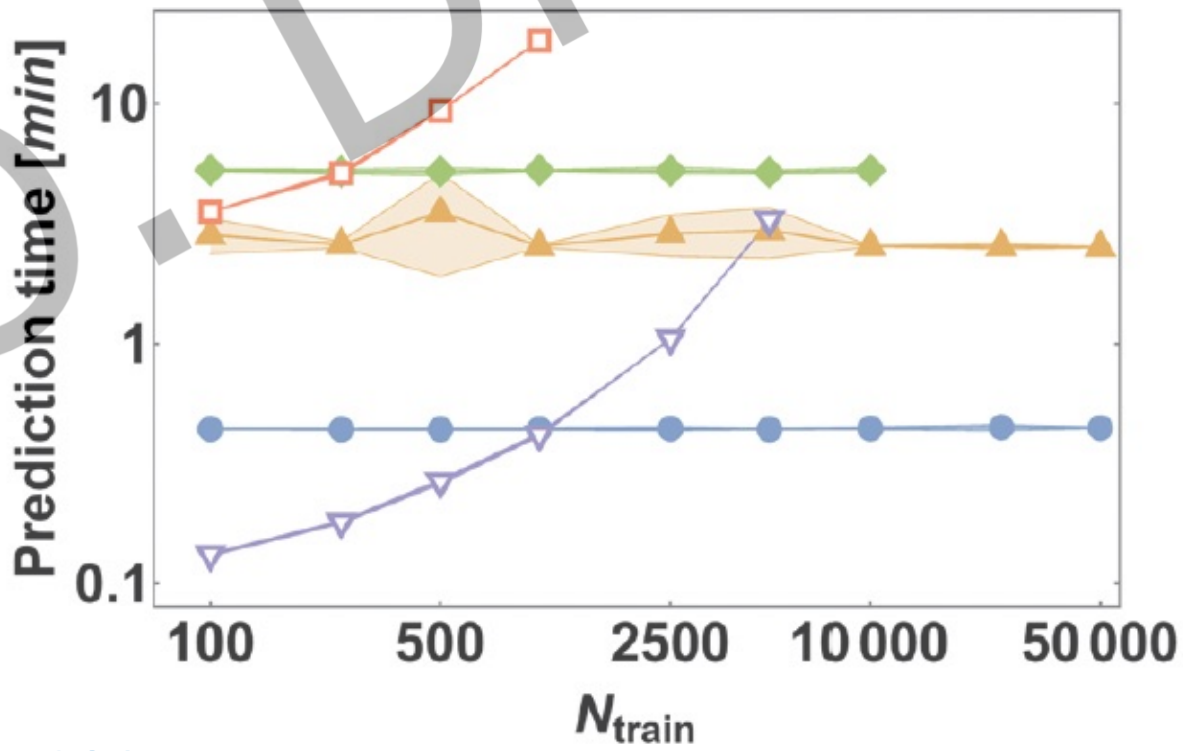
$$\log(\varepsilon) \approx \log(a) - b \log N_{tr}$$

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

Supervised Machine Learning

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

$$F_{A,d} = -\frac{\partial E}{\partial x_{A,d}}$$

Input (x) → *f*(x) → Output (y)

Given collection of known {x,y} find a function *f*(x)

training set train ML model

Use this function for making new predictions given just {x'}

- Data
- Choice of x (descriptor)
- **Choice of y (labels)**
- Fitting function (ML algorithm, ML model)
- Optimization of ML model parameters

Typically included in the NN loss

$$L = \omega_E L_E + \omega_F L_F$$

$$\mathrm{L} = \omega_E \sum_{j=1}^{N_{\mathrm{tr}}} \left(f\left(\mathbf{x}_j; \boldsymbol{\alpha}\right) - y_j\right)^2 + \omega_F \sum_{j=1}^{N_{\mathrm{tr}}} \sum_{d=1}^{p} \left(\frac{\partial f\left(\mathbf{x}_j; \boldsymbol{\alpha}\right)}{\partial x_{j,d}} - \frac{\partial y_j}{\partial x_{j,d}}\right)^2$$

Explicit inclusion in the MLP function

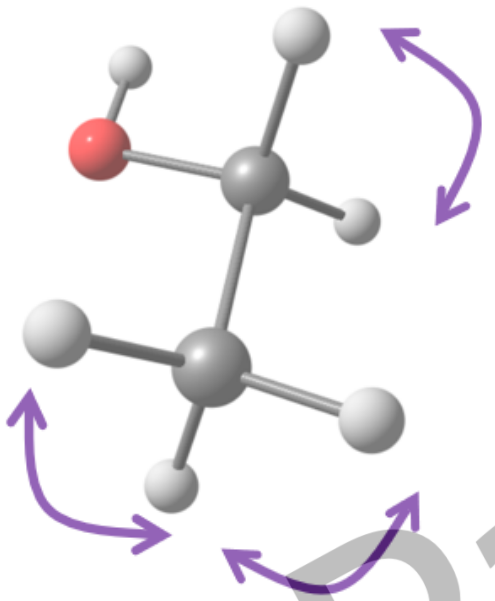
$$f(\mathbf{x}') = \sum_{i=1}^{N_{\mathrm{tr}}} \alpha_j k(\mathbf{x}_i, \mathbf{x}') + \sum_{i=1}^{N_{\mathrm{tr}}} \sum_{d=1}^{p} \alpha_{id}^{\mathrm{drv}} \frac{\partial k(\mathbf{x}_i, \mathbf{x}')}{\partial x_{i,d}}$$

$$\frac{\partial f(\mathbf{x}')}{\partial x_t'} = \sum_{i=1}^{N_{\mathrm{tr}}} \alpha_i \frac{\partial k(\mathbf{x}_i, \mathbf{x}')}{\partial x_t'} + \sum_{i=1}^{N_{\mathrm{tr}}} \sum_{d=1}^{p} \alpha_{id}^{\mathrm{drv}} \frac{\partial^2 k(\mathbf{x}_i, \mathbf{x}')}{\partial x_{i,d} \partial x_t'}$$

Explicit inclusion in the MLP function

(p)KREG

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



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

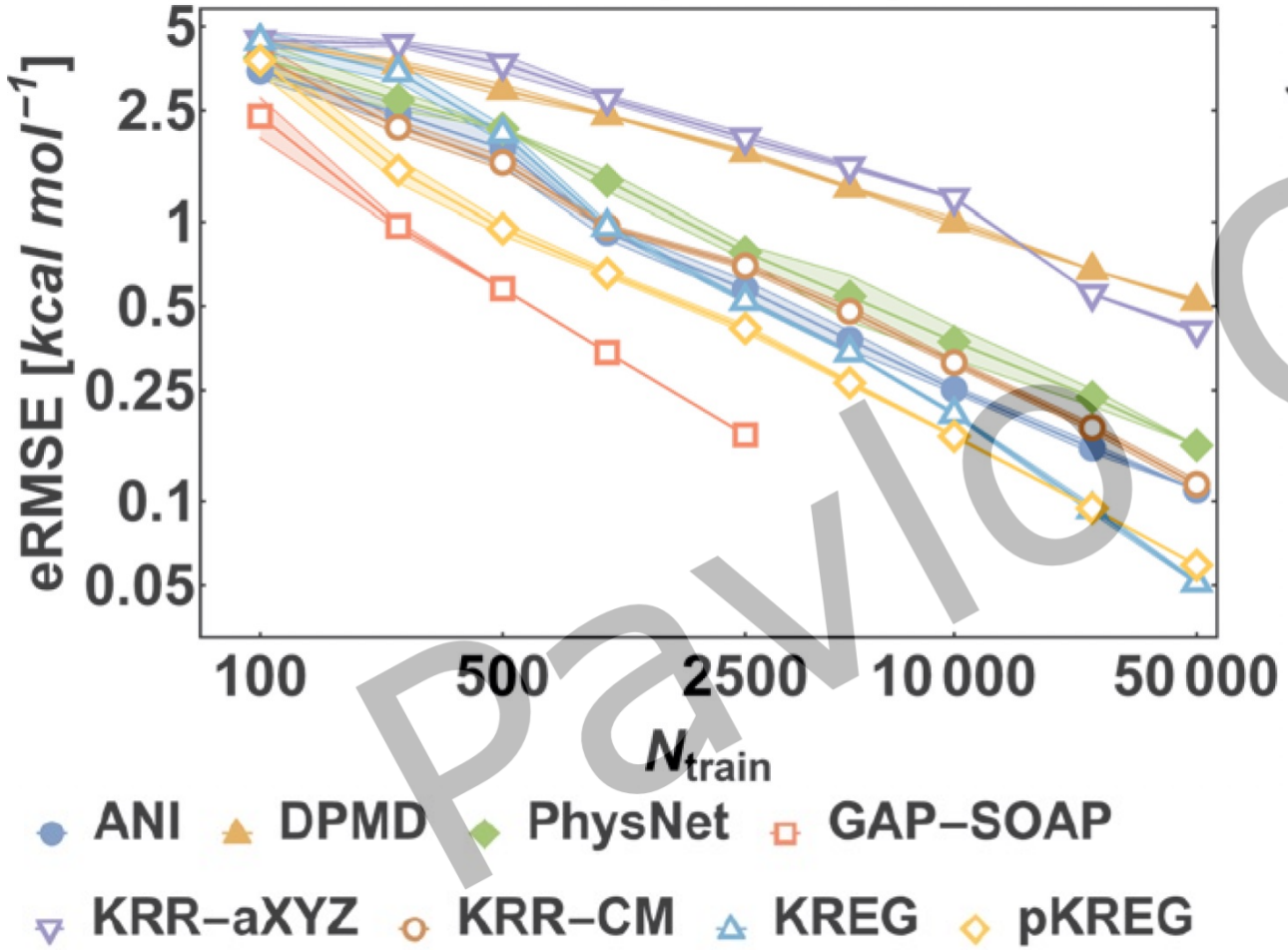
$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2 / 2\sigma^2 \right)$$

Permutations

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

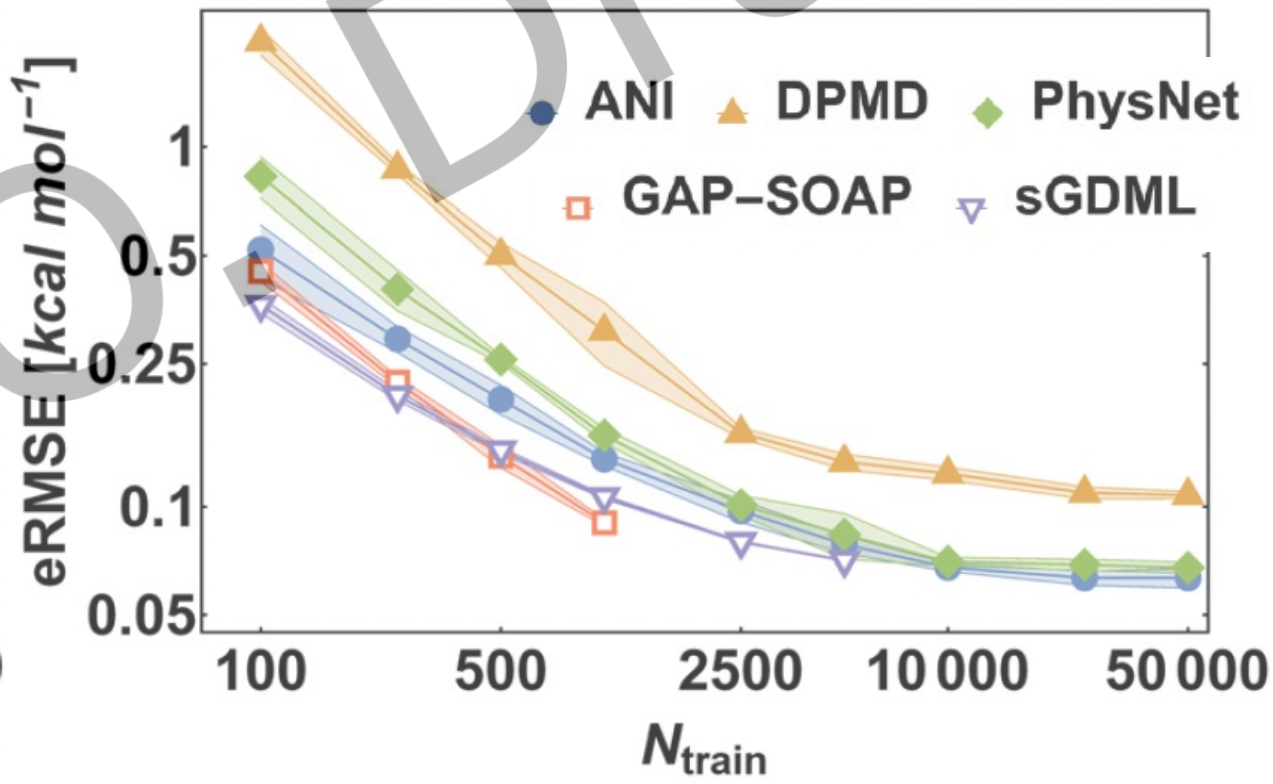
Energies-only

Ethanol (MD17)



Energies+gradients

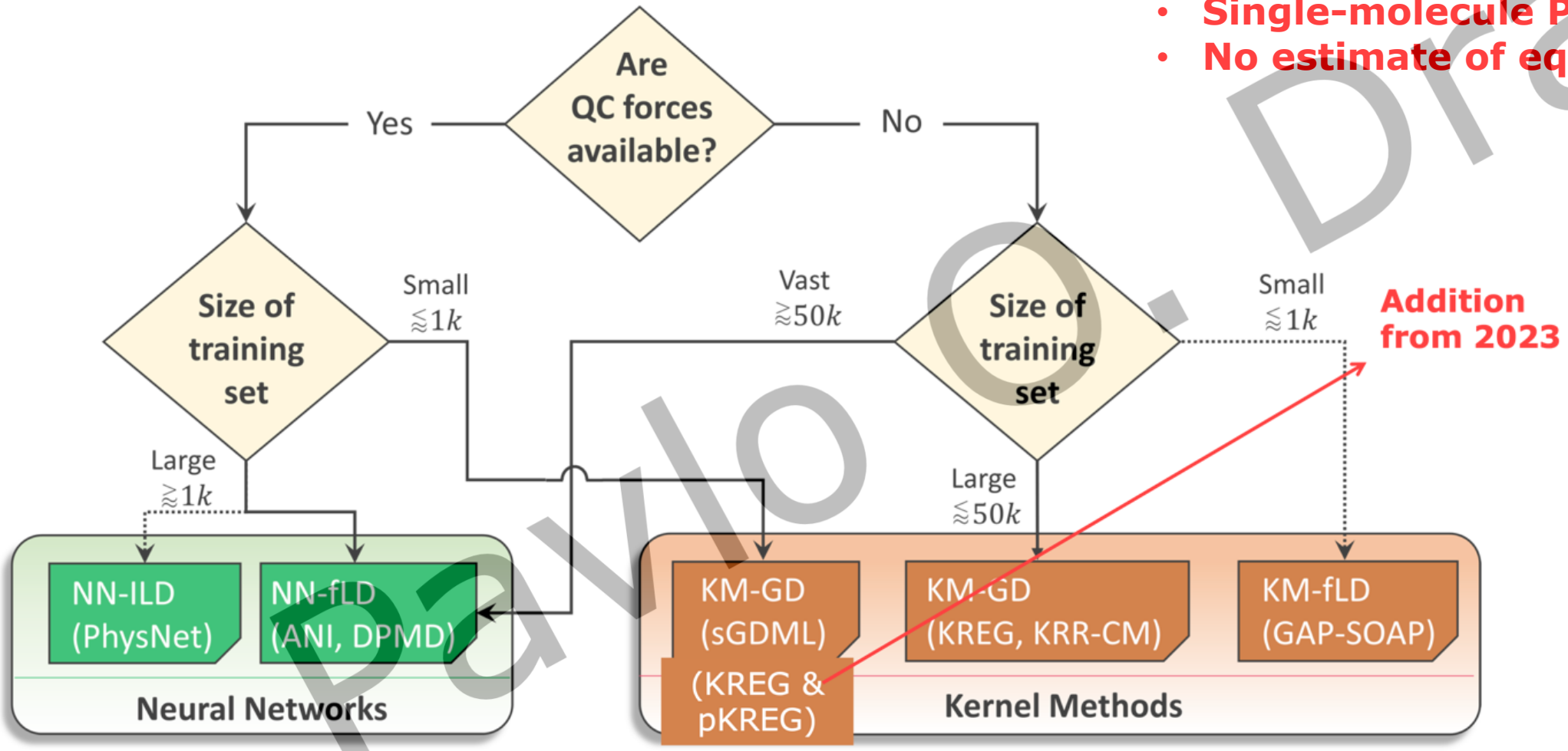
Ethanol (MD17)



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

Choosing the 'Right' ML Potential for a single molecule

Recommendations from 2021



Limitations:

- **Single-molecule PES**
- **No estimate of equivariant MLPs**

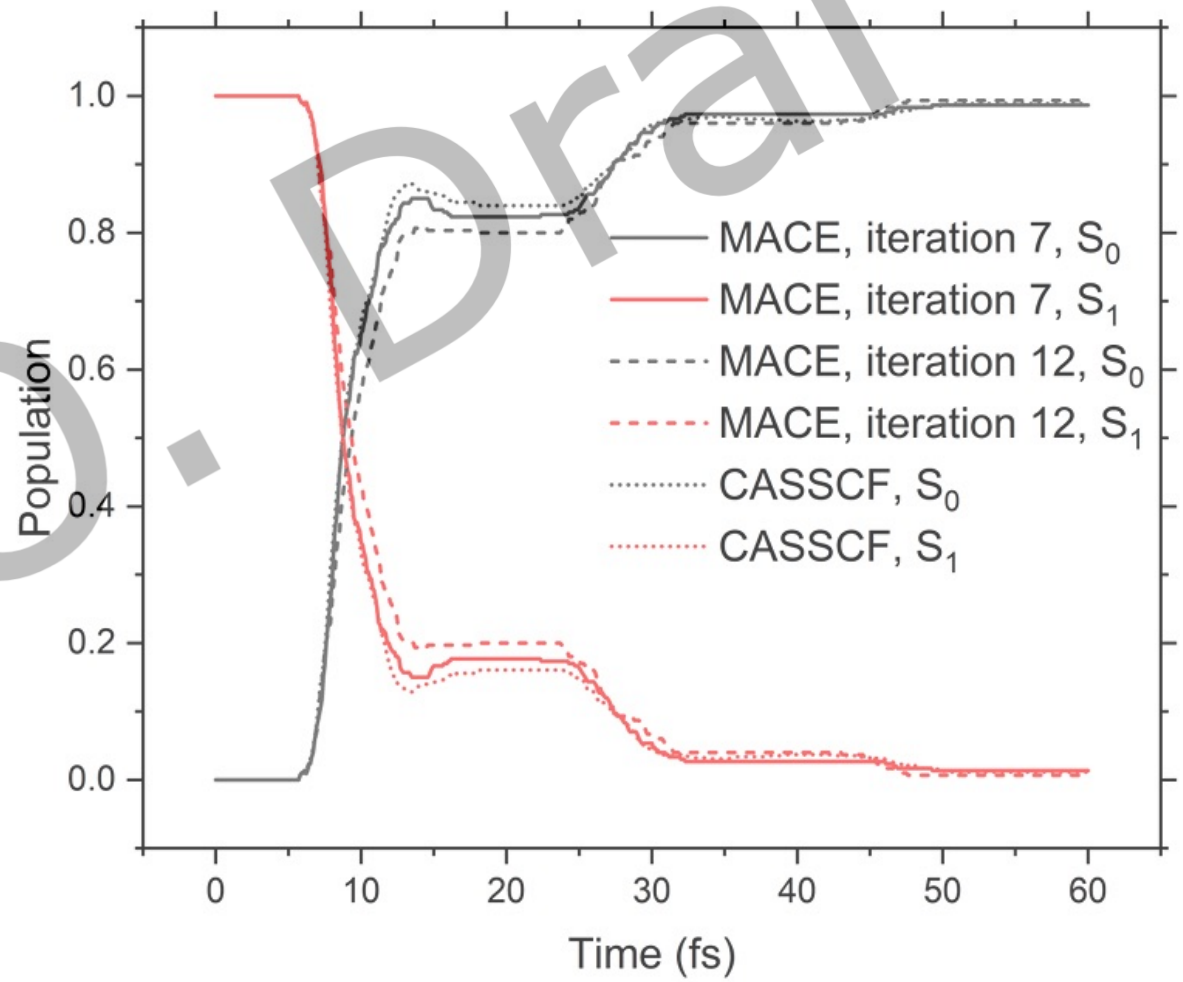
KREG & pKREG: Y.-F. Hou, F. Ge, P. O. Dral. *J. Chem. Theory Comput.* **2023**, *19*, 2369–2379

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

Can equivariant NNs help?

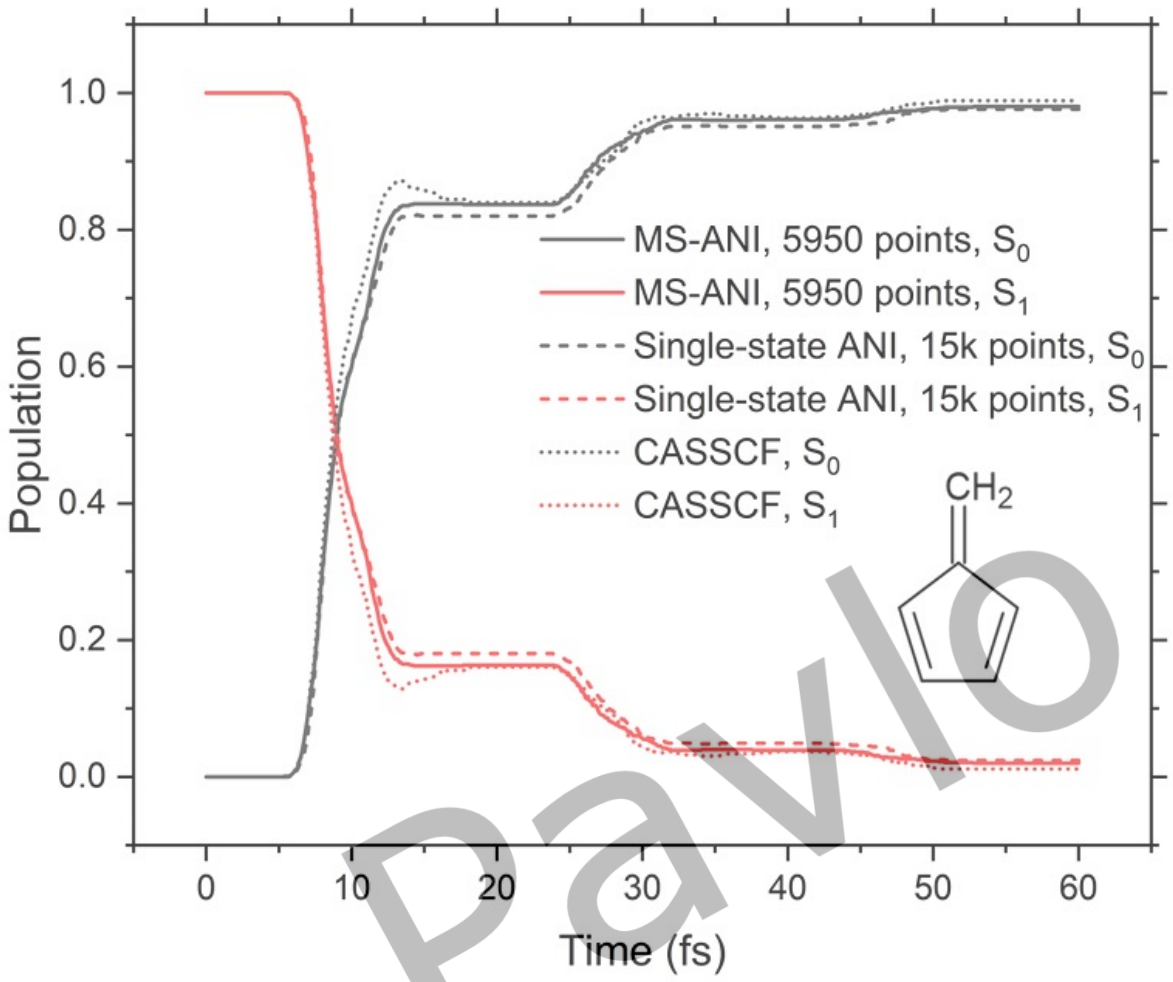
Crashed after 28 days of active learning after 13 iterations and producing 3850 training points



Use equivariant MLPs if you can like use CCSD(T) if you can (mostly we use ANI-type like mostly people use DFT)

M. Martyka, L. Zhang, F. Ge, Y.-F. Hou, J. Jankowska, M. Barbatti, P. O. Dral. *Charting electronic-state manifolds across molecules with multi-state learning and gap-driven dynamics via efficient and robust active learning*. https://doi.org/10.26434/chemrxiv-2024-dtc1w

Active learning for surface hopping



19 iterations of active learning

Three days *on*
RTX 4090 GPU and 16 Intel Xeon Gold 6226R CPUs
for the entire procedure with trainings and labeling

M. Martyka, L. Zhang, F. Ge, Y.-F. Hou, J. Jankowska, M. Barbatti, P. O. Dral. *Charting electronic-state manifolds across molecules with multi-state learning and gap-driven dynamics via efficient and robust active learning*. https://doi.org/10.26434/chemrxiv-2024-dtc1w

XACS
Xiamen Atomistic Computing Suite
XACScloud.com

THE JOURNAL OF PHYSICAL CHEMISTRY A
A JOURNAL OF THE AMERICAN CHEMICAL SOCIETY

MLatom.com

pubs.acs.org/JPCA

Perspective

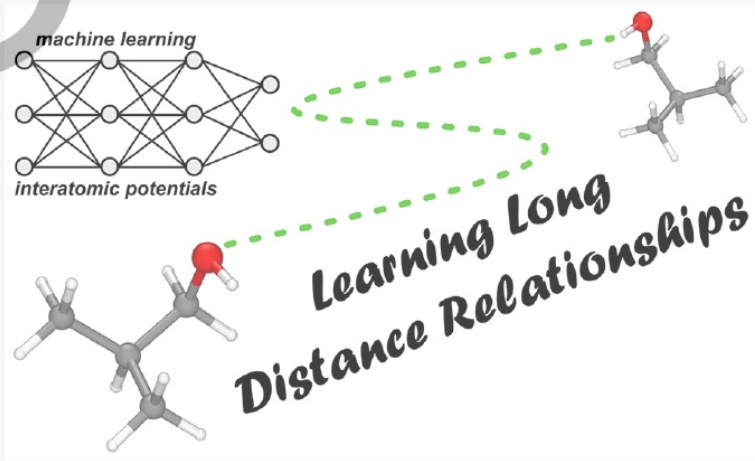
Machine Learning Interatomic Potentials and Long-Range Physics

Dylan M. Anstine and Olexandr Isayev*

Cite This: *J. Phys. Chem. A* 2023, 127, 2417–2431

Read Online

ABSTRACT: Advances in machine learned interatomic potentials (MLIPs), such as those using neural networks, have resulted in short-range models that can infer interaction energies with near ab initio accuracy and orders of magnitude reduced computational cost. For many atom systems, including macromolecules, biomolecules, and condensed matter, model accuracy can become reliant on the description of short- and long-range physical interactions. The latter terms can be difficult to incorporate into an MLIP framework. Recent research has produced numerous models with considerations for nonlocal electrostatic and dispersion interactions, leading to a large range of applications that can be addressed using MLIPs. In light of this, we present a Perspective focused on key methodologies and models being used where the presence of nonlocal physics and chemistry are crucial for describing system properties. The strategies covered include MLIPs augmented with dispersion corrections, electrostatics calculated with charges predicted from atomic environment descriptors, the use of self-consistency and message passing iterations to propagated nonlocal system information, and charges obtained via equilibration schemes. We aim to provide a pointed discussion to support the development of machine learning-based interatomic potentials for systems where contributions from only nearsighted terms are deficient.



CHEMICAL REVIEWS

pubs.acs.org/CR

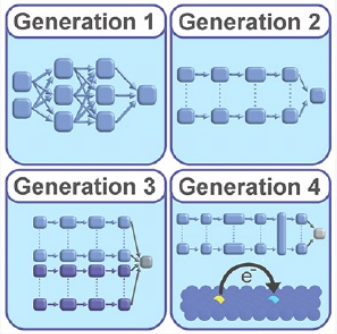
Review

Four Generations of High-Dimensional Neural Network Potentials

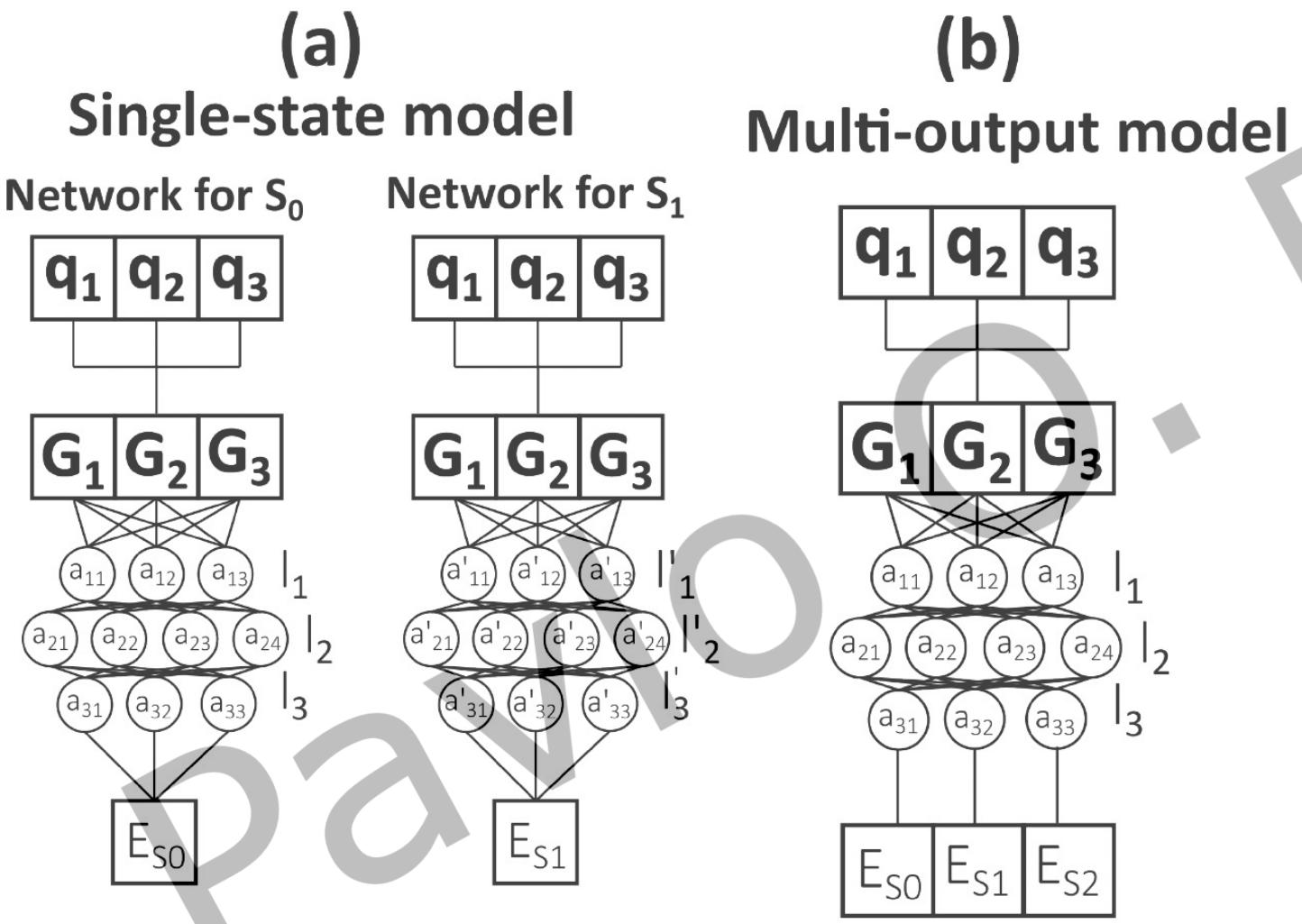
Jörg Behler*

Cite This: *Chem. Rev.* 2021, 121, 10037−10072

ABSTRACT: Since their introduction about 25 years ago, machine learning (ML) potentials have become an important tool in the field of atomistic simulations. After the initial decade, in which neural networks were successfully used to construct potentials for rather small molecular systems, the development of high-dimensional neural network potentials (HDNNPs) in 2007 opened the way for the application of ML potentials in simulations of large systems containing thousands of atoms. To date, many other types of ML potentials have been proposed continuously increasing the range of problems that can be studied. In this review, the methodology of the family of HDNNPs including new recent developments will be discussed using a classification scheme into four generations of potentials, which is also applicable to many other types of ML potentials. The first generation is formed by early neural network potentials designed for low-dimensional systems. High-dimensional neural network potentials established the second generation and are based on three key steps: first, the expression of the total energy as a sum of environment-dependent atomic energy contributions; second, the description of the atomic environments by atom-centered symmetry functions as descriptors fulfilling the requirements of rotational, translational, and permutation invariance; and third, the iterative construction of the reference electronic structure data sets by active learning. In third-generation HDNNPs, in addition, long-range interactions are included employing environment-dependent partial charges expressed by atomic neural networks. In fourth-generation HDNNPs, which are just emerging, in addition, nonlocal phenomena such as long-range charge transfer can be included. The applicability and remaining limitations of HDNNPs are discussed along with an outlook at possible future developments.

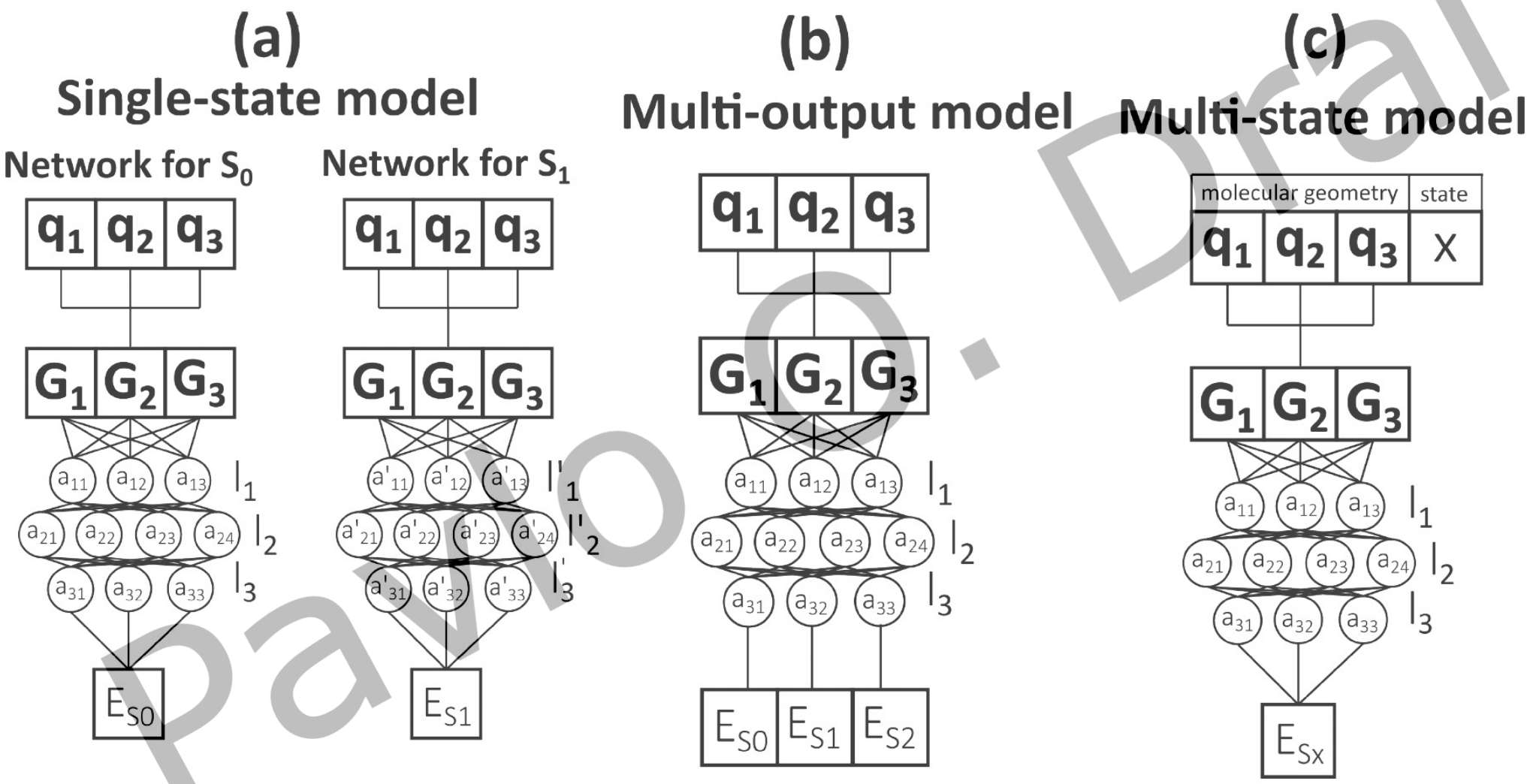


Possible MLPs for excited states



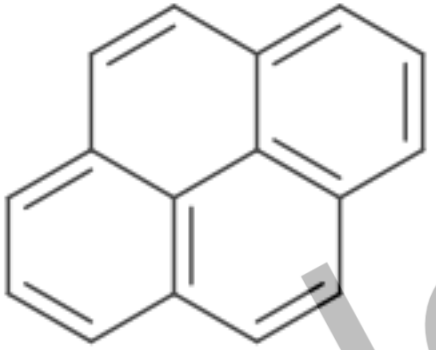
M. Martyka, L. Zhang, F. Ge, Y.-F. Hou, J. Jankowska, M. Barbatti, P. O. Dral. *Charting electronic-state manifolds across molecules with multi-state learning and gap-driven dynamics via efficient and robust active learning*. https://doi.org/10.26434/chemrxiv-2024-dtc1w

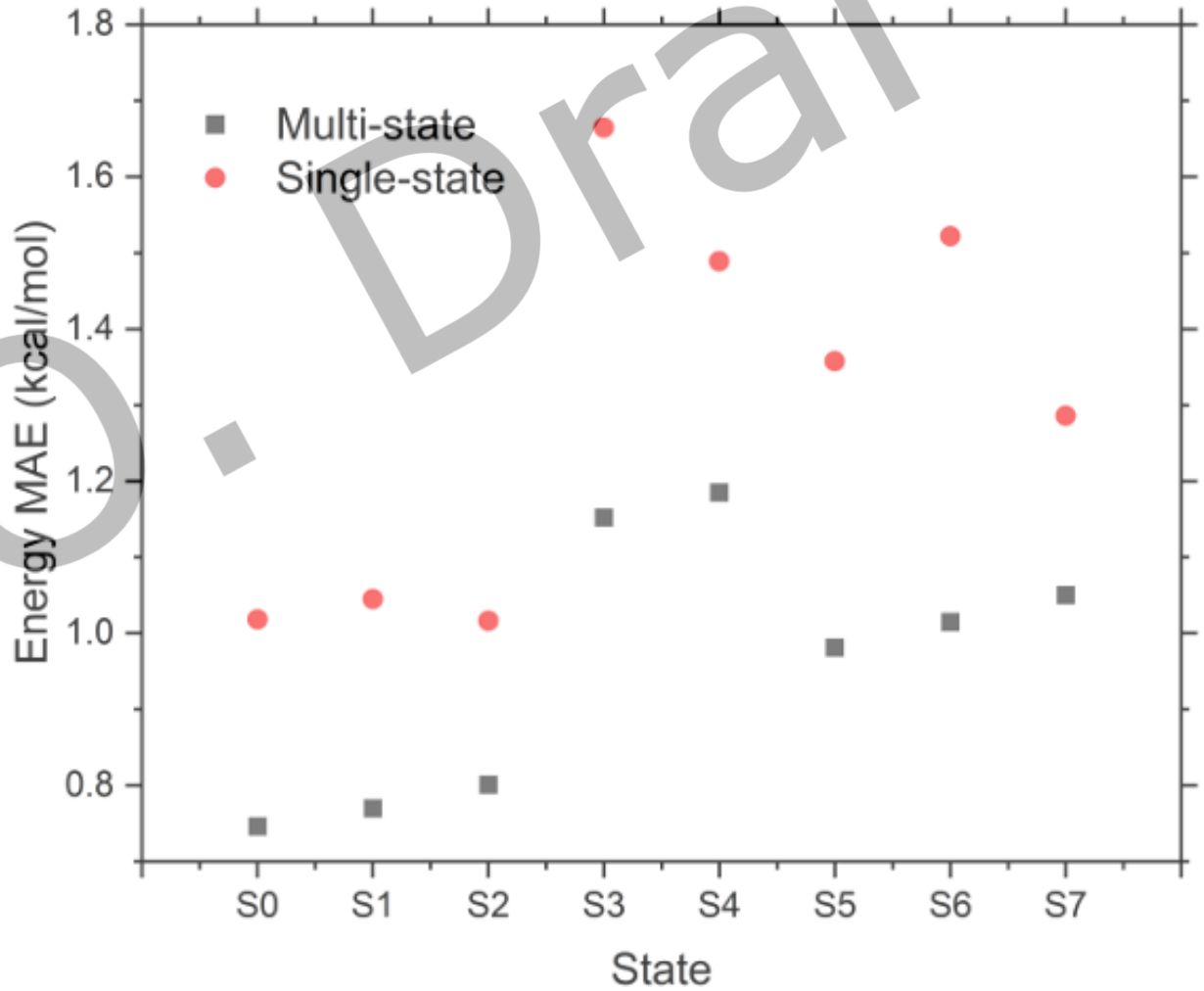
Possible MLPs for excited states



M. Martyka, L. Zhang, F. Ge, Y.-F. Hou, J. Jankowska, M. Barbatti, P. O. Dral. *Charting electronic-state manifolds across molecules with multi-state learning and gap-driven dynamics via efficient and robust active learning*. https://doi.org/10.26434/chemrxiv-2024-dtc1w

Multi-state ANI (MS-ANI)

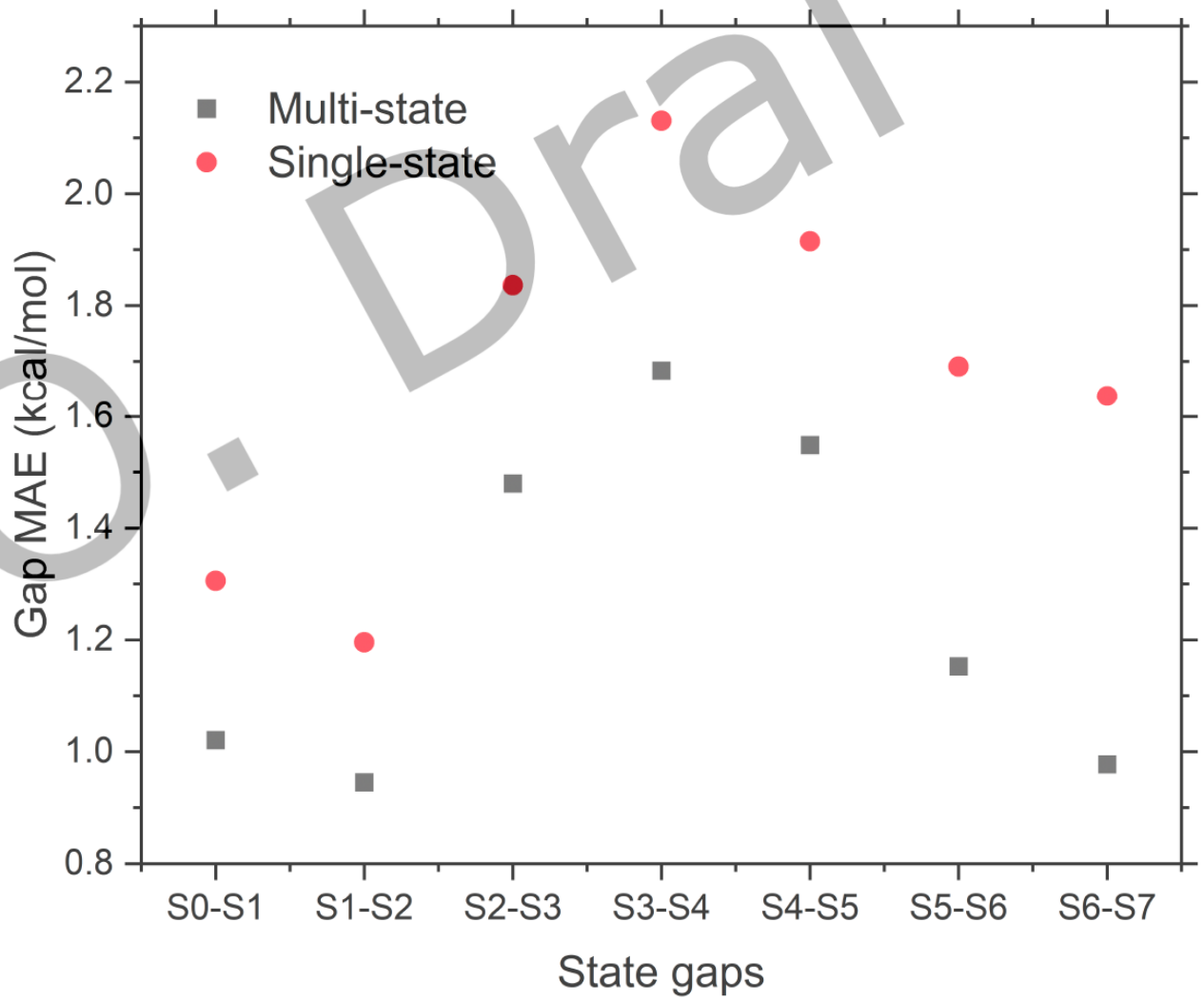




M. Martyka, L. Zhang, F. Ge, Y.-F. Hou, J. Jankowska, M. Barbatti, P. O. Dral. *Charting electronic-state manifolds across molecules with multi-state learning and gap-driven dynamics via efficient and robust active learning*. https://doi.org/10.26434/chemrxiv-2024-dtc1w

Multi-state ANI (MS-ANI)

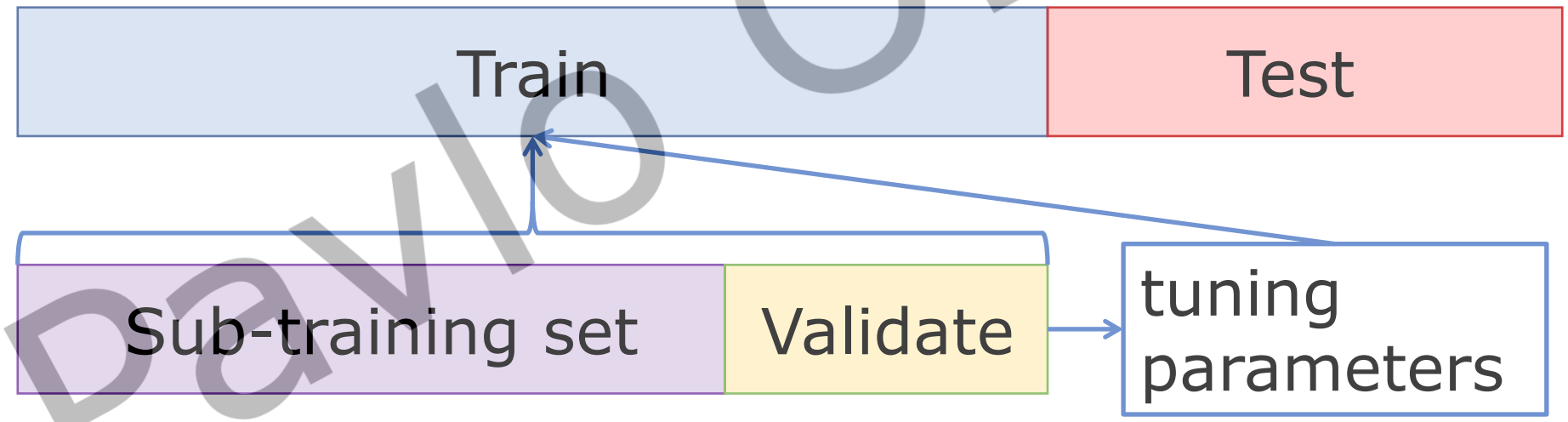
$$L = \omega_E L_E + \omega_F L_F + \omega_{\text{gap}} L_{\text{gap}}.$$



M. Martyka, L. Zhang, F. Ge, Y.-F. Hou, J. Jankowska, M. Barbatti, P. O. Dral. *Charting electronic-state manifolds across molecules with multi-state learning and gap-driven dynamics via efficient and robust active learning*. https://doi.org/10.26434/chemrxiv-2024-dtc1w

ML: Error Estimation

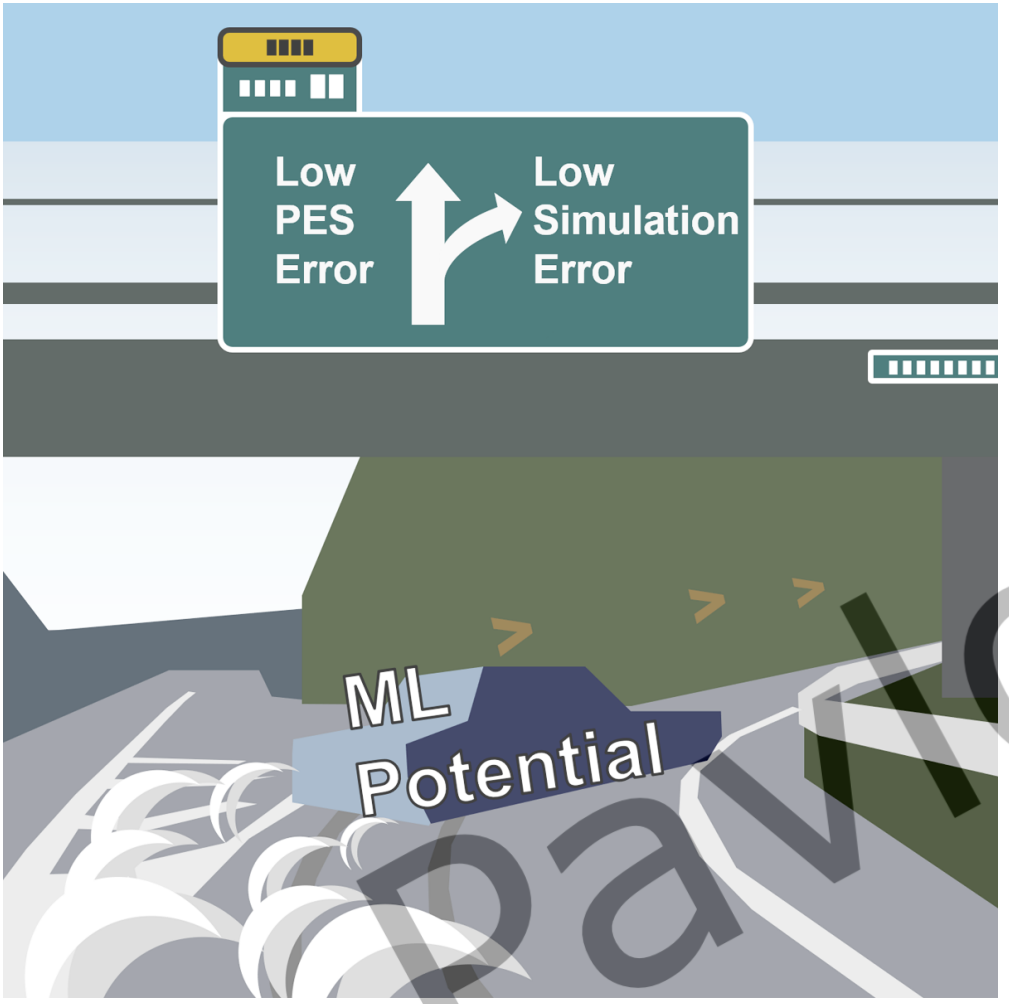
- Often ML error for its own training set is close to zero
- Using errors in the validation set would be also incorrect, because their minimization is a part of the training process
- We should estimate errors on a **completely independent test set**

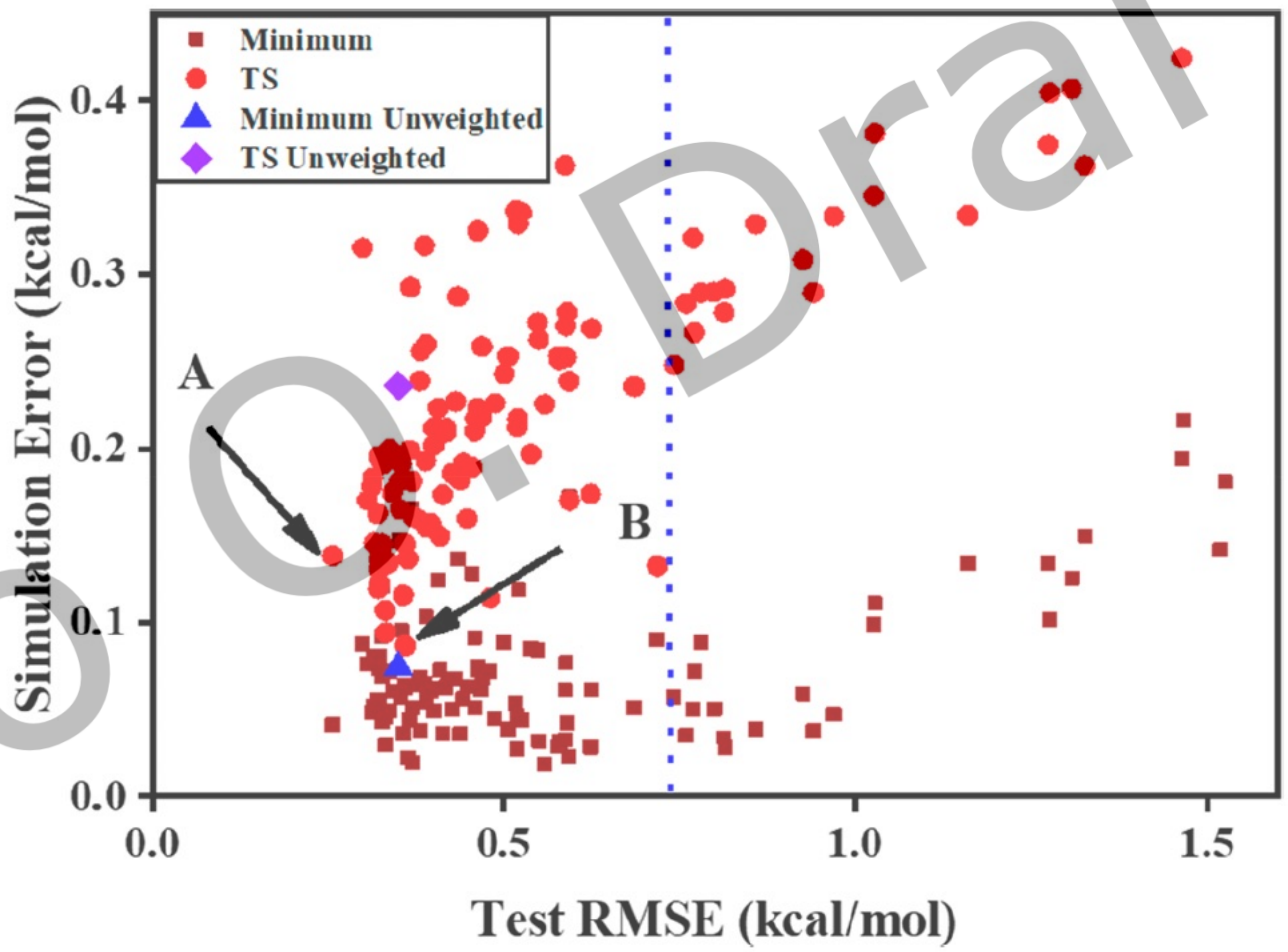


Hastie, Tibshirani, Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed.; Springer-Verlag, **2009**

The ultimate test is the performance in the required application!

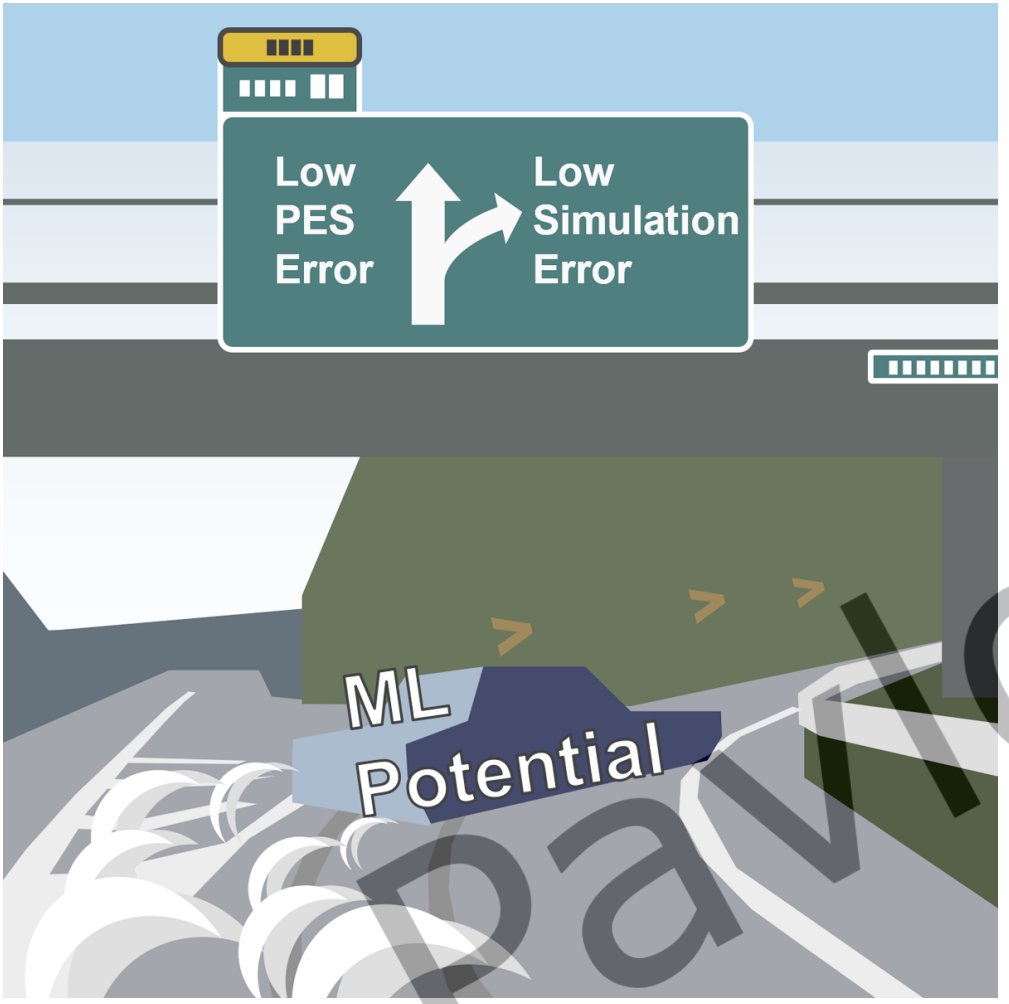
Quality of simulation ≠ quality of fit





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.

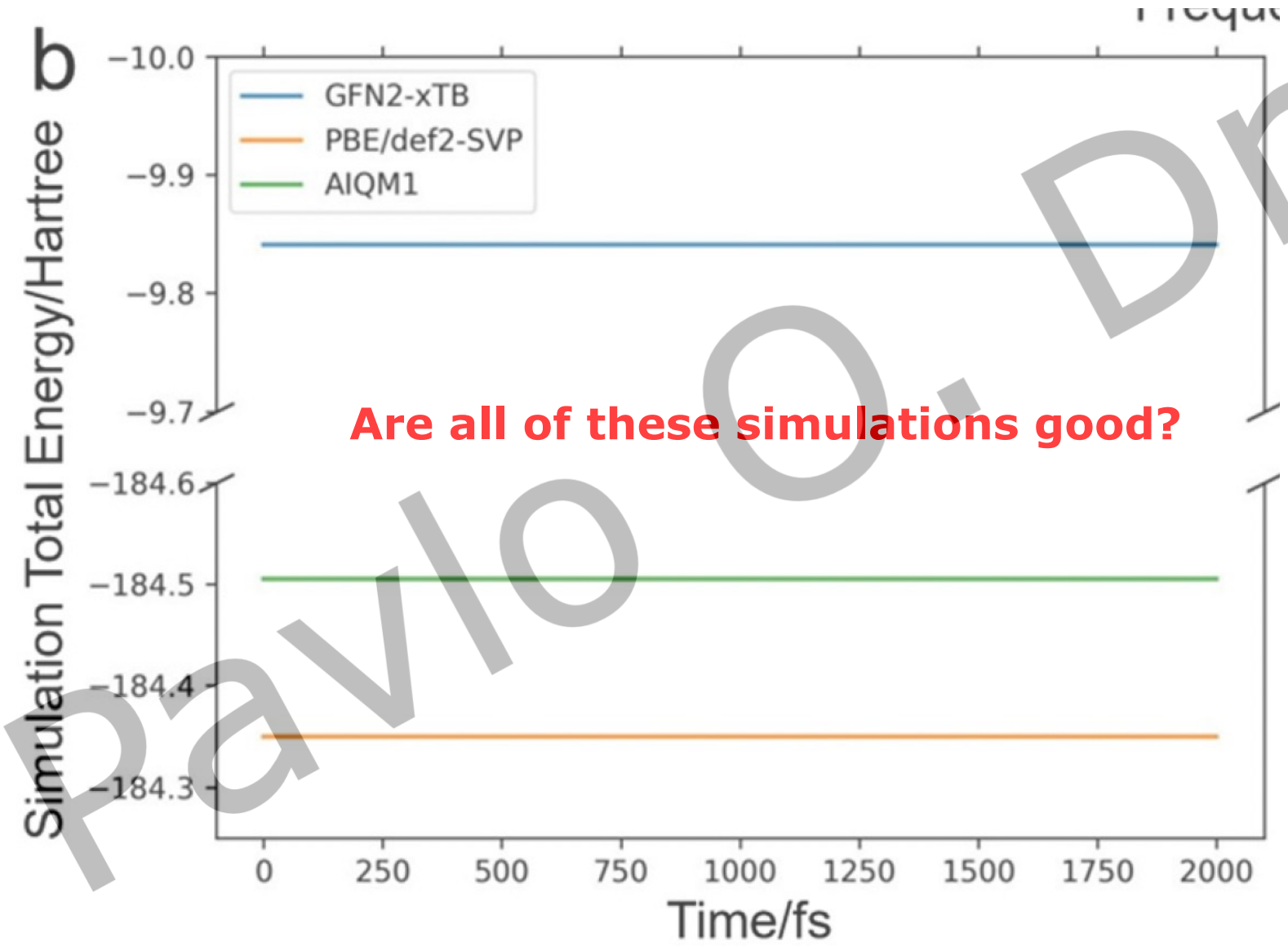
"DMC-certified"



"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."

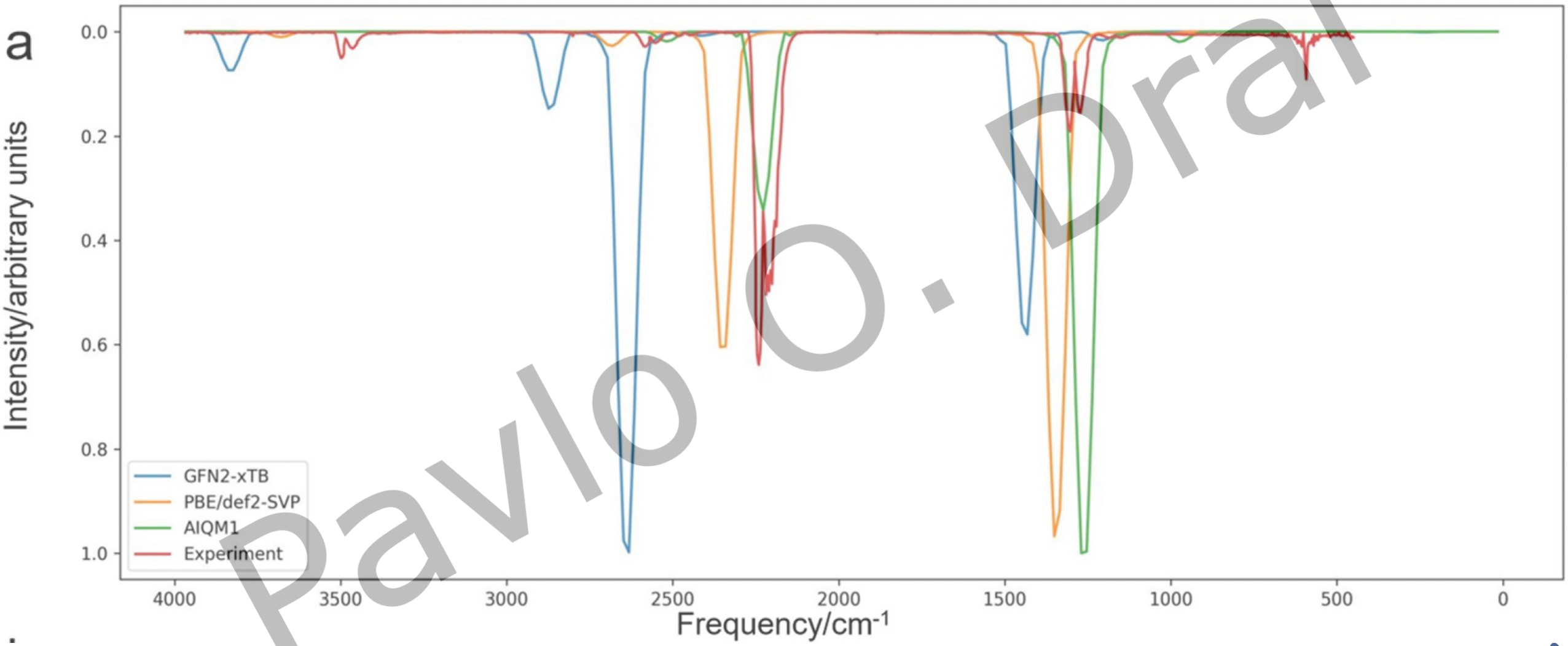
→ 60 hours on a single GPU with ANI…

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.



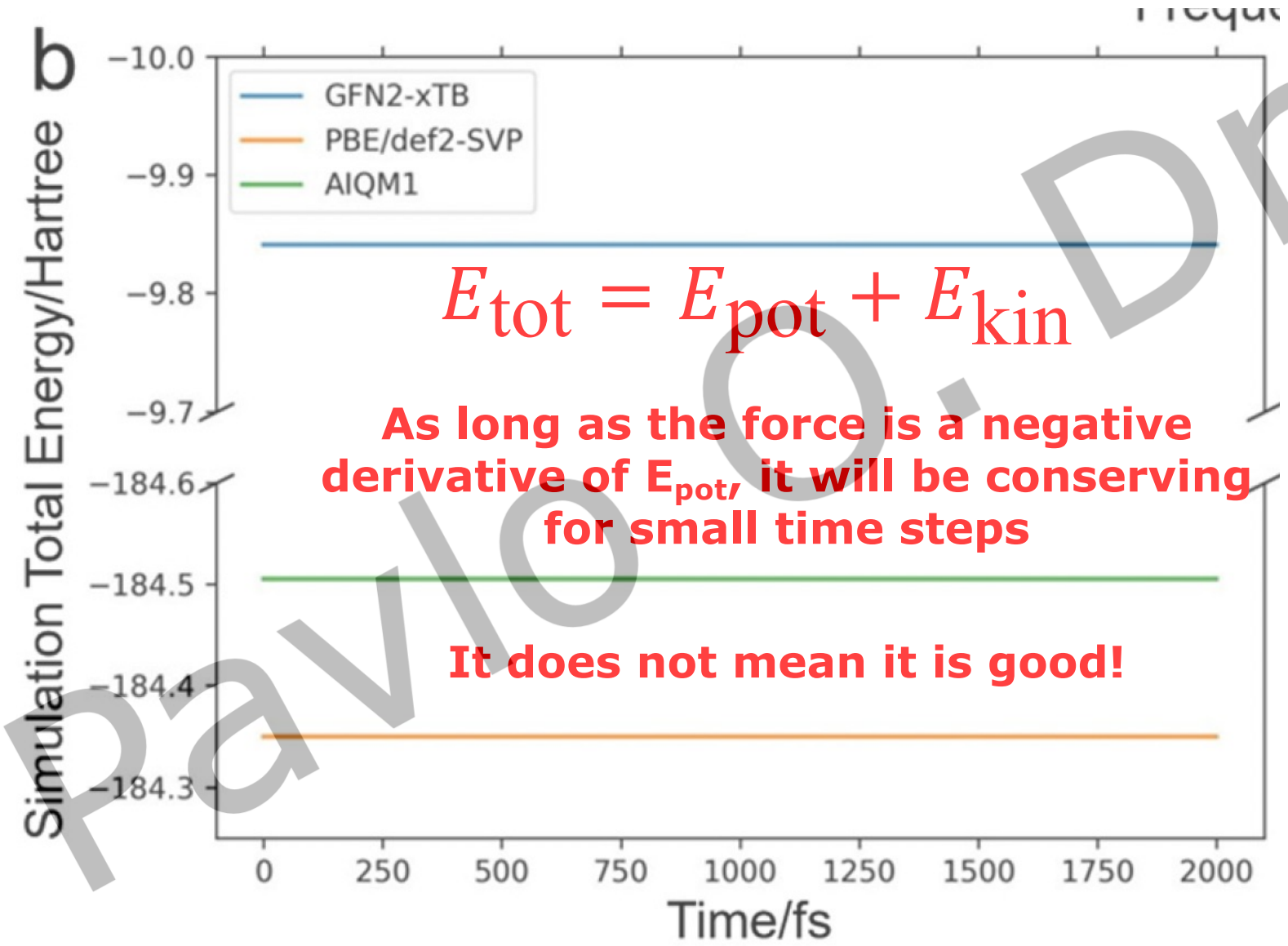
Are all of these simulations good?

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

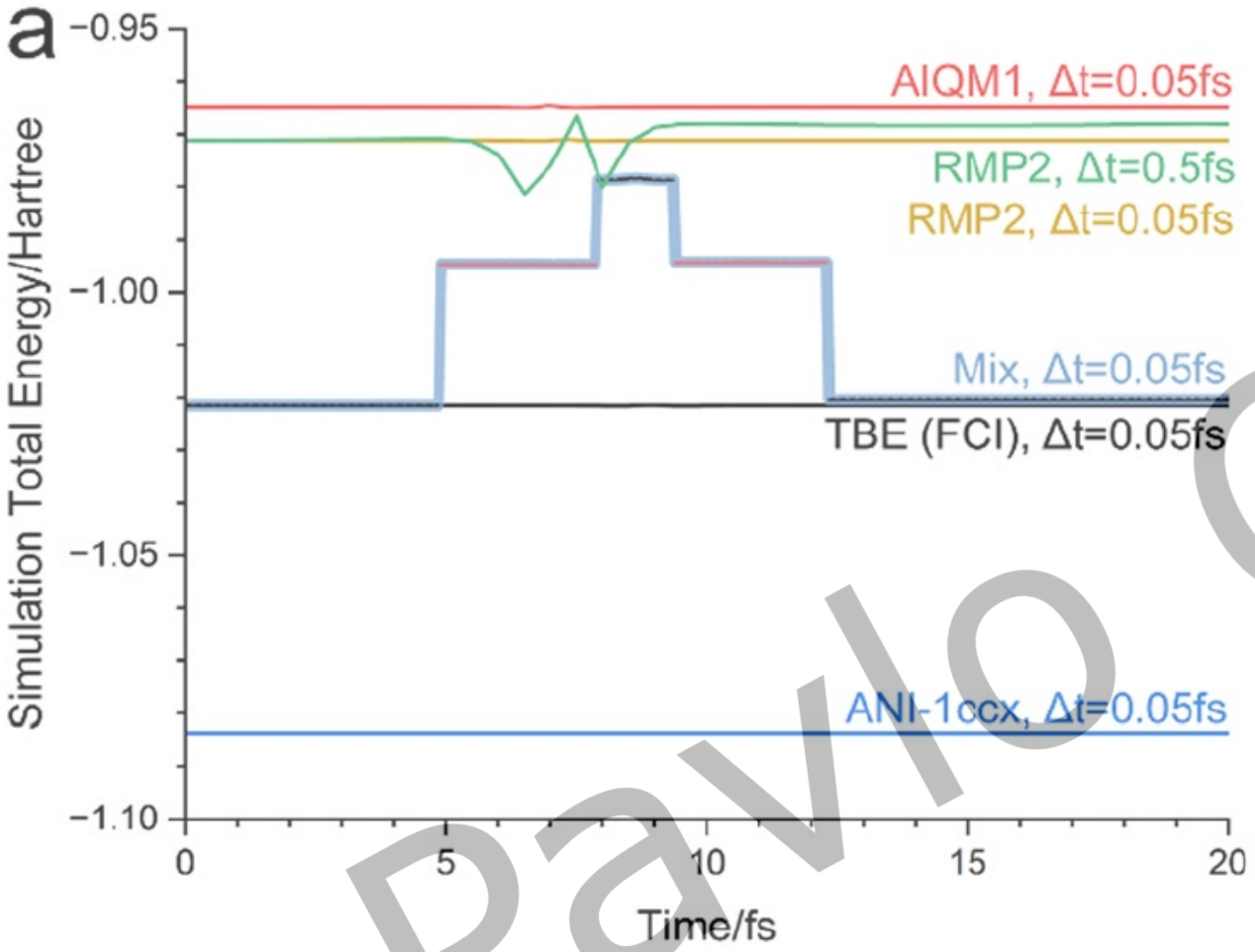


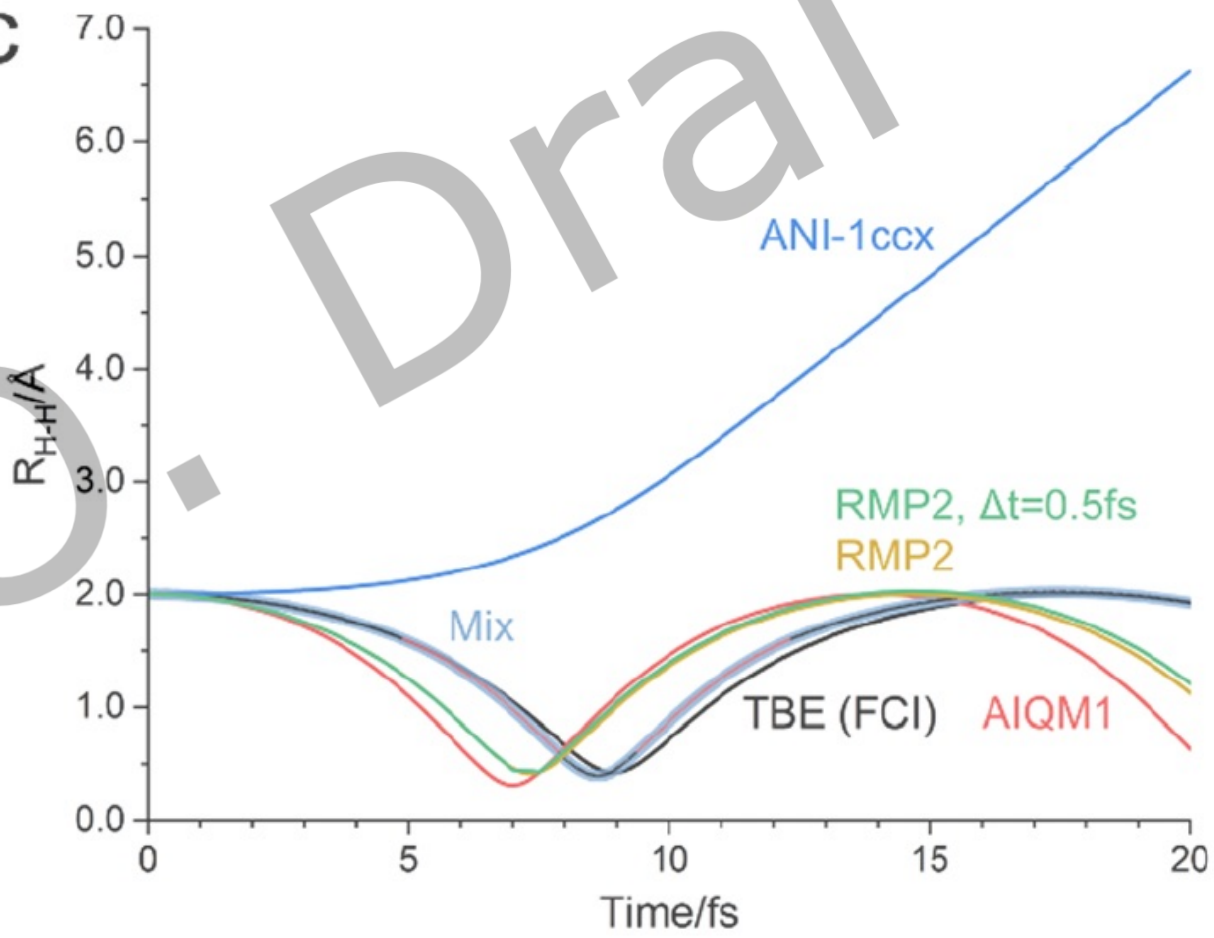
$$E_{\text{tot}} = E_{\text{pot}} + E_{\text{kin}}$$

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

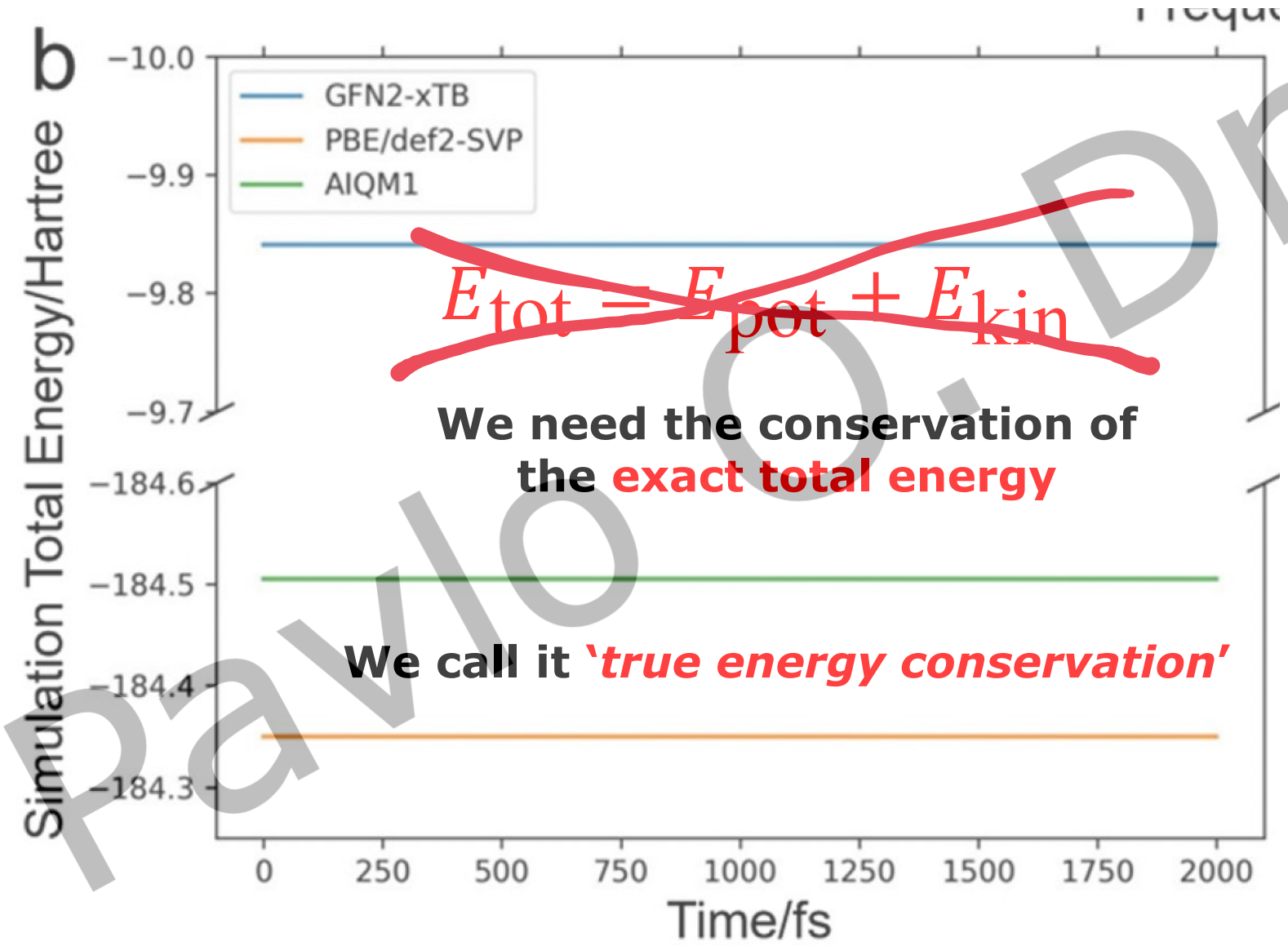
It does not mean it is good!

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

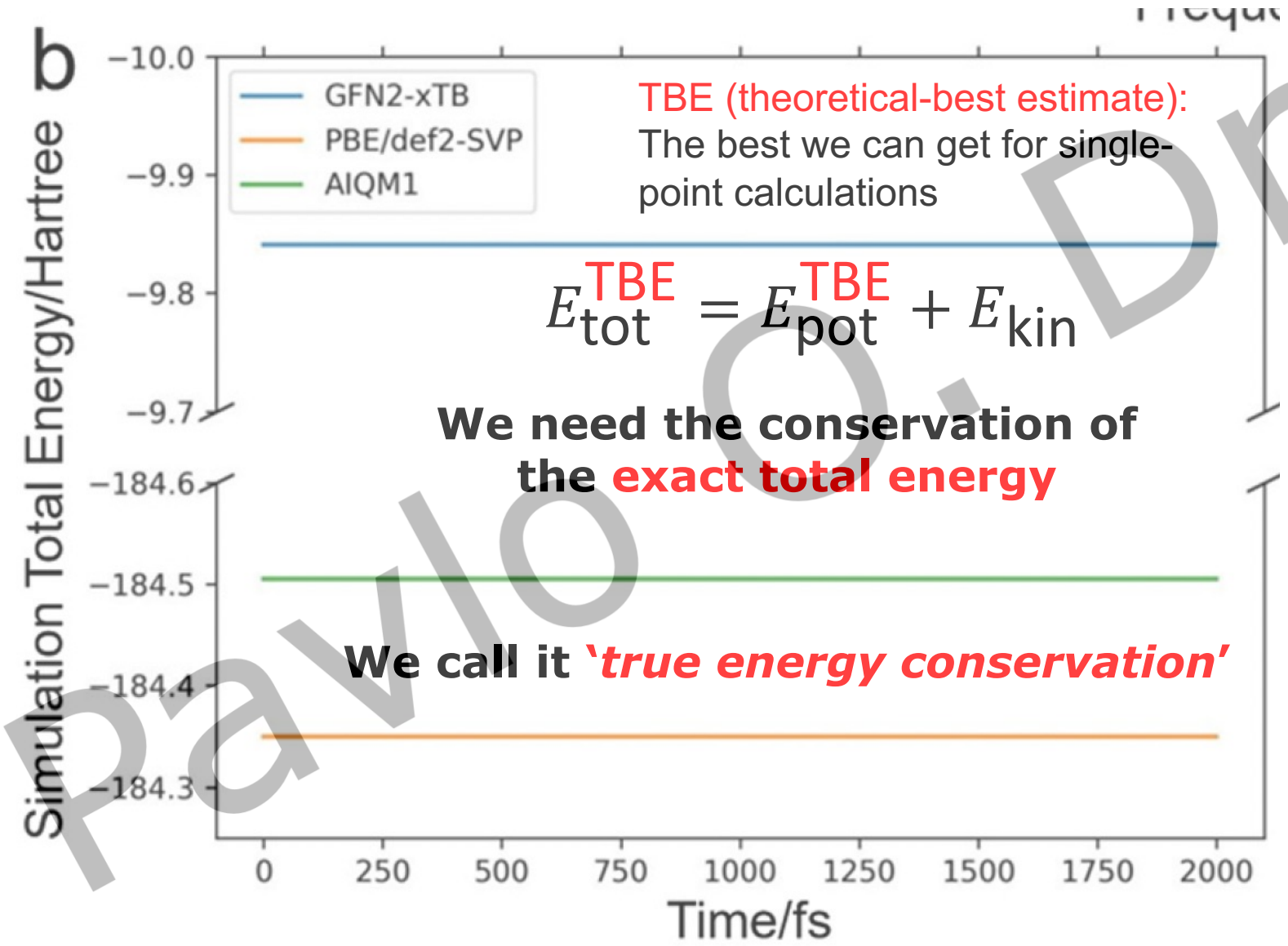


$$\cancel{E_{\text{tot}} = E_{\text{pot}} + E_{\text{kin}}}$$

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



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

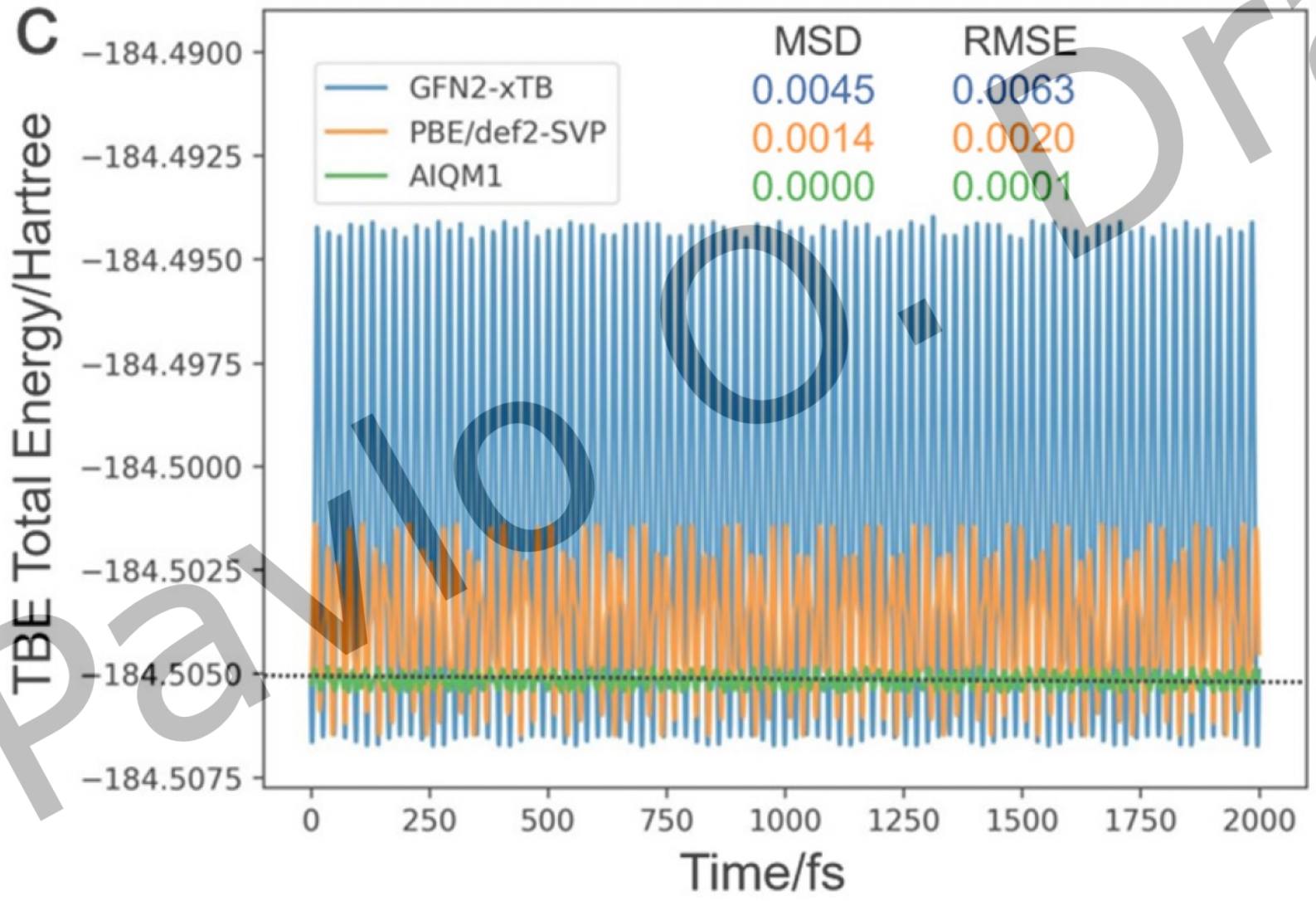
We call it *'true energy conservation'*

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

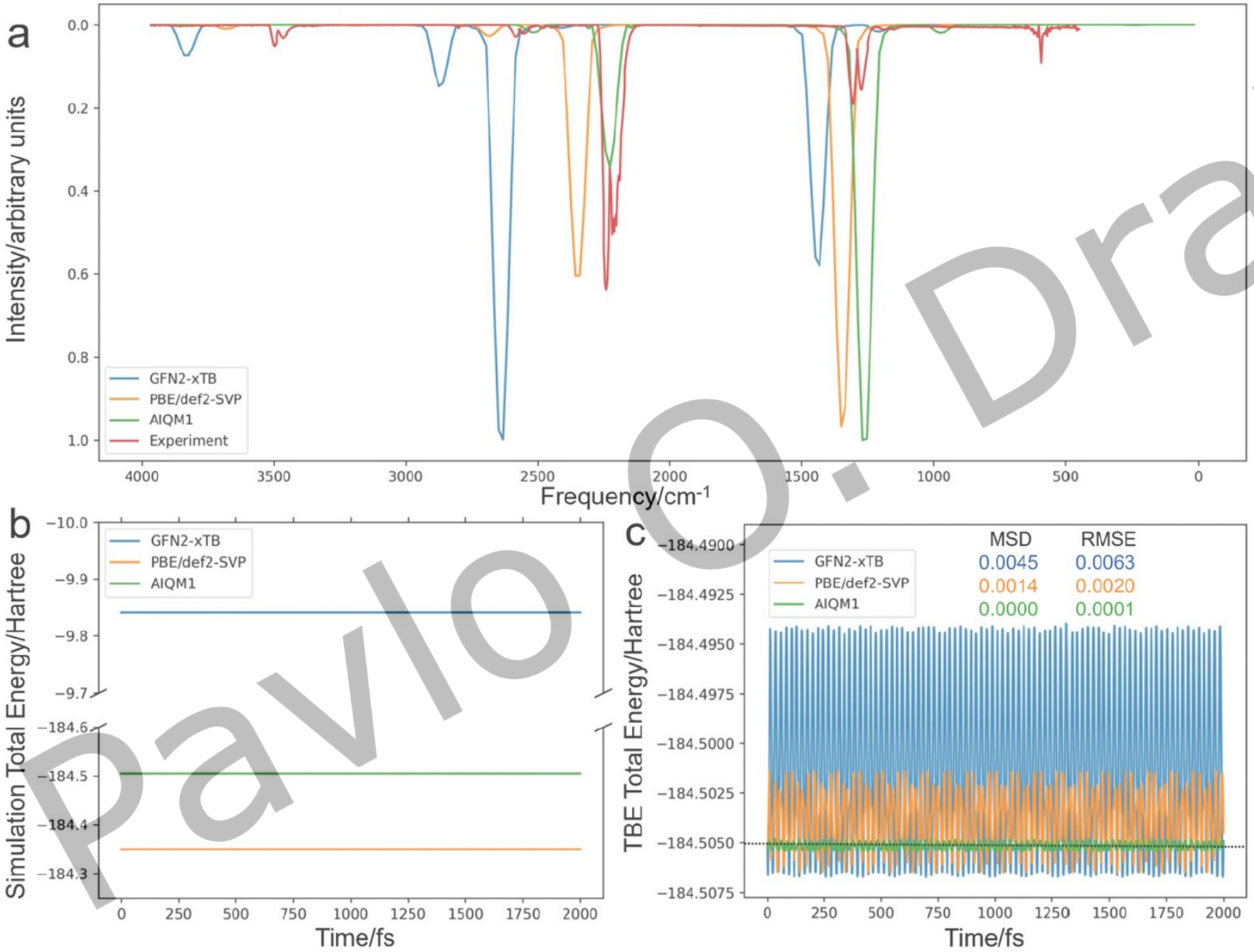
$$\text{RMSE} = \sqrt{1/\text{number of time steps} \sum \left(E_{\text{tot}}^{\text{TBE}}(\text{time step}) - E_{\text{tot}}^{\text{TBE}}(\text{time zero}) \right)^2}$$

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

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

Supervised Machine Learning

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

$$F_{A,d} = -\frac{\partial E}{\partial x_{A,d}}$$

Input (x) → *f*(x) → Output (y)

Given collection of known {x,y} find a function *f*(x)

training set train ML model

Use this function for making new predictions given just {x'}

- **Data**
- Choice of x (descriptor)
- Choice of y (labels)
- Fitting function (ML algorithm, ML model)
- Optimization of ML model parameters

Categories of machine learning potentials

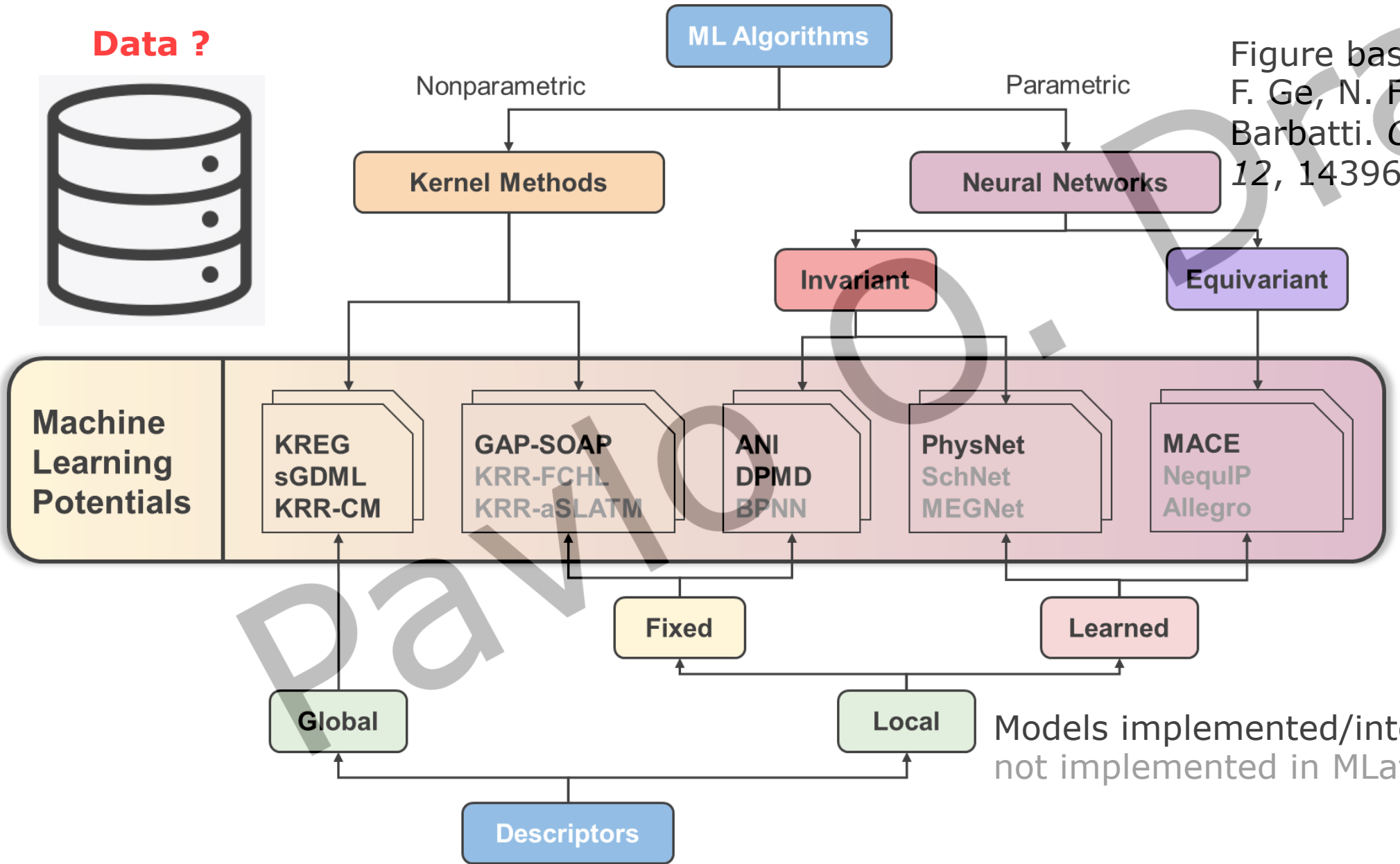
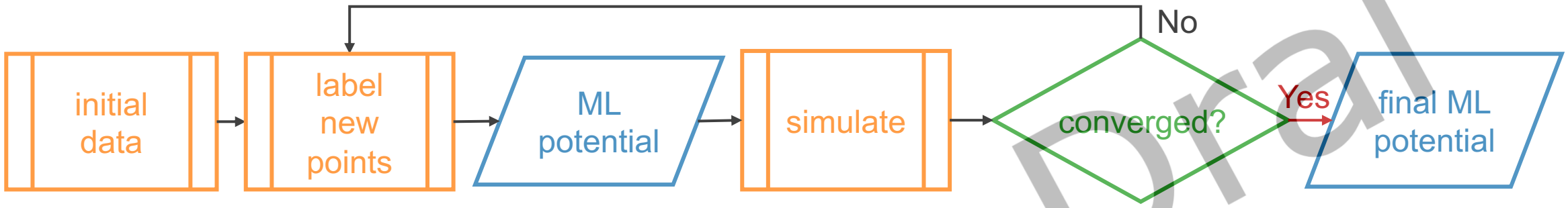


Figure based on M. Pinheiro Jr, F. Ge, N. Ferré, P. O. Dral, M. Barbatti. *Chem. Sci.* **2021**, *12*, 14396–14413

Models implemented/interfaced in MLatom
not implemented in MLatom

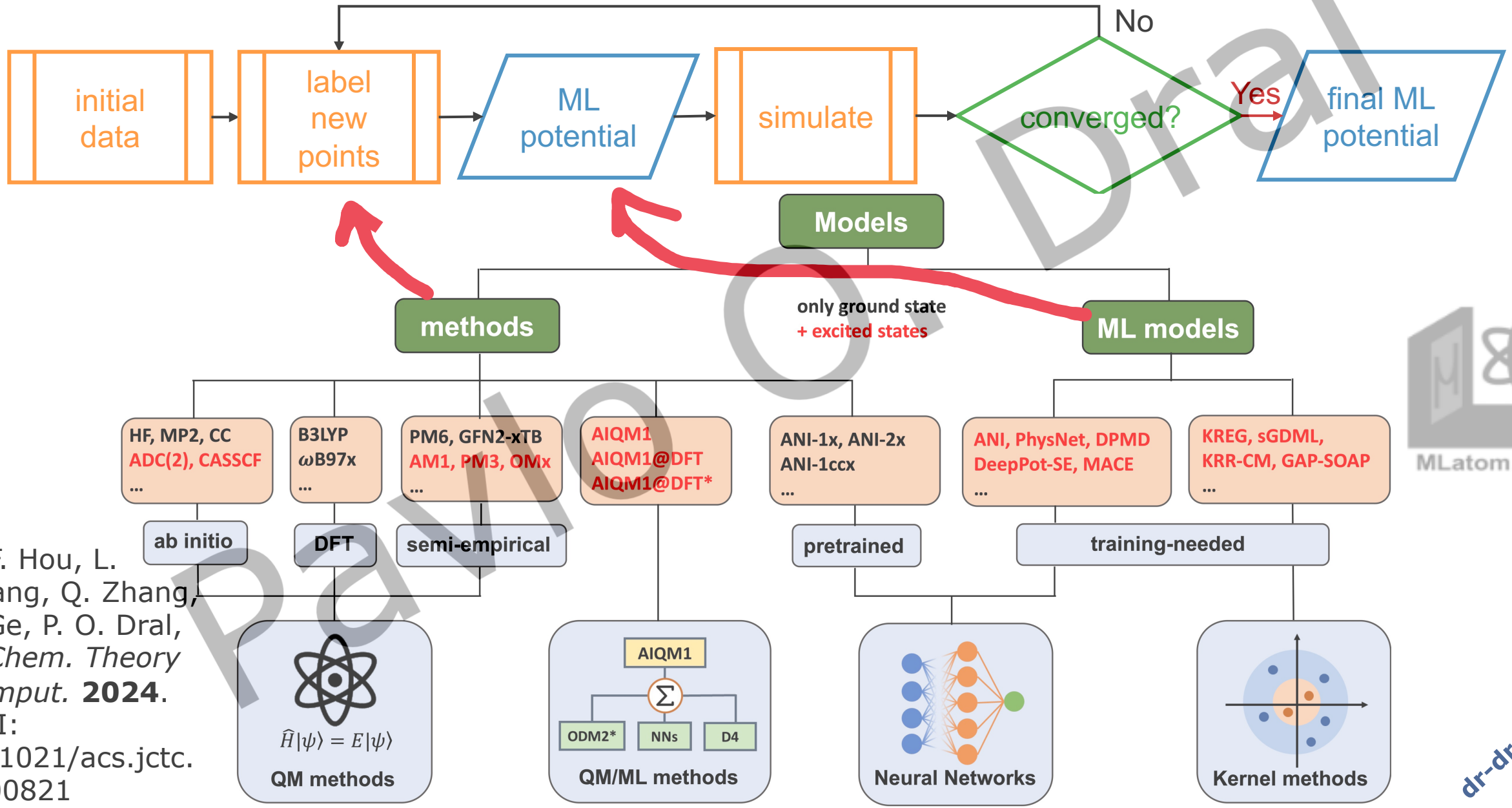
Active learning



Each step is highly non-trivial and you need methods & software!

Figure from: Y.-F. Hou, L. Zhang, Q. Zhang, F. Ge, P. O. Dral, *J. Chem. Theory Comput.* **2024**. DOI: 10.1021/acs.jctc.4c00821

Active learning



Y.-F. Hou, L. Zhang, Q. Zhang, F. Ge, P. O. Dral, *J. Chem. Theory Comput.* **2024**. DOI: 10.1021/acs.jctc.4c00821

Active learning

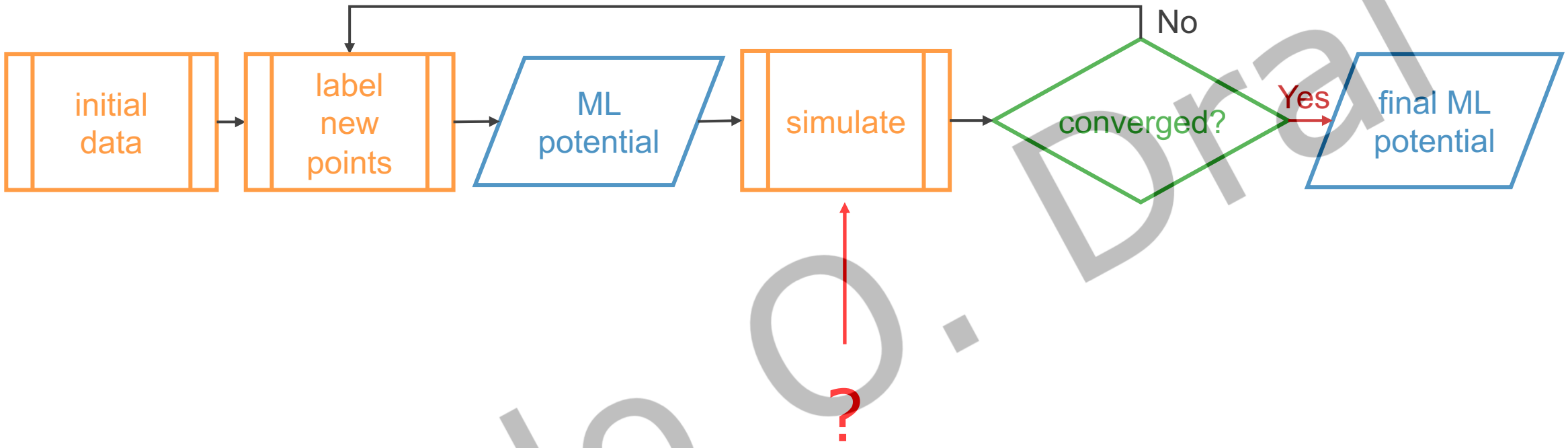
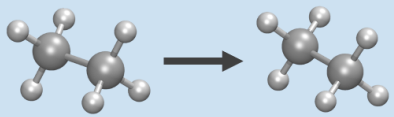


Figure from: Y.-F. Hou, L. Zhang, Q. Zhang, F. Ge, P. O. Dral, *J. Chem. Theory Comput.* **2024**. DOI: 10.1021/acs.jctc.4c00821

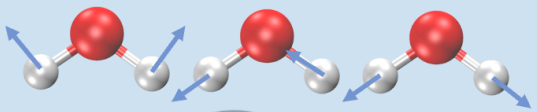
MLatom ecosystem

Single point calculations — Energies, forces, Hessian matrix…

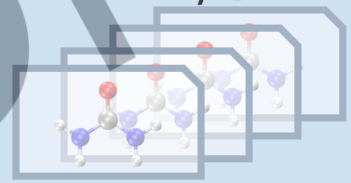
Geometry optimizations



Frequency calculations



Molecular dynamics

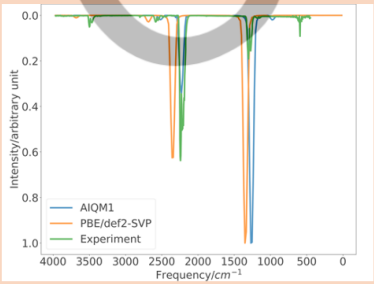


Thermochemistry calculations

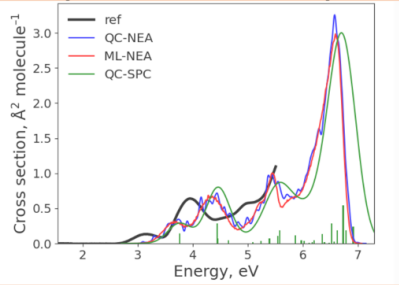
Heat of formation:

$$\Delta H_{\mathrm{at},T} = \left[\sum_A H_T(A)\right] - H_T$$

(Ro)vibrational spectra



One-photon UV/vis spectra



Simulations

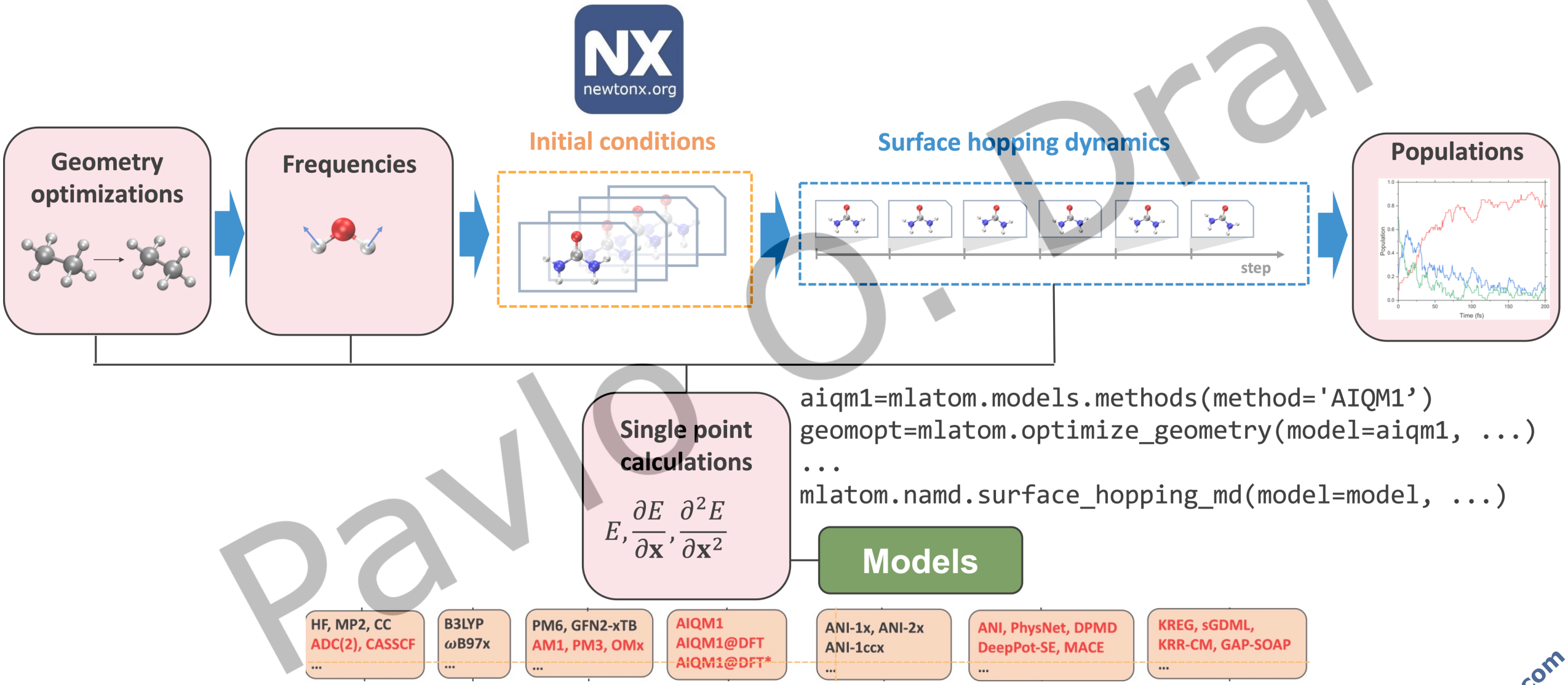
Properties & spectra

Quantum dissipative dynamics

Two-photon absorption spectra

P. O. Dral, F. Ge, Y.-F. Hou, P. Zheng, Y. Chen, M. Barbatti, O. Isayev, C. Wang, B.-X. Xue, M. Pinheiro Jr, Y. Su, Y. Dai, Y. Chen, S. Zhang, L. Zhang, A. Ullah, Q. Zhang, Y. Ou. *J. Chem. Theory Comput.* **2024,** *20*, 1193

Surface-hopping dynamics




```
aiqm1=mlatom.models.methods(method='AIQM1’)
geomopt=mlatom.optimize_geometry(model=aiqm1, ...)
...
mlatom.namd.surface_hopping_md(model=model, ...)
```

L. Zhang, S. Pios, M. Martyka, F. Ge, Y. F. Hou, Y. Chen, L. Chen, J. Jankowska, M. Barbatti, P. O. Dral. *JCTC*, **2024**, *20*, 5043–5057

Active learning

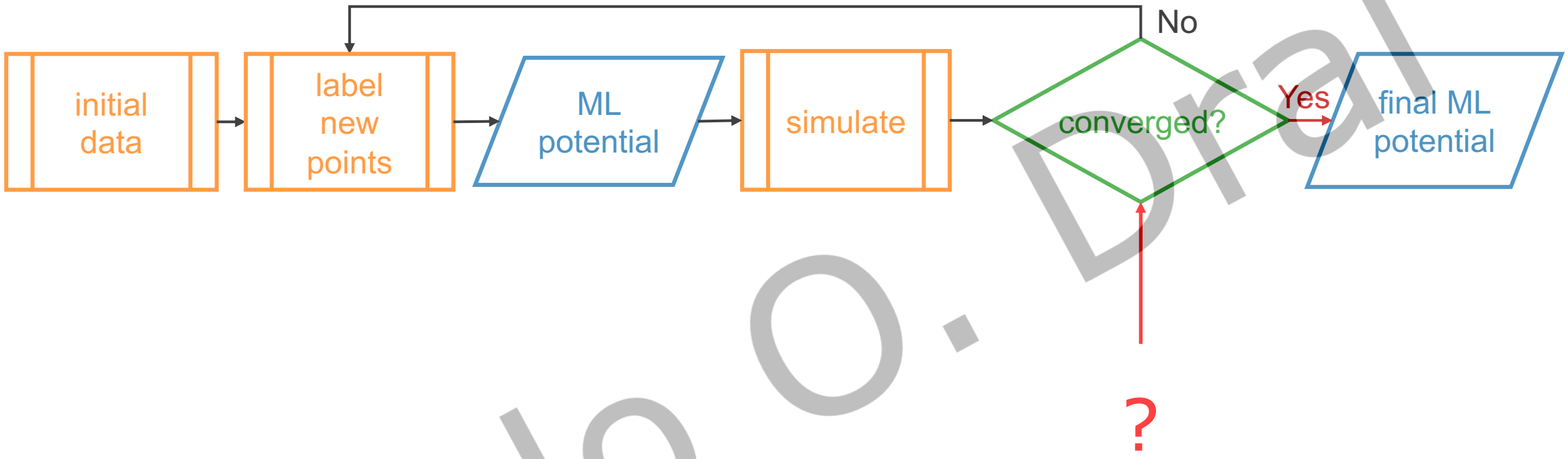
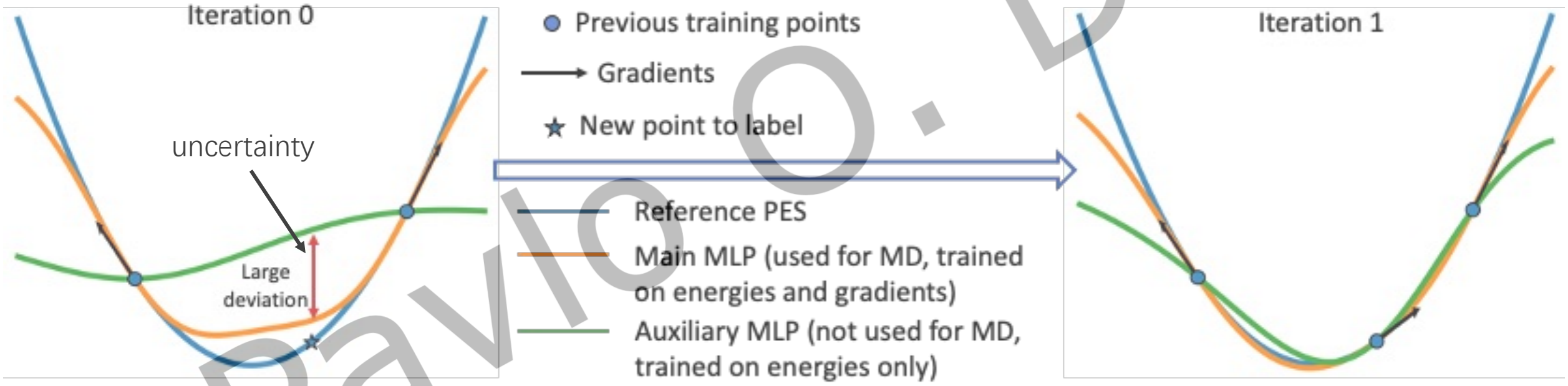


Figure from: Y.-F. Hou, L. Zhang, Q. Zhang, F. Ge, P. O. Dral, *J. Chem. Theory Comput.* **2024**. DOI: 10.1021/acs.jctc.4c00821

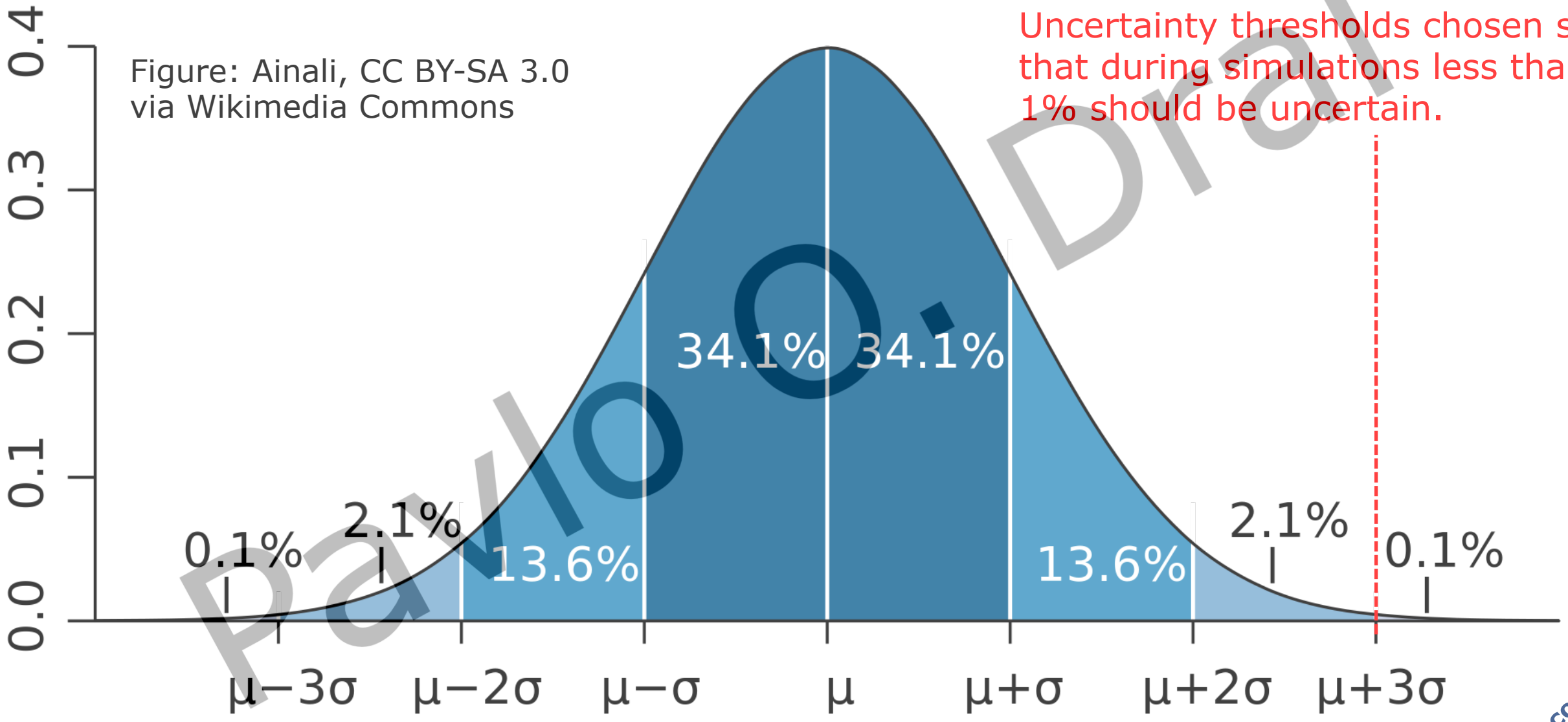
Physics-informed active learning

Utilizing different amount of physics-derived information for uncertainty quantification



Y.-F. Hou, L. Zhang, Q. Zhang, F. Ge, P. O. Dral. *arXiv:2404.11811*.

Automatic uncertainty quantification



Y.-F. Hou, L. Zhang, Q. Zhang, F. Ge, P. O. Dral. *arXiv:2404.11811*.

Active learning

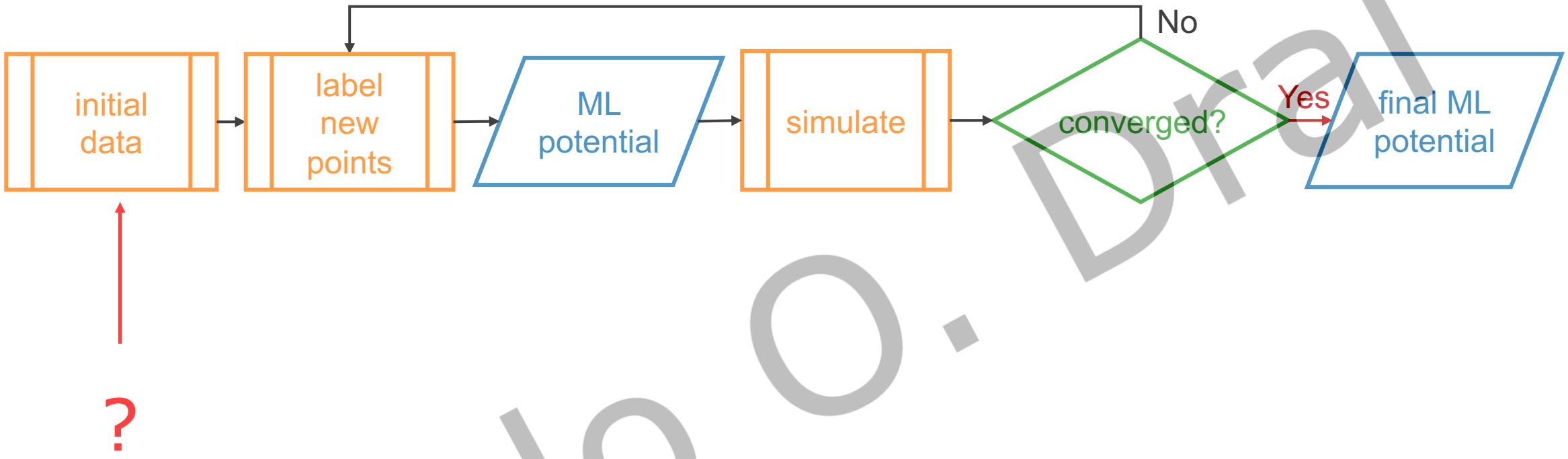
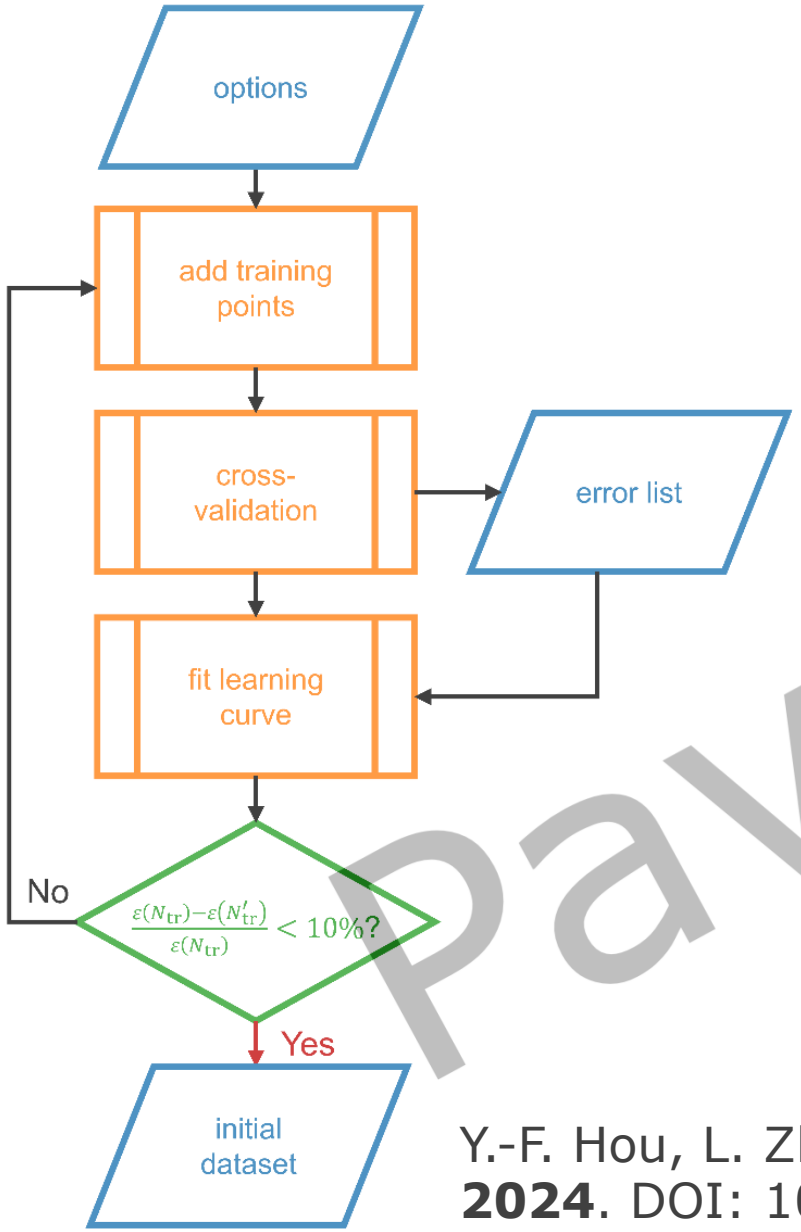
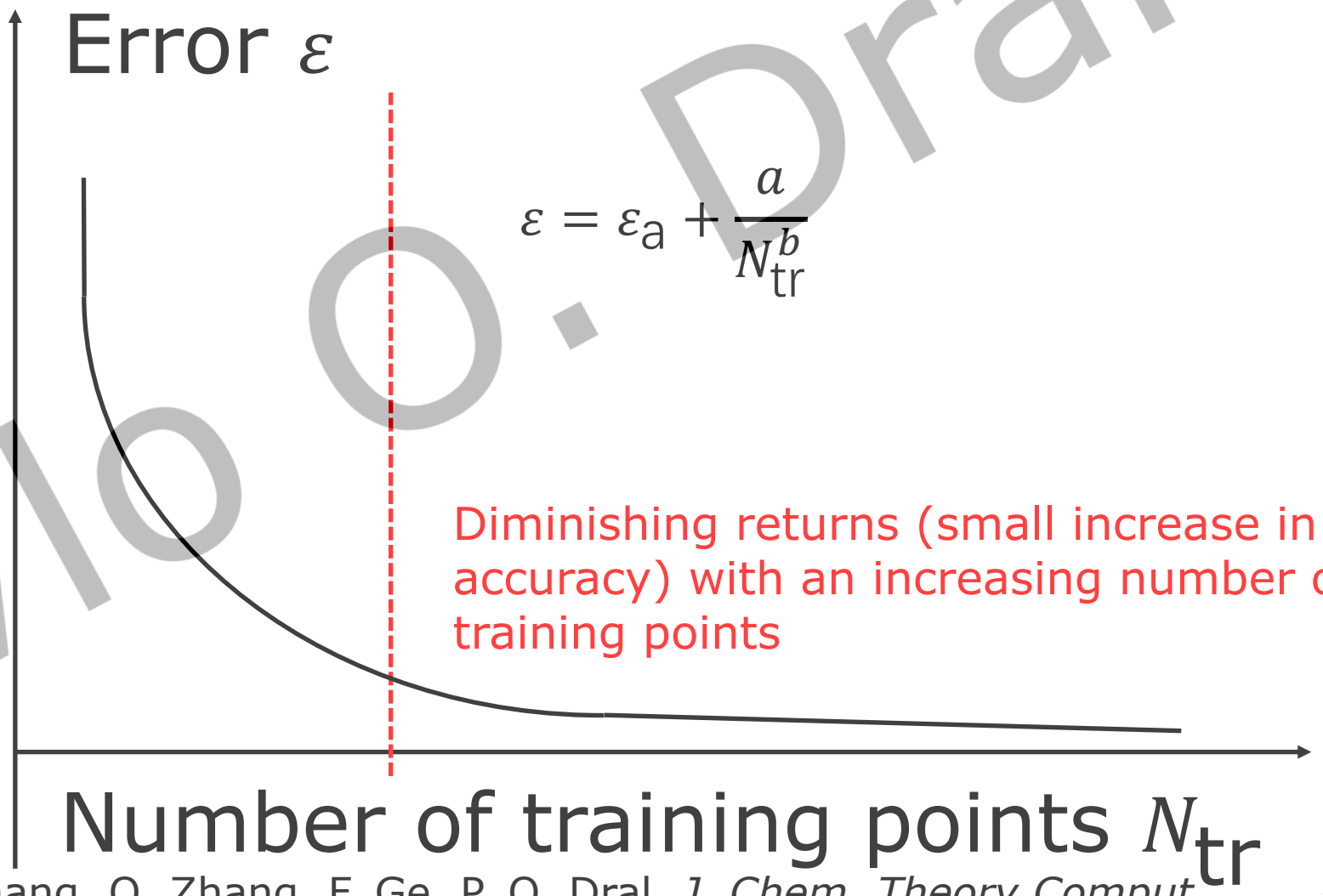


Figure from: Y.-F. Hou, L. Zhang, Q. Zhang, F. Ge, P. O. Dral, *J. Chem. Theory Comput.* **2024**. DOI: 10.1021/acs.jctc.4c00821

Automatic initial data building





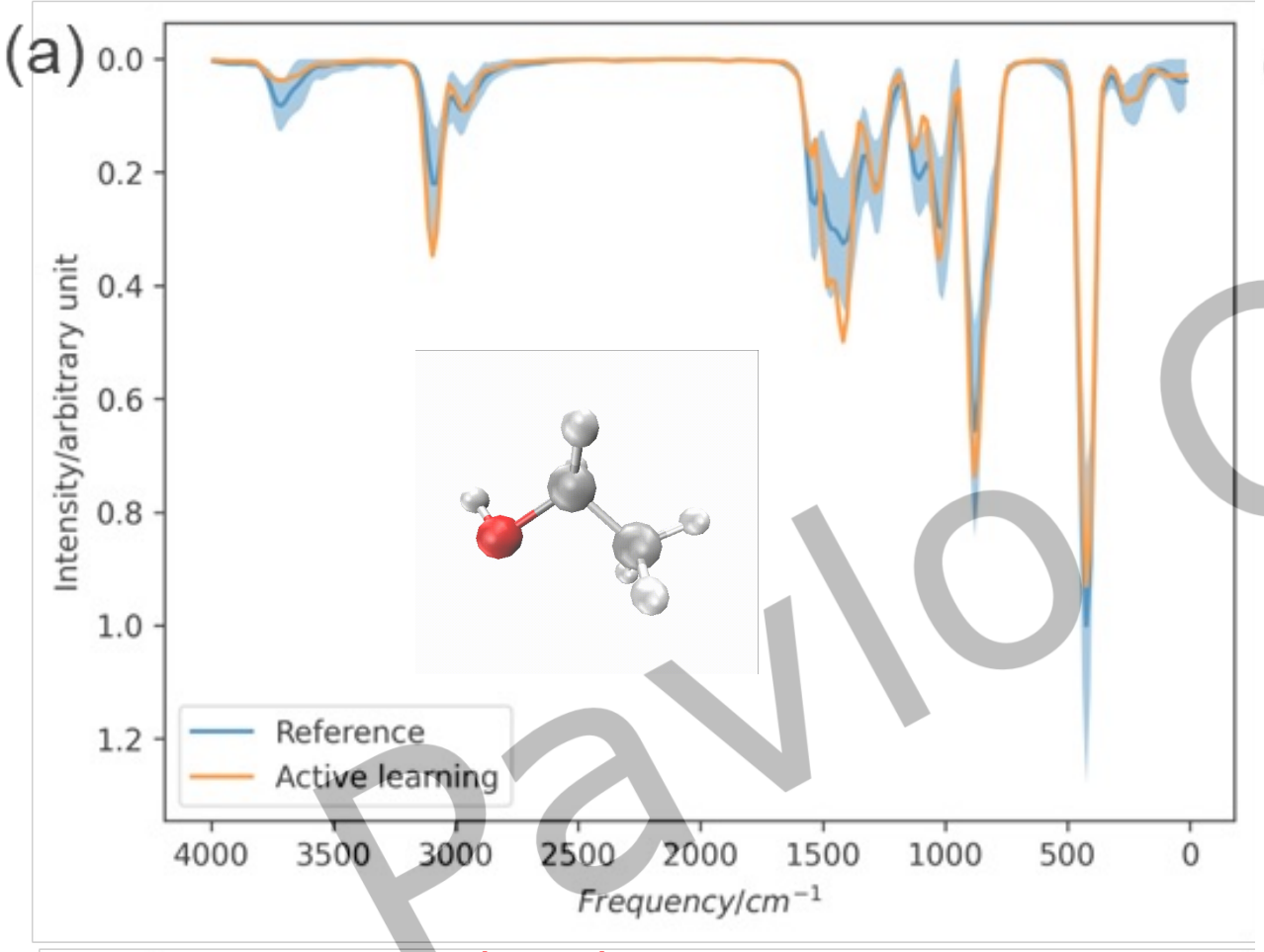
$$\varepsilon = \varepsilon_a + \frac{a}{N_{tr}^b}$$

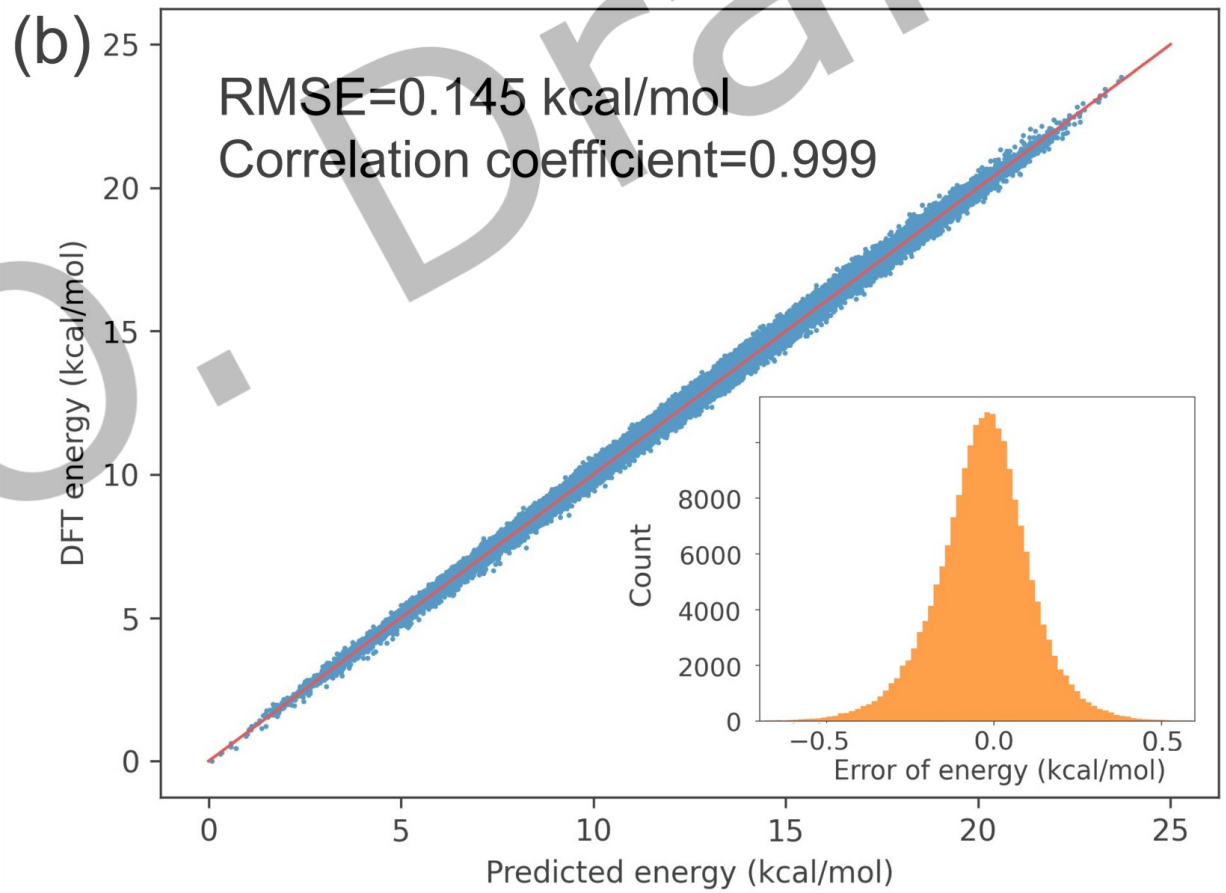
Diminishing returns (small increase in accuracy) with an increasing number of training points

Y.-F. Hou, L. Zhang, Q. Zhang, F. Ge, P. O. Dral, *J. Chem. Theory Comput.* **2024**. DOI: 10.1021/acs.jctc.4c00821

Accurate vibrational spectra

< 1000 training points with ANI potential



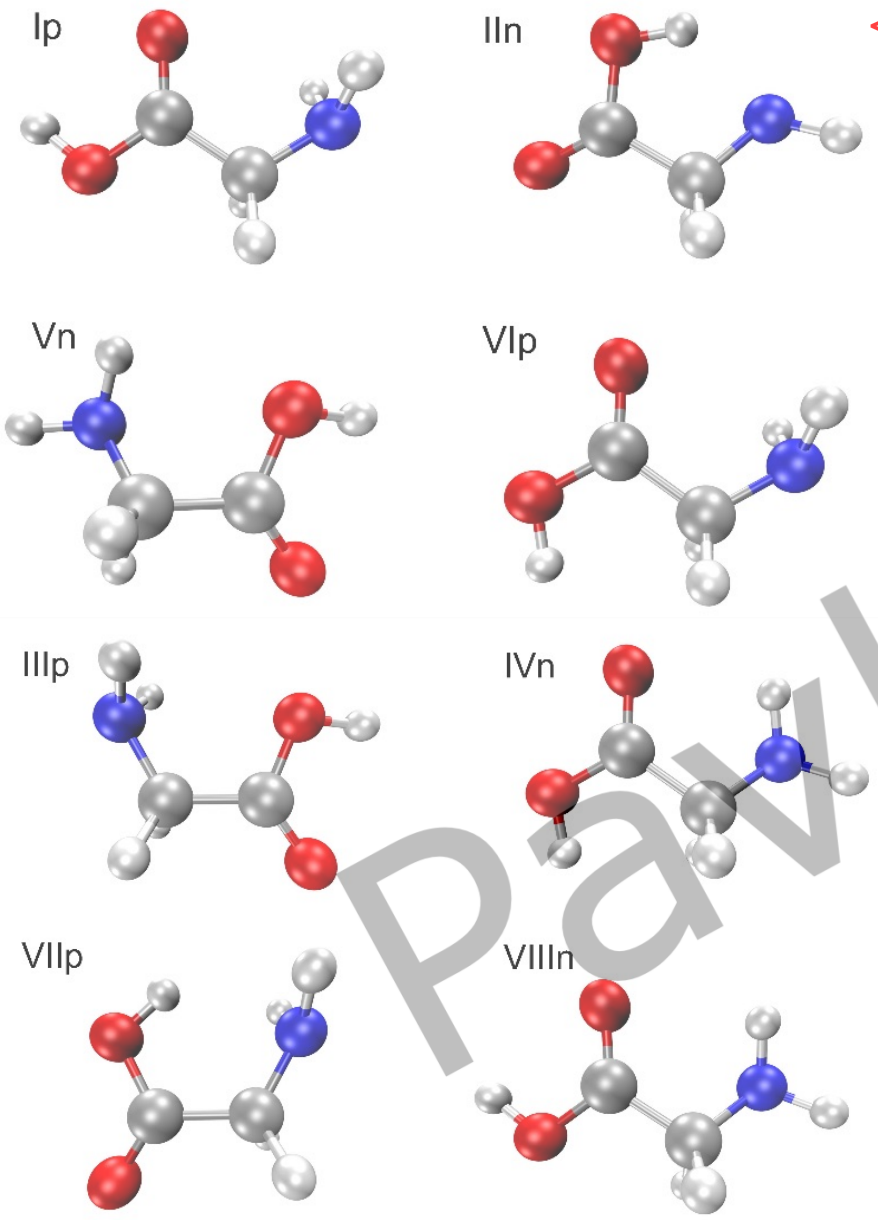


Ethanol spectra

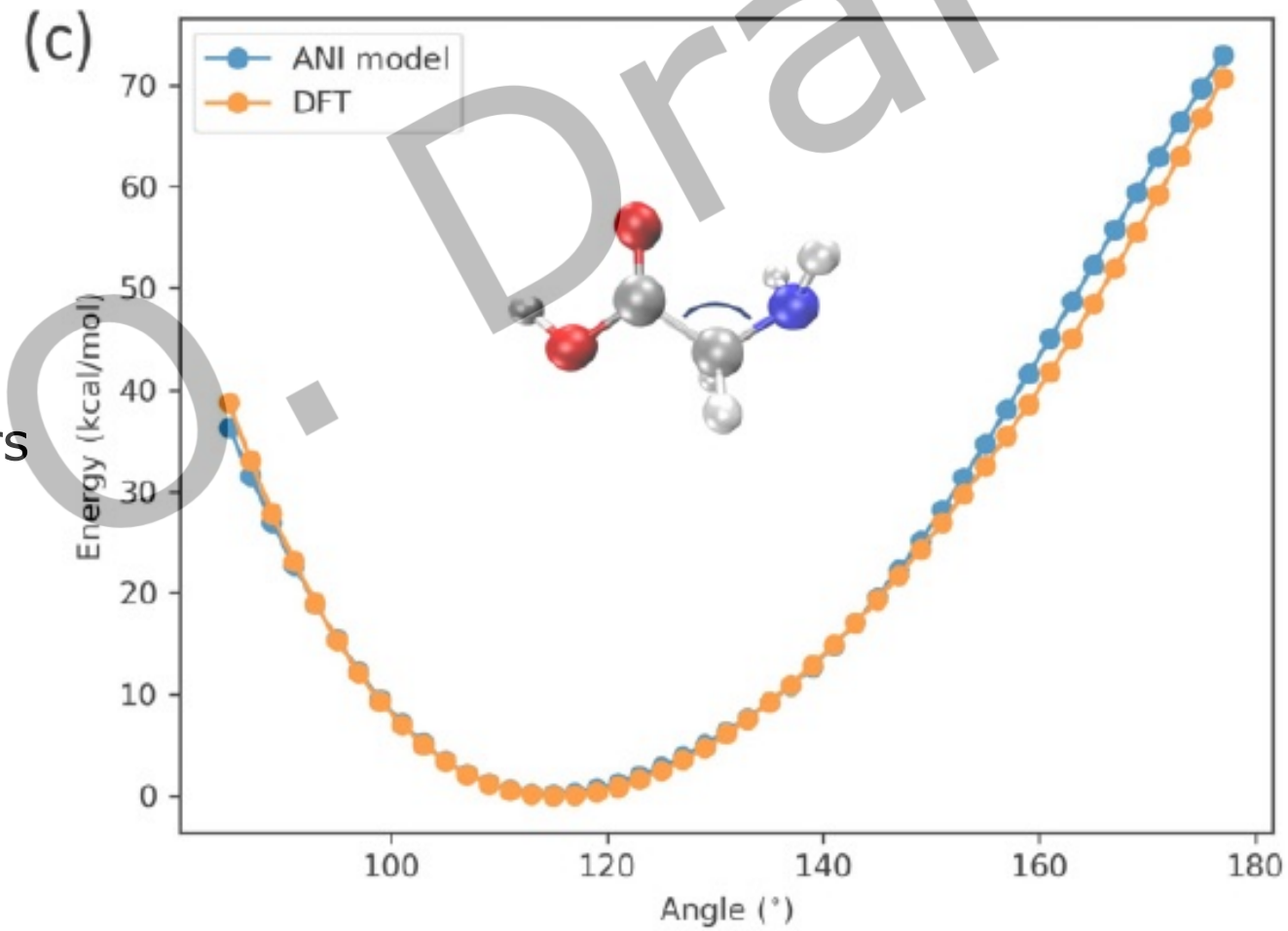
Figure from: Y.-F. Hou, L. Zhang, Q. Zhang, F. Ge, P. O. Dral, *J. Chem. Theory Comput.* **2024**. DOI: 10.1021/acs.jctc.4c00821

Conformer search through quasi-classical MD

< 2000 training points with ANI potential



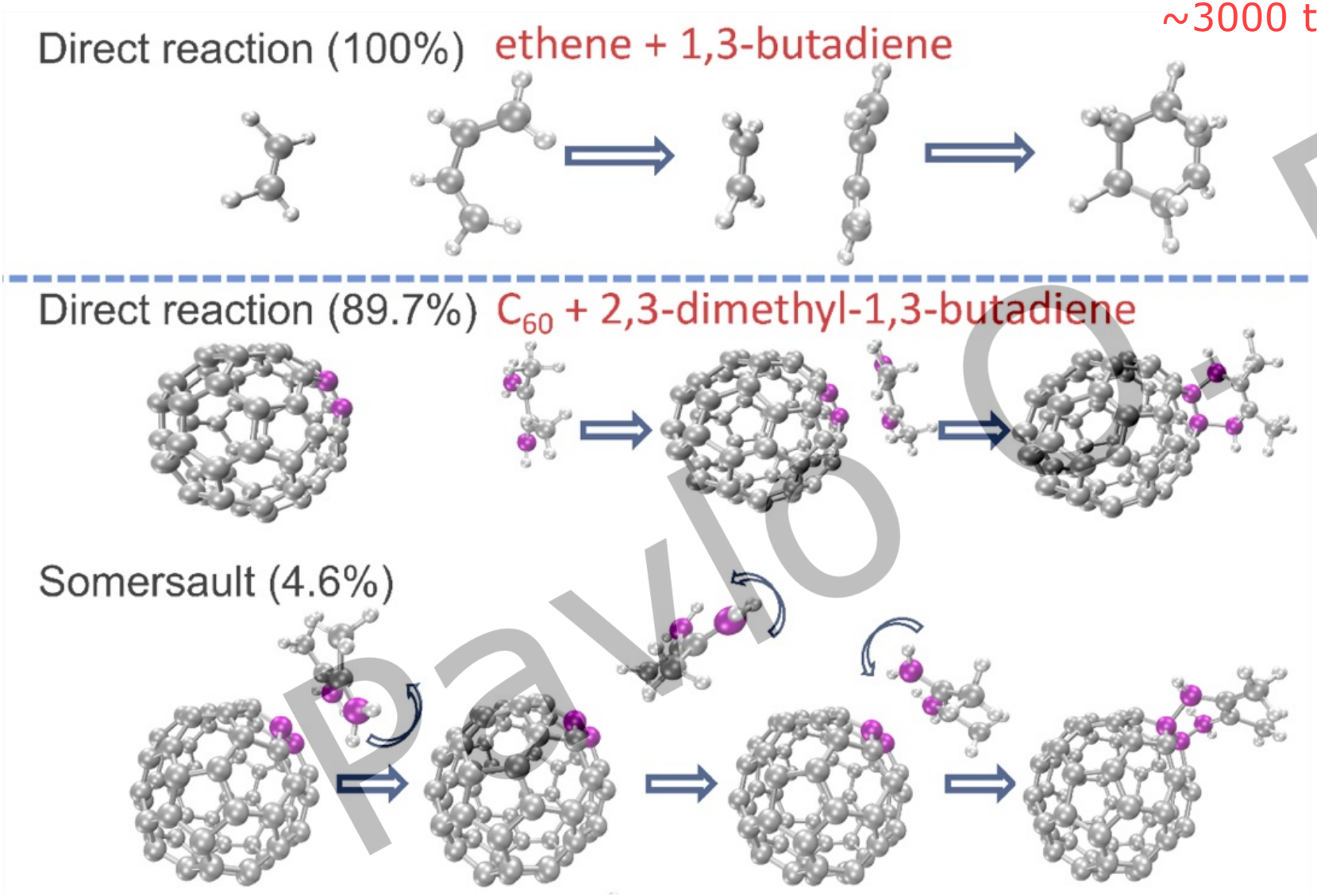
Glycine
8
conformers

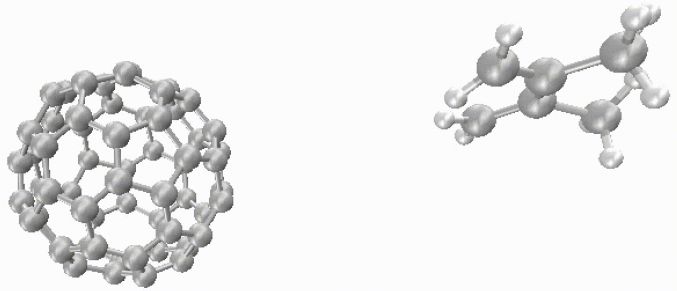


Y.-F. Hou, L. Zhang, Q. Zhang, F. Ge, P. O. Dral, *J. Chem. Theory Comput.* **2024**. DOI: 10.1021/acs.jctc.4c00821

Time-resolved mechanisms

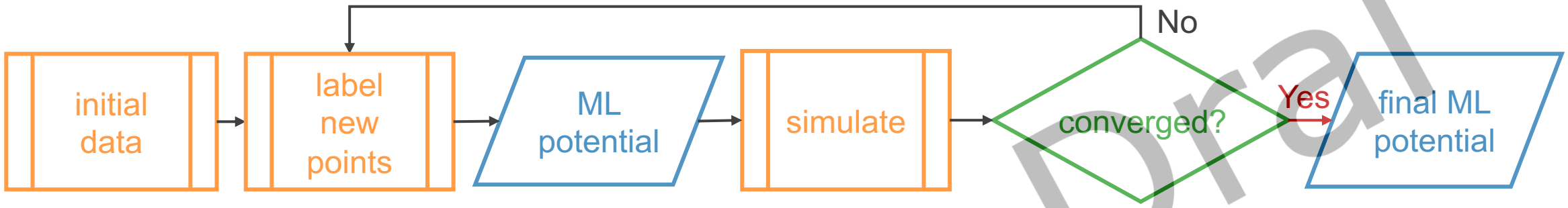
~3000 training points with ANI potential





Y.-F. Hou, Q. Zhang, P. O. Dral. https://doi.org/10.26434/chemrxiv-2024-hwsl4.

Active learning

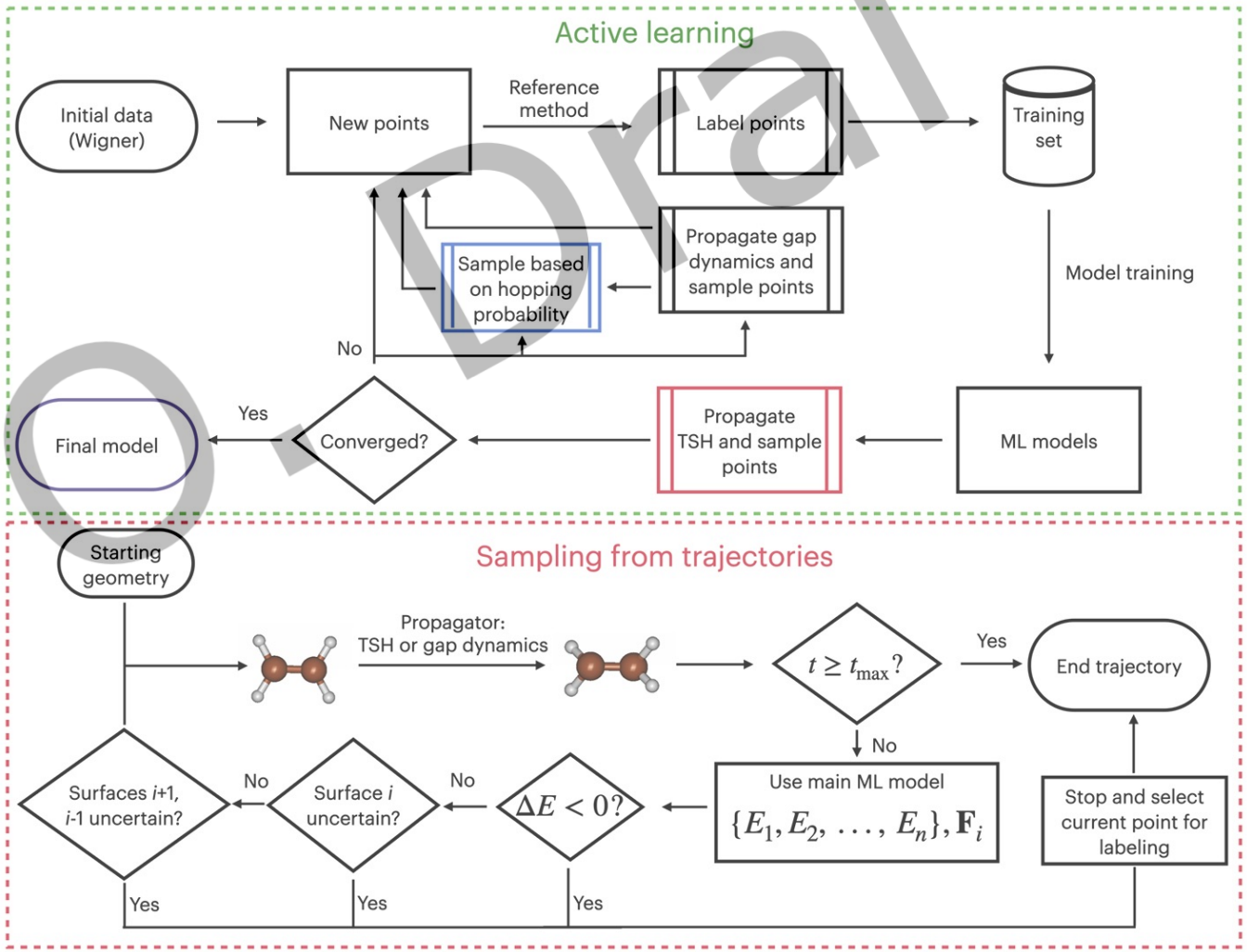


Does it work for NAMD?

Figure from: Y.-F. Hou, L. Zhang, Q. Zhang, F. Ge, P. O. Dral, *J. Chem. Theory Comput.* **2024**. DOI: 10.1021/acs.jctc.4c00821

Yes, but in more complex workflows

Not so simple for excited states:



M. Martyka, L. Zhang, F. Ge, Y.-F. Hou, J. Jankowska, M. Barbatti, P. O. Dral. *Charting electronic-state manifolds across molecules with multi-state learning and gap-driven dynamics via efficient and robust active learning*. https://doi.org/10.26434/chemrxiv-2024-dtc1w

Active learning

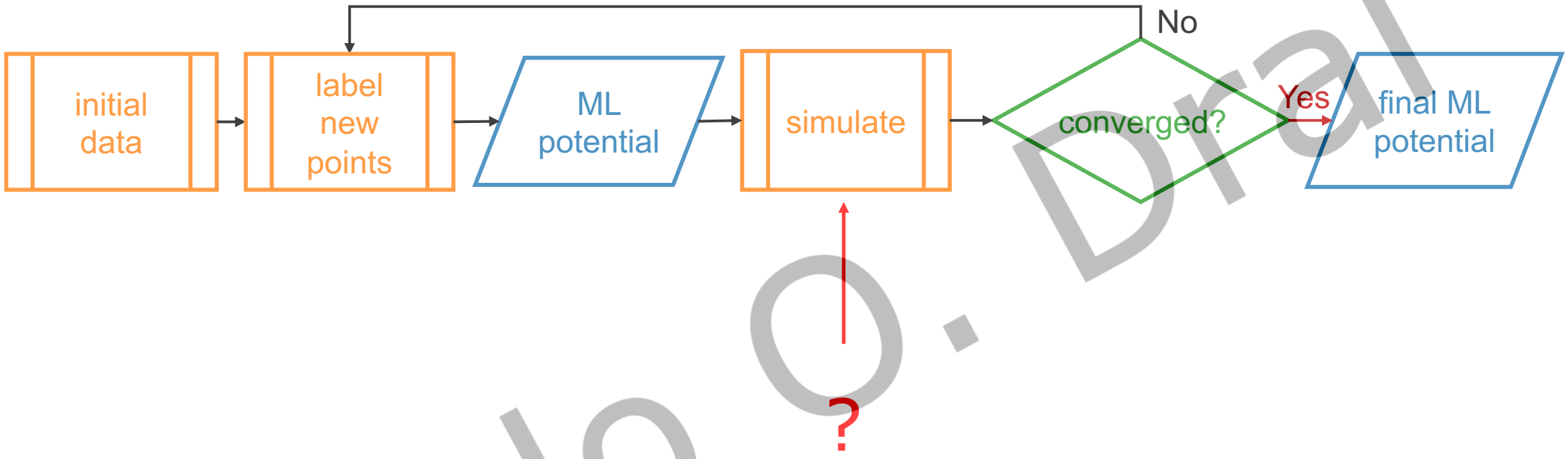
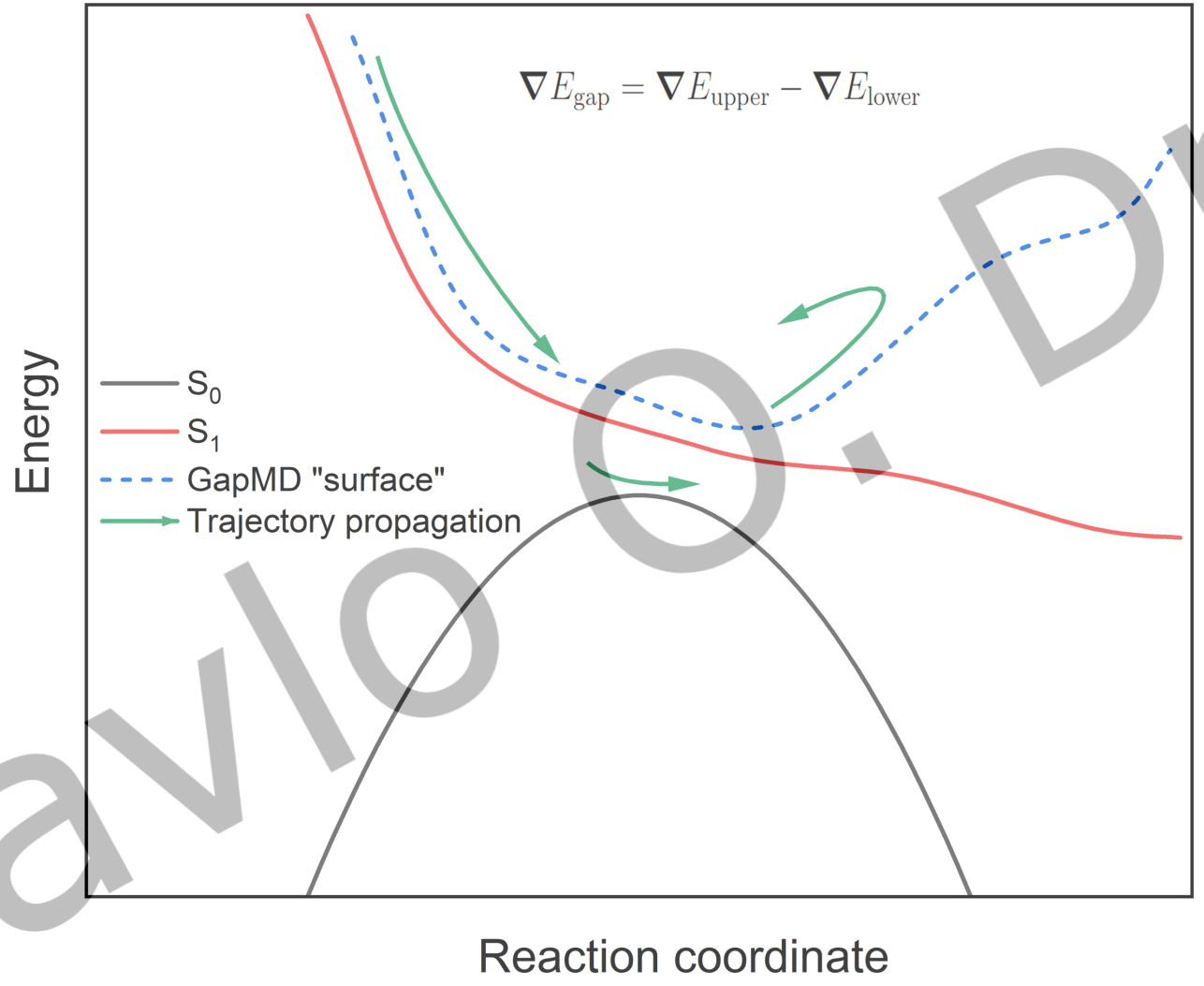
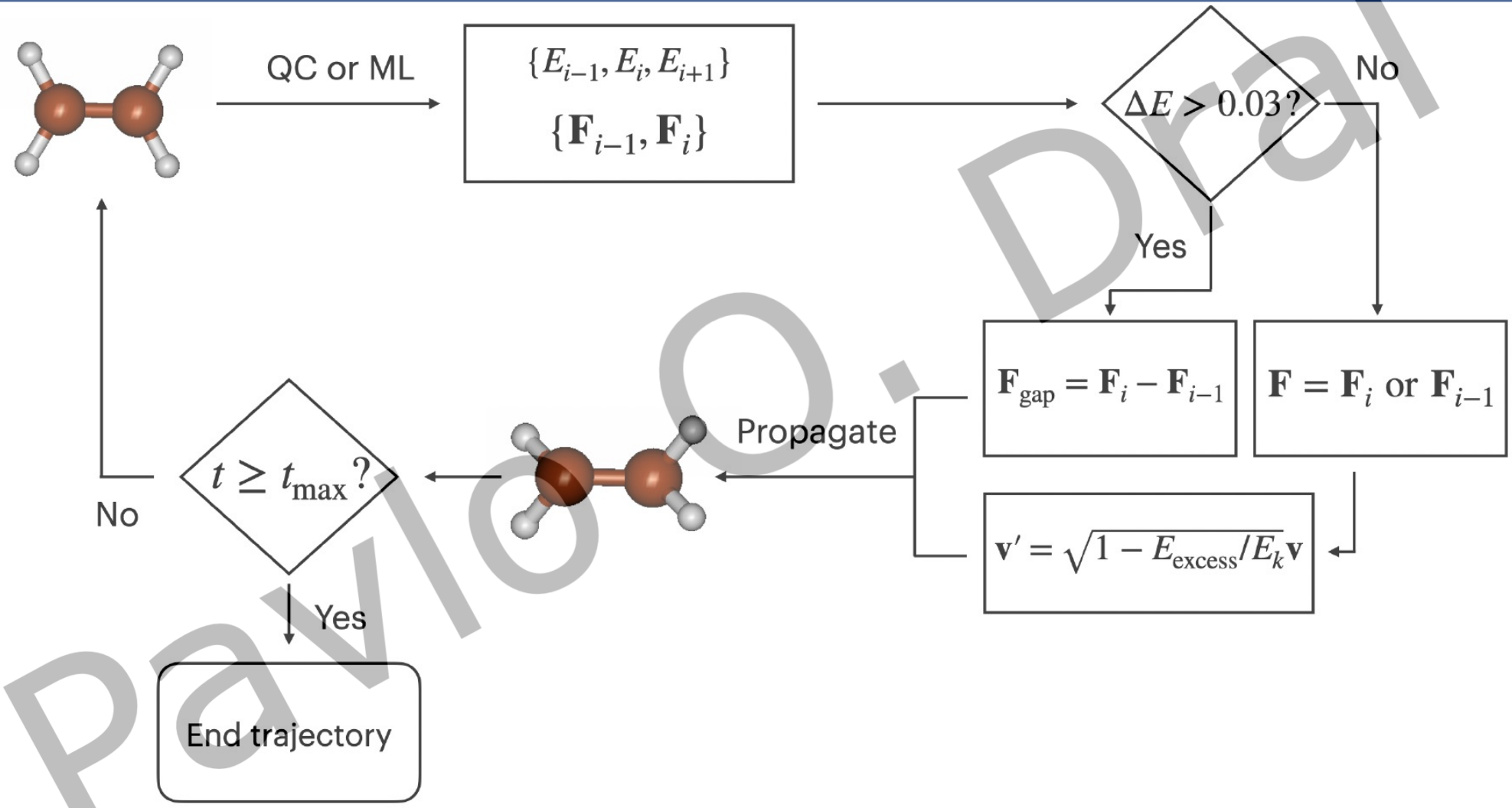


Figure from: Y.-F. Hou, L. Zhang, Q. Zhang, F. Ge, P. O. Dral, *J. Chem. Theory Comput.* **2024**. DOI: 10.1021/acs.jctc.4c00821

New: GapMD!



M. Martyka, L. Zhang, F. Ge, Y.-F. Hou, J. Jankowska, M. Barbatti, P. O. Dral. *Charting electronic-state manifolds across molecules with multi-state learning and gap-driven dynamics via efficient and robust active learning*. https://doi.org/10.26434/chemrxiv-2024-dtc1w



M. Martyka, L. Zhang, F. Ge, Y.-F. Hou, J. Jankowska, M. Barbatti, P. O. Dral. *Charting electronic-state manifolds across molecules with multi-state learning and gap-driven dynamics via efficient and robust active learning*. https://doi.org/10.26434/chemrxiv-2024-dtc1w

Active learning

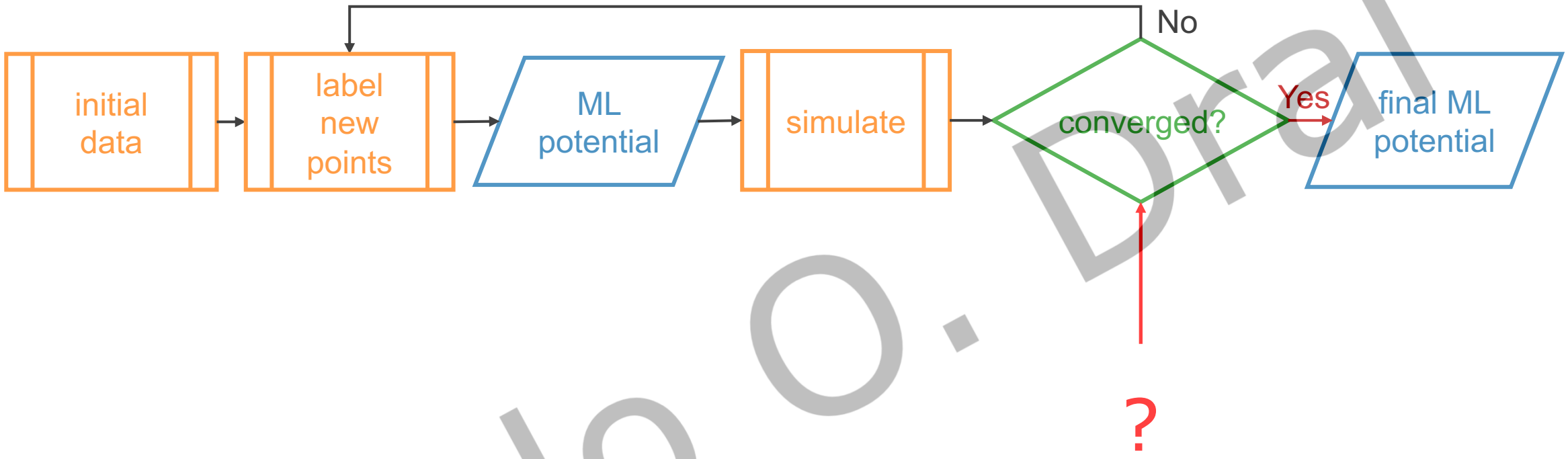
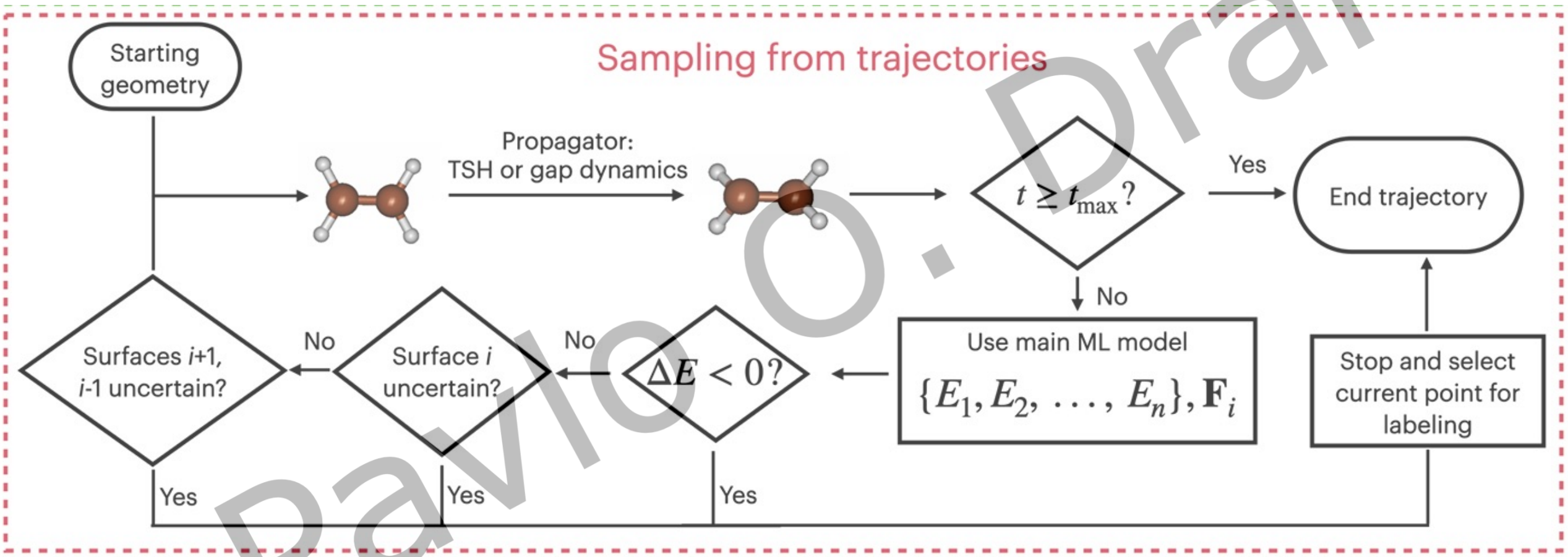


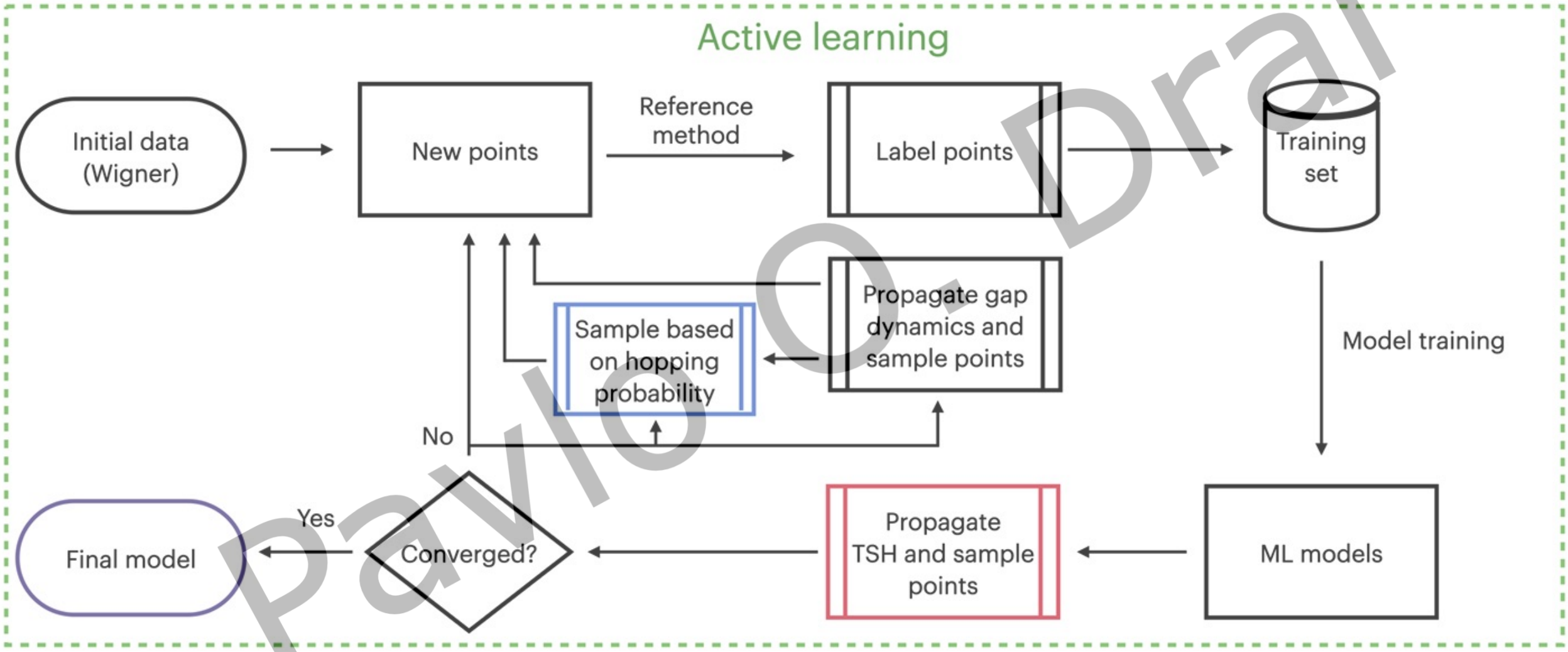
Figure from: Y.-F. Hou, L. Zhang, Q. Zhang, F. Ge, P. O. Dral, *J. Chem. Theory Comput.* **2024**. DOI: 10.1021/acs.jctc.4c00821

Physics-based sampling for NAMD



M. Martyka, L. Zhang, F. Ge, Y.-F. Hou, J. Jankowska, M. Barbatti, P. O. Dral. *Charting electronic-state manifolds across molecules with multi-state learning and gap-driven dynamics via efficient and robust active learning*. https://doi.org/10.26434/chemrxiv-2024-dtc1w

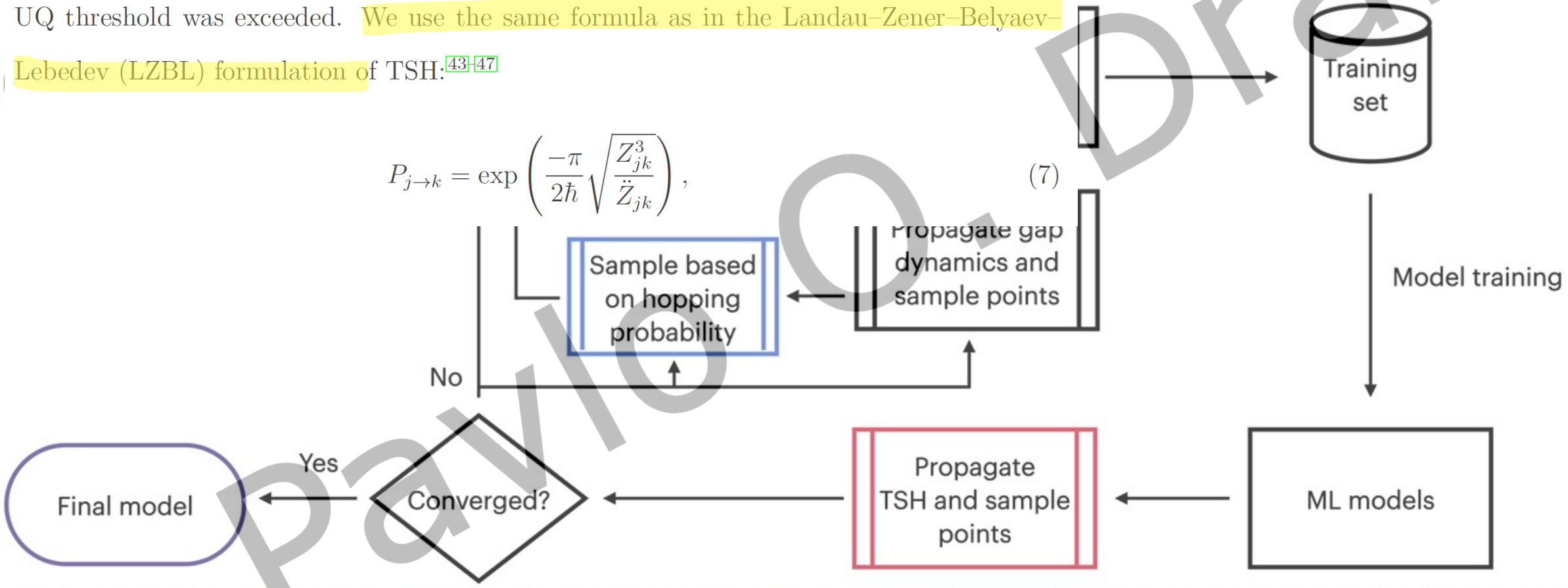
Physics-based sampling for NAMD



M. Martyka, L. Zhang, F. Ge, Y.-F. Hou, J. Jankowska, M. Barbatti, P. O. Dral. *Charting electronic-state manifolds across molecules with multi-state learning and gap-driven dynamics via efficient and robust active learning*. https://doi.org/10.26434/chemrxiv-2024-dtc1w

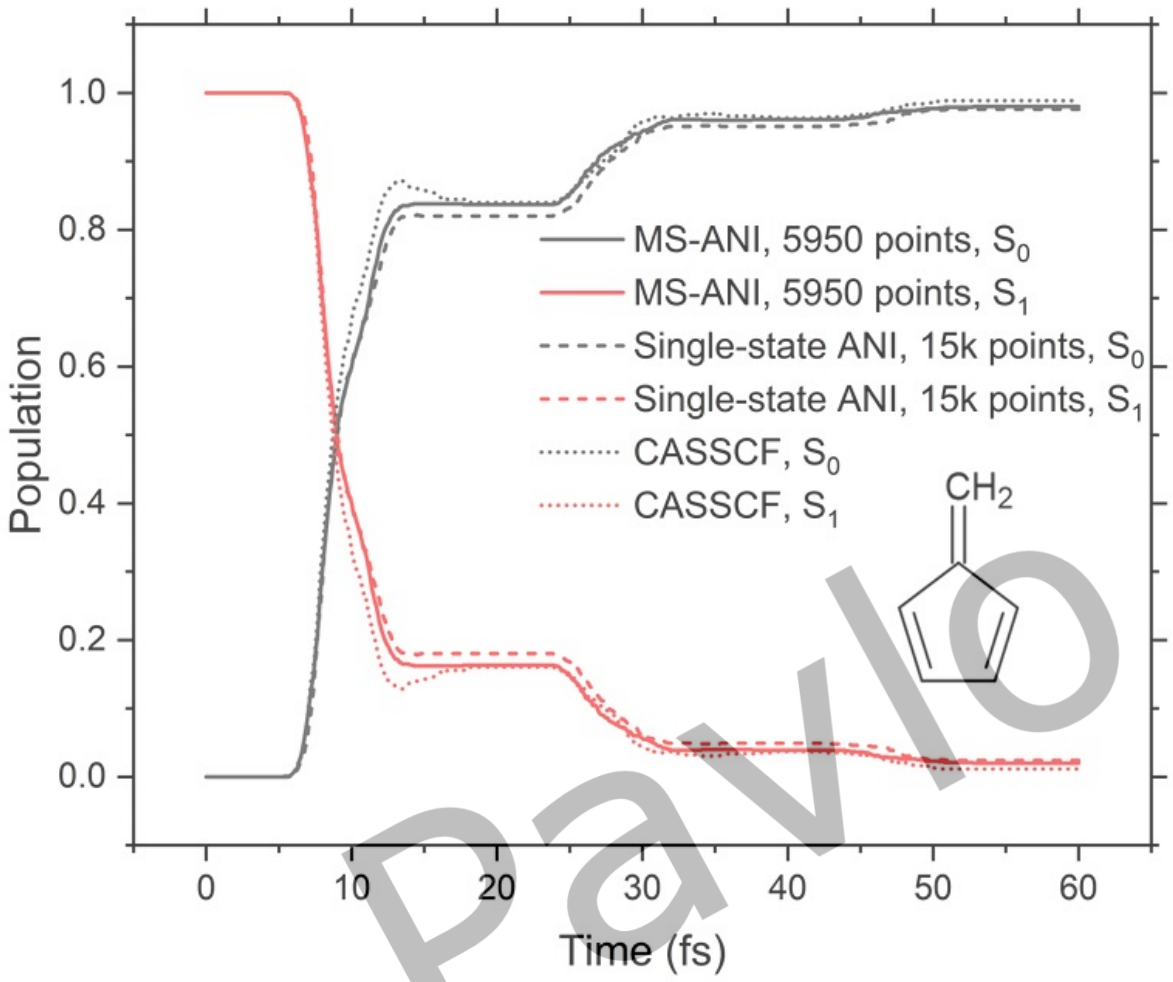
the robust performance in TSH. To further refine the model in terms of hopping probability, after the propagation of each TSH trajectory, we evaluate their uncertainties. The UQ metrics are computed as absolute deviations between the hopping probabilities evaluated using the energies predicted by the main, and auxiliary models at each time step before the UQ threshold was exceeded. We use the same formula as in the Landau–Zener–Belyaev–Lebedev (LZBL) formulation of TSH:[43–47]

$$P_{j\to k} = \exp\left(\frac{-\pi}{2\hbar}\sqrt{\frac{Z_{jk}^3}{\ddot{Z}_{jk}}}\right), \tag{7}$$



M. Martyka, L. Zhang, F. Ge, Y.-F. Hou, J. Jankowska, M. Barbatti, P. O. Dral. *Charting electronic-state manifolds across molecules with multi-state learning and gap-driven dynamics via efficient and robust active learning*. https://doi.org/10.26434/chemrxiv-2024-dtc1w

Active learning for surface hopping



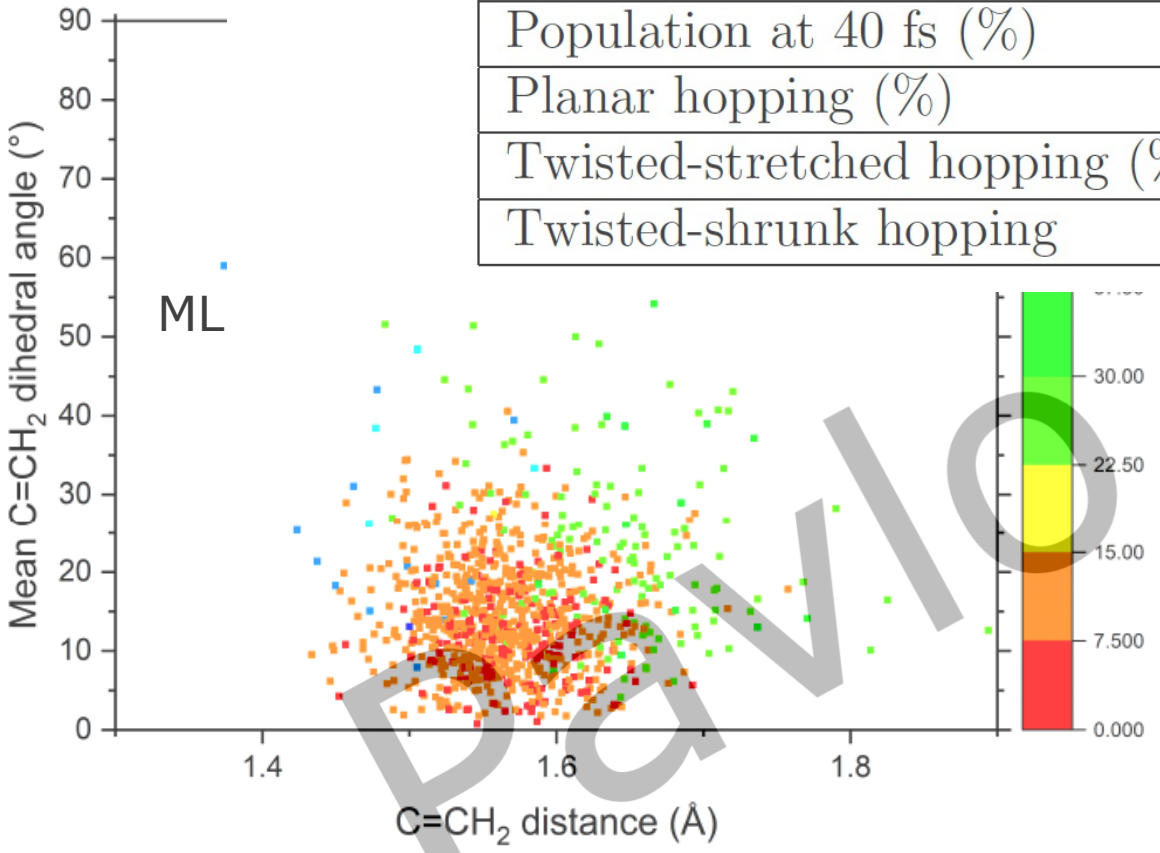
19 iterations of active learning

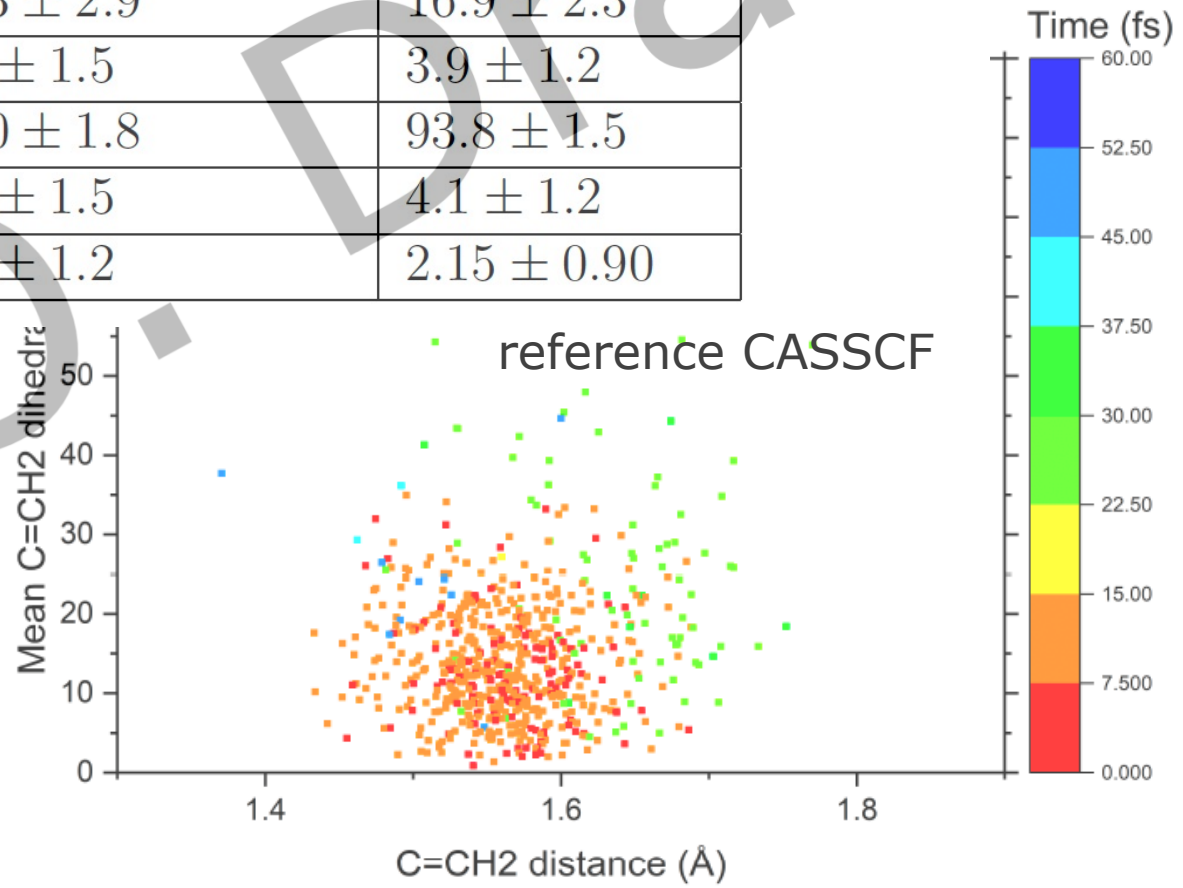
Three days *on*
RTX 4090 GPU and 16 Intel Xeon Gold 6226R CPUs
for the entire procedure with trainings and labeling

M. Martyka, L. Zhang, F. Ge, Y.-F. Hou, J. Jankowska, M. Barbatti, P. O. Dral. *Charting electronic-state manifolds across molecules with multi-state learning and gap-driven dynamics via efficient and robust active learning*. https://doi.org/10.26434/chemrxiv-2024-dtc1w

Table 1: Mean value and error bars (95% confidence interval) of observables describing the deactivation channels and kinetics of fulvene for ML dynamics and reference CASSCF dynamics.

Observable	CASSCF dynamics	ML dynamics
Population at 20 fs (%)	16.3 ± 2.9	16.9 ± 2.3
Population at 40 fs (%)	3.7 ± 1.5	3.9 ± 1.2
Planar hopping (%)	94.0 ± 1.8	93.8 ± 1.5
Twisted-stretched hopping (%)	3.7 ± 1.5	4.1 ± 1.2
Twisted-shrunk hopping	2.3 ± 1.2	2.15 ± 0.90

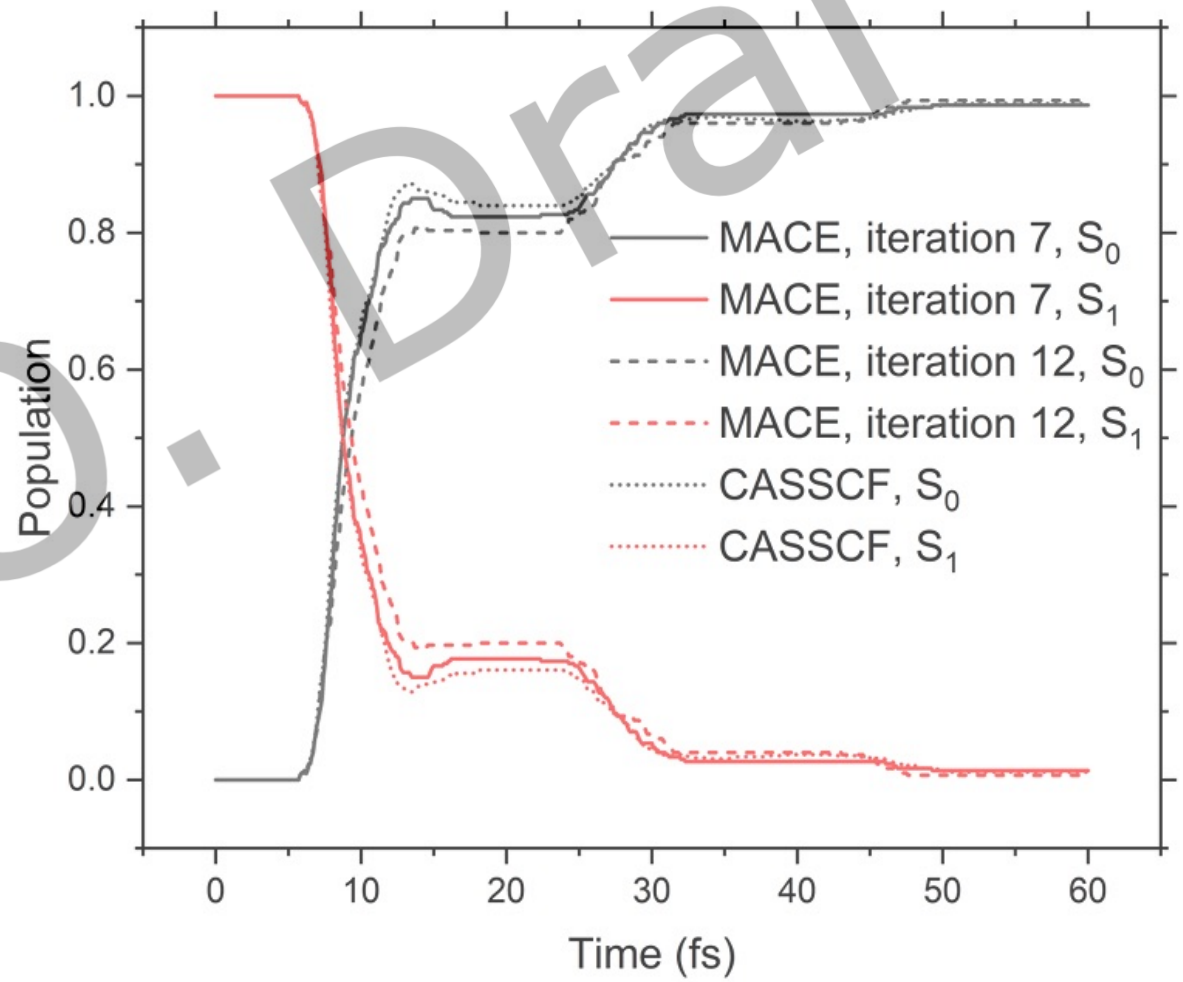




M. Martyka, L. Zhang, F. Ge, Y.-F. Hou, J. Jankowska, M. Barbatti, P. O. Dral. *Charting electronic-state manifolds across molecules with multi-state learning and gap-driven dynamics via efficient and robust active learning*. https://doi.org/10.26434/chemrxiv-2024-dtc1w

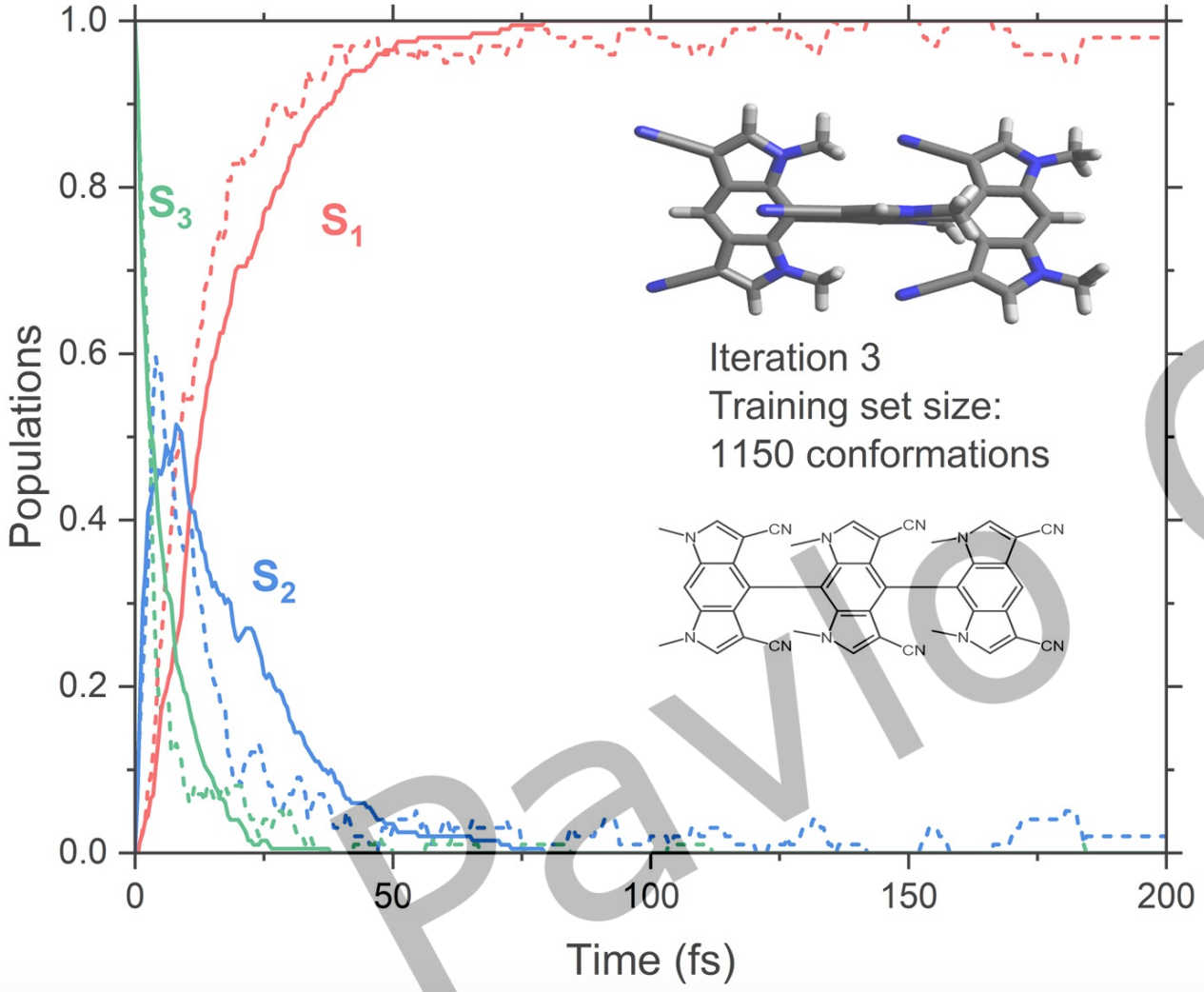
Can equivariant NNs help?

Crashed after 28 days of active learning after 13 iterations and producing 3850 training points



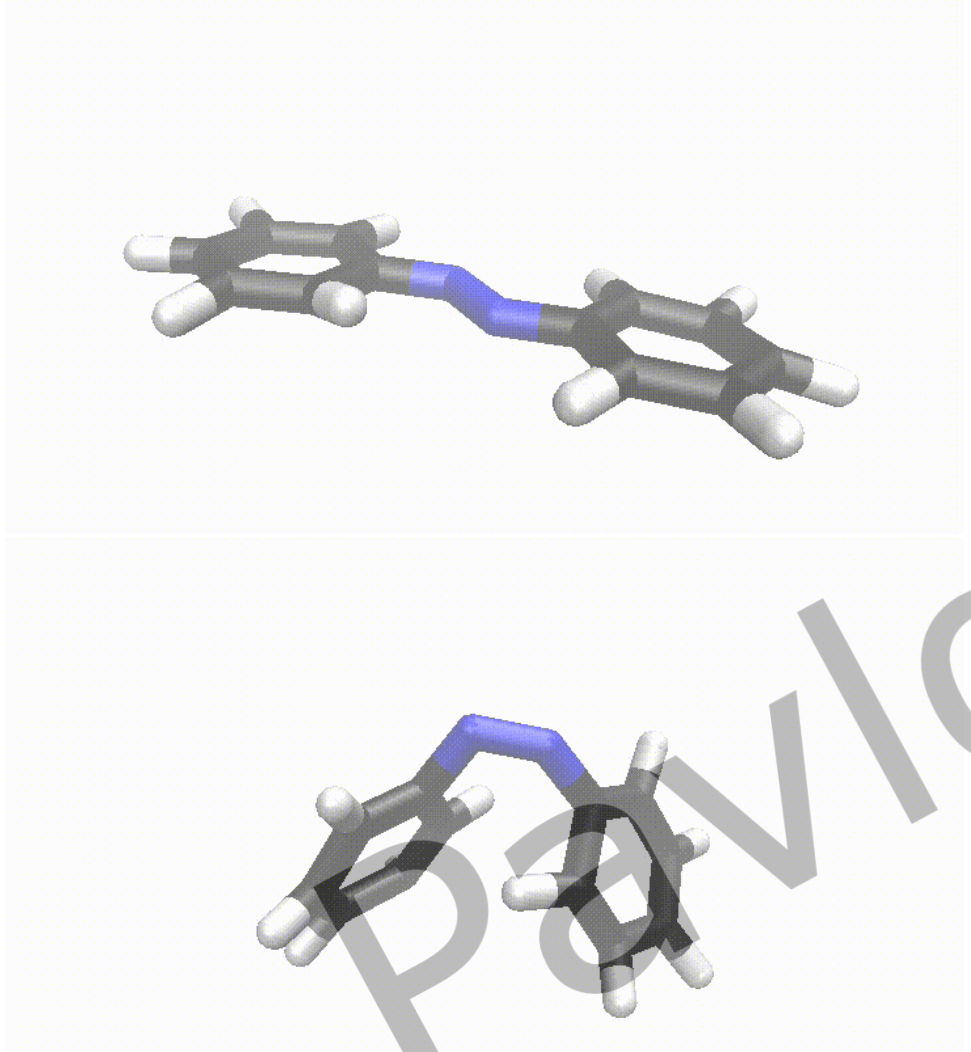
M. Martyka, L. Zhang, F. Ge, Y.-F. Hou, J. Jankowska, M. Barbatti, P. O. Dral. *Charting electronic-state manifolds across molecules with multi-state learning and gap-driven dynamics via efficient and robust active learning*. https://doi.org/10.26434/chemrxiv-2024-dtc1w

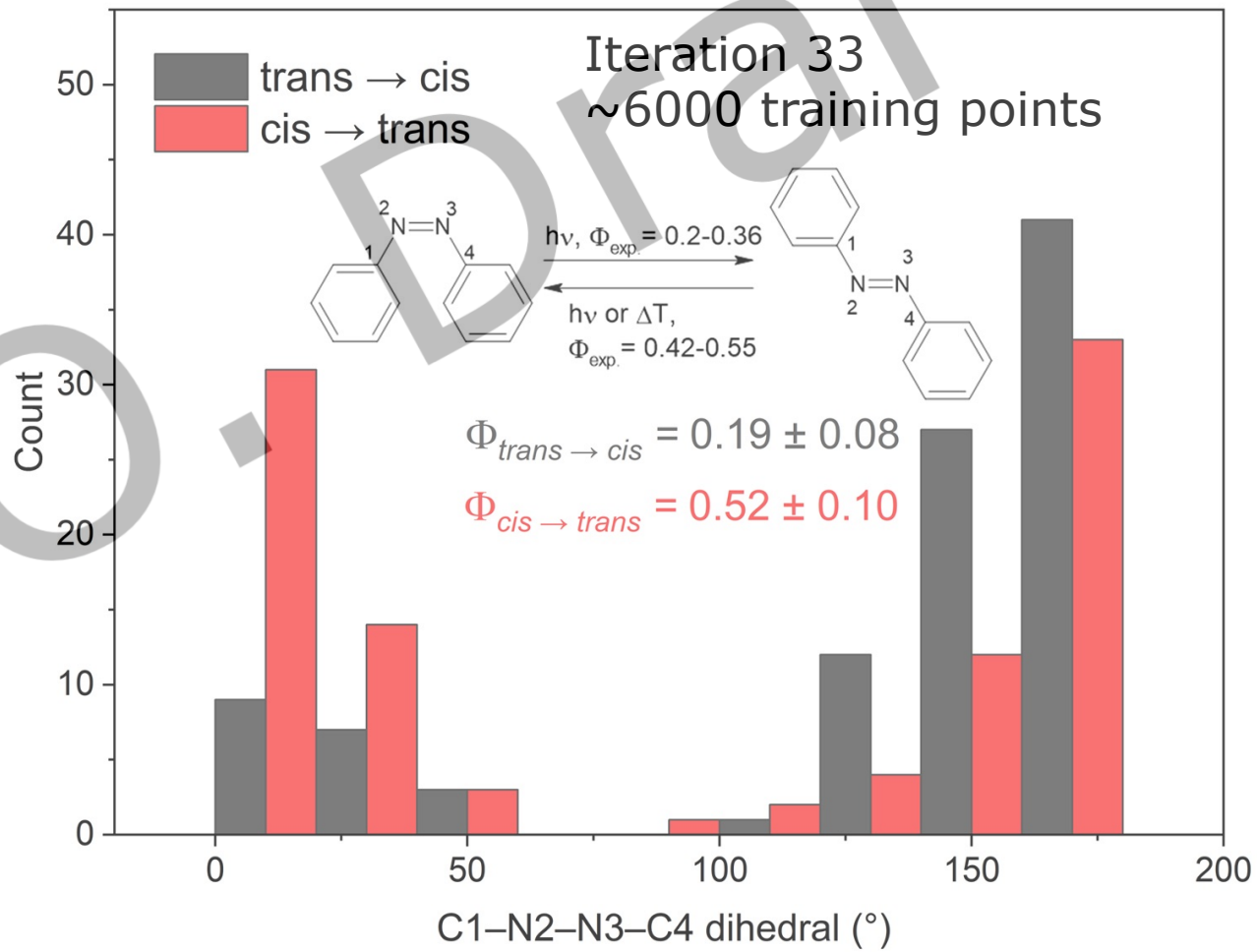
Active learning for surface hopping



M. Martyka, L. Zhang, F. Ge, Y.-F. Hou, J. Jankowska, M. Barbatti, P. O. Dral. *Charting electronic-state manifolds across molecules with multi-state learning and gap-driven dynamics via efficient and robust active learning*. https://doi.org/10.26434/chemrxiv-2024-dtc1w

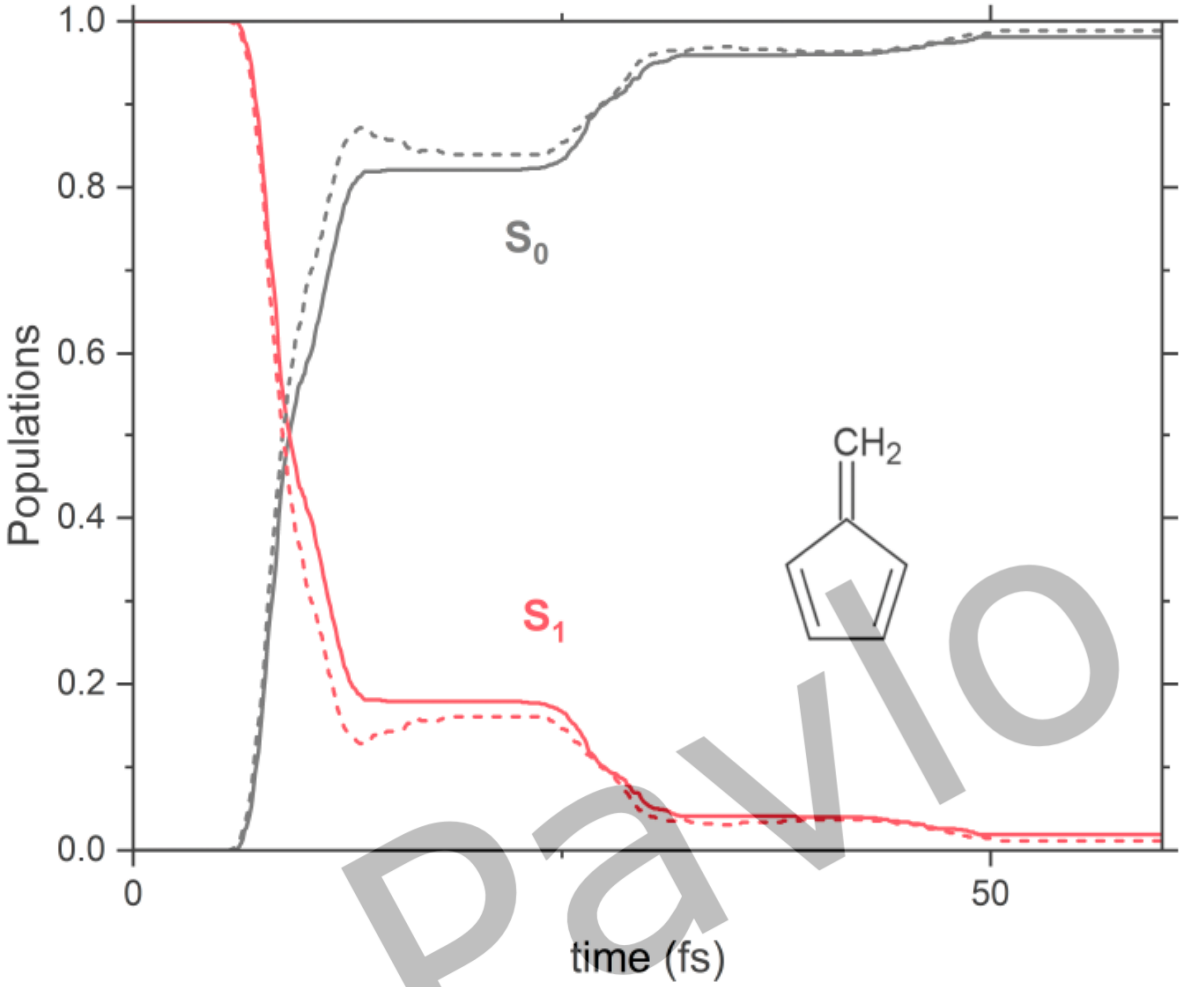
Active learning for photoreactions

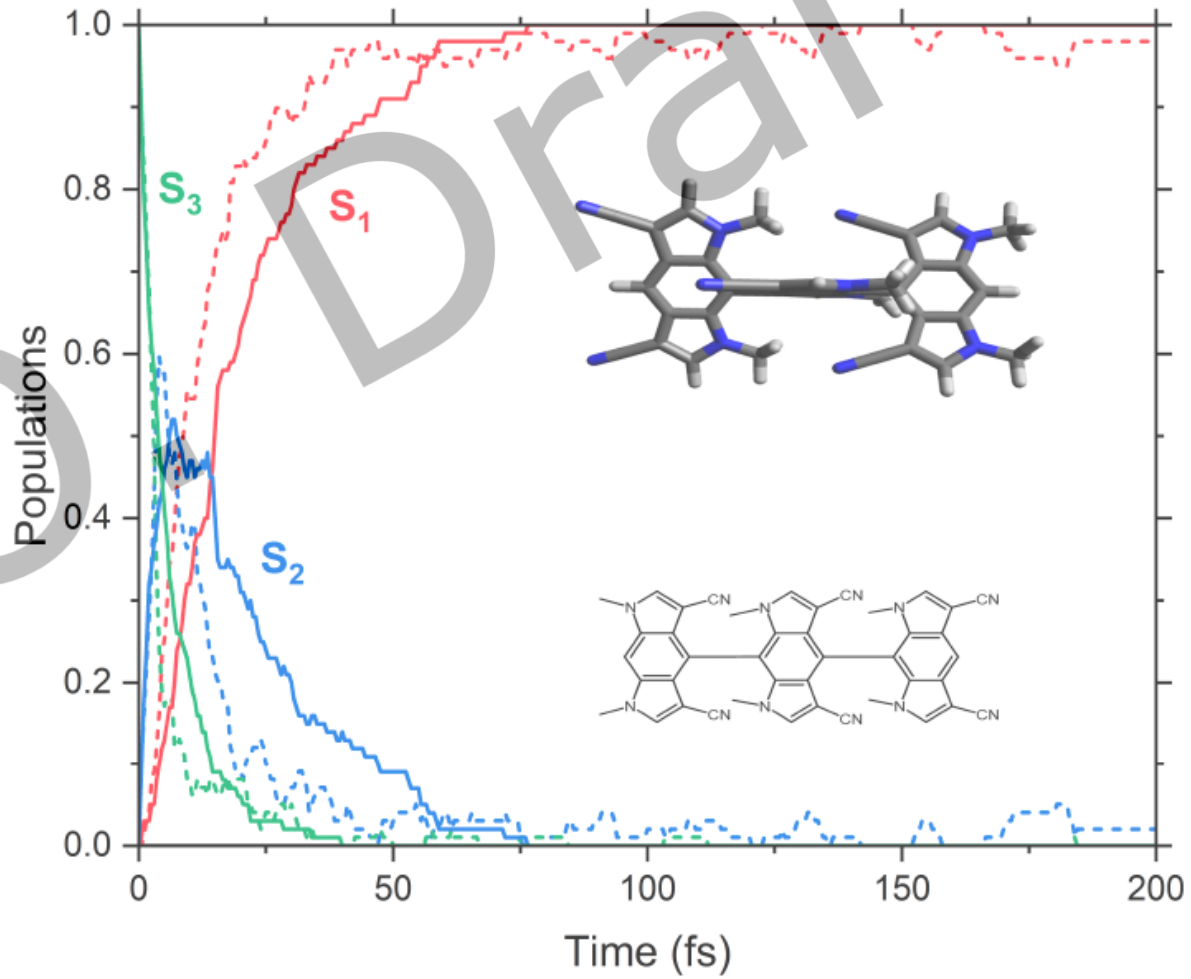




M. Martyka, L. Zhang, F. Ge, Y.-F. Hou, J. Jankowska, M. Barbatti, P. O. Dral. *Charting electronic-state manifolds across molecules with multi-state learning and gap-driven dynamics via efficient and robust active learning*. https://doi.org/10.26434/chemrxiv-2024-dtc1w

One model works across systems!





M. Martyka, L. Zhang, F. Ge, Y.-F. Hou, J. Jankowska, M. Barbatti, P. O. Dral. *Charting electronic-state manifolds across molecules with multi-state learning and gap-driven dynamics via efficient and robust active learning*. https://doi.org/10.26434/chemrxiv-2024-dtc1w

Categories of machine learning potentials

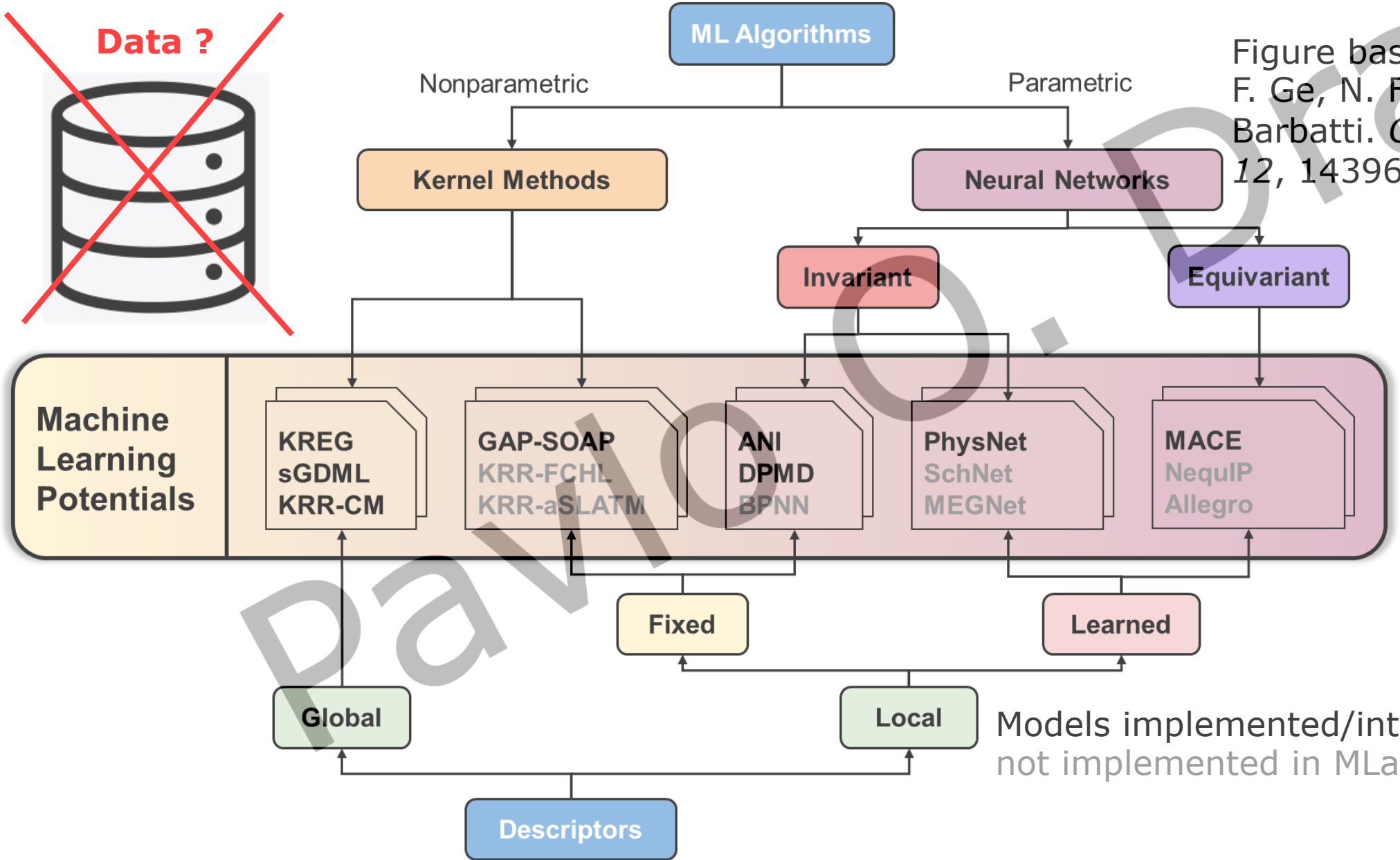
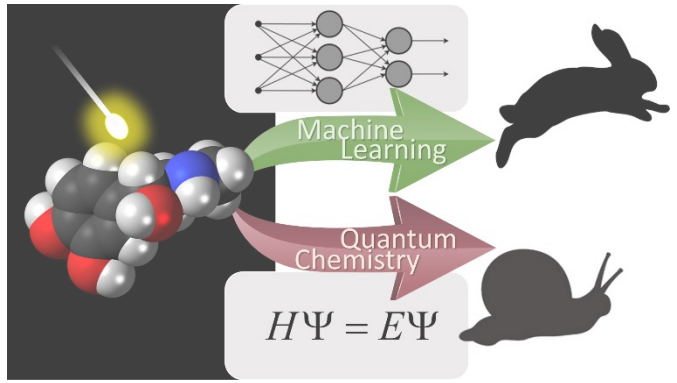


Figure based on M. Pinheiro Jr, F. Ge, N. Ferré, P. O. Dral, M. Barbatti. *Chem. Sci.* **2021**, *12*, 14396–14413

Models implemented/interfaced in MLatom
not implemented in MLatom

Universal models (no training needed)

P. O. Dral, M. Barbatti, *Nat. Rev. Chem.* **2021,** *5*, 388



Timing

Ab initio

ML-ab initio

DFT

ML-DFT

DM21, CF22D, DENS24…

Semi-empirical

ML-SQC

AIQM1

AIQM1: P. Zheng, R. Zubatyuk, W. Wu, O. Isayev, P. O. Dral, *Nat. Commun.* **2021**, *12*, 7022

Machine Learning

ANI-1ccx, AIMnet-2, ANI-1xnr, …

Molecular Mechanics

Accuracy

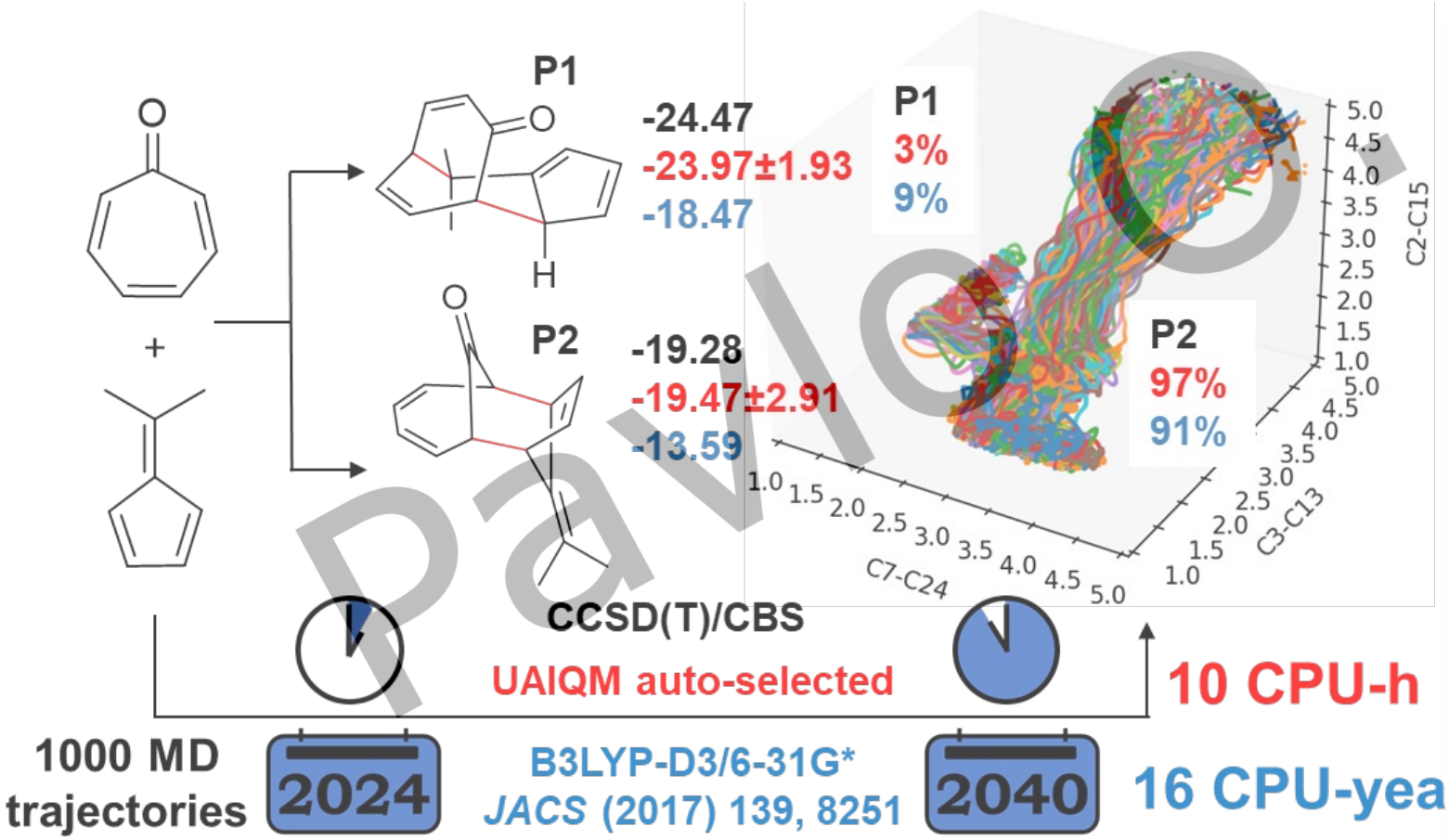
Figure: P. O. Dral, *J. Phys. Chem. Lett.* **2020**, *11*, 2336

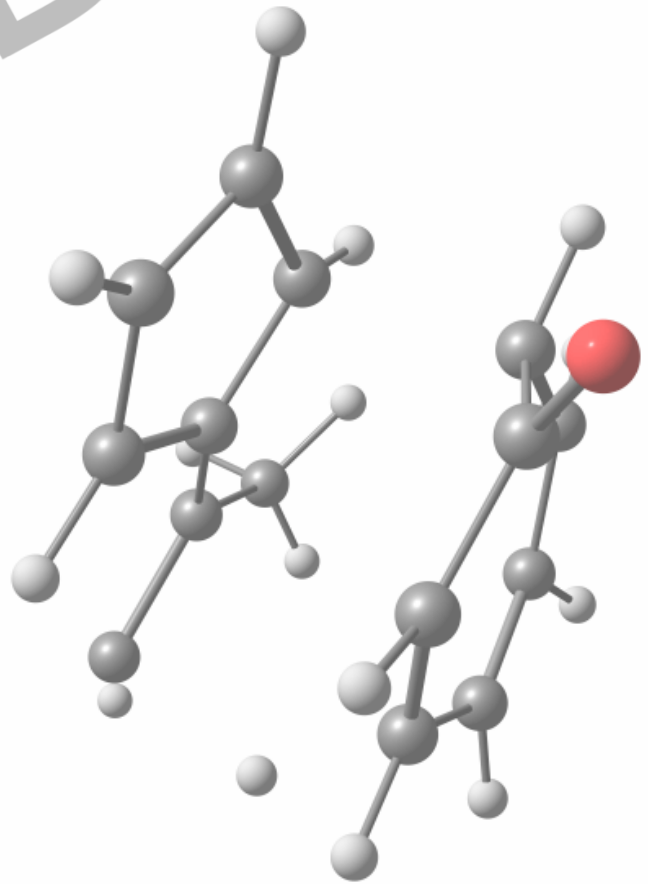
Accessible, versatile, and robust AI-enhanced computational chemistry platform MLatom@XACS:

No reason to do non-ML computational chemistry!

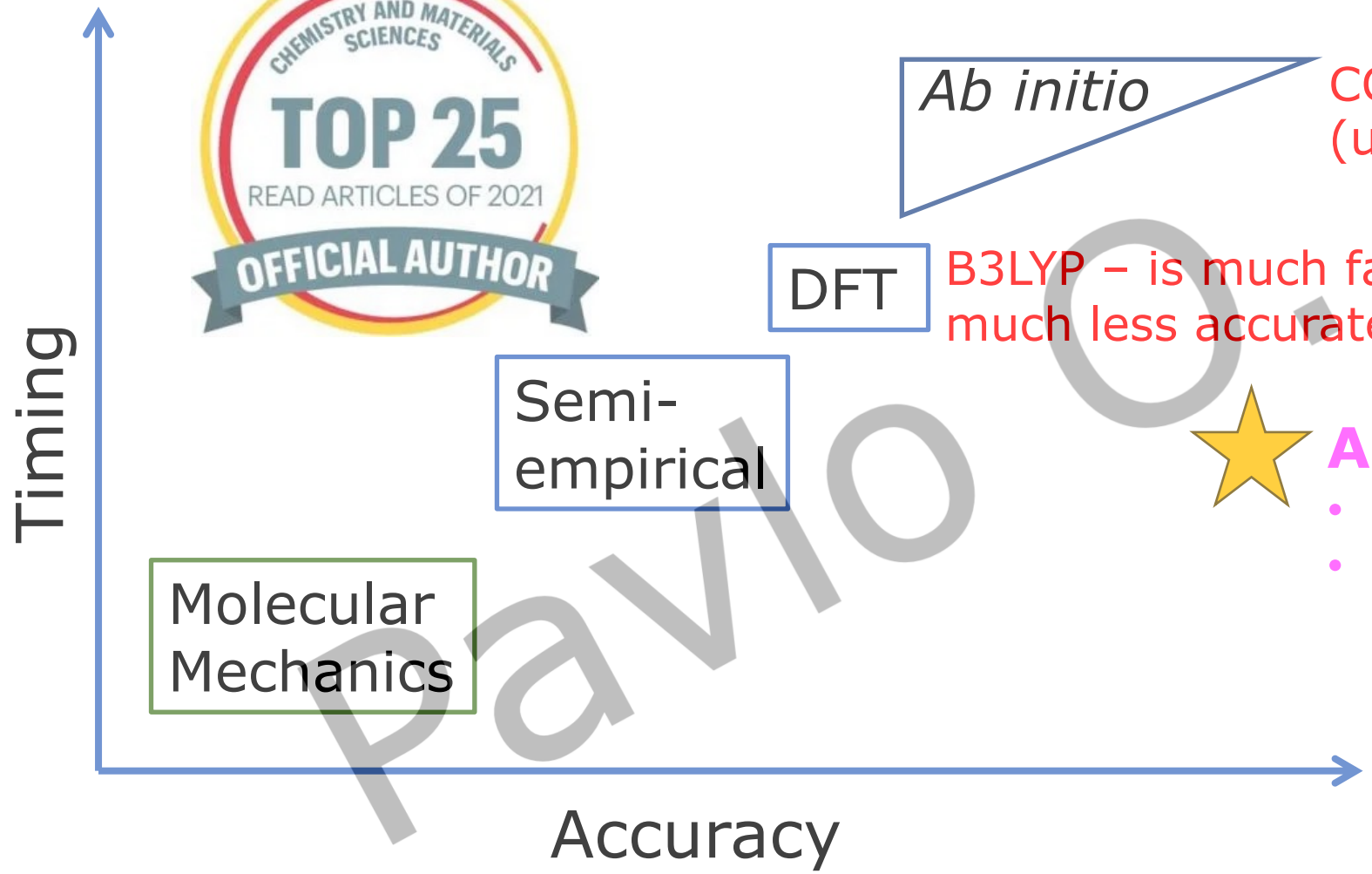
AI methods: **easy to use**, *also* **online**

AI methods are **faster, more accurate, and trustworthy**





Quantum chemistry approximations



CCSD(T) – gold standard, very slow (use it if you can!)

B3LYP – is much faster than the CCSD(T) but is much less accurate

Do not use B3LYP if you can use AIQM1!

AIQM1*

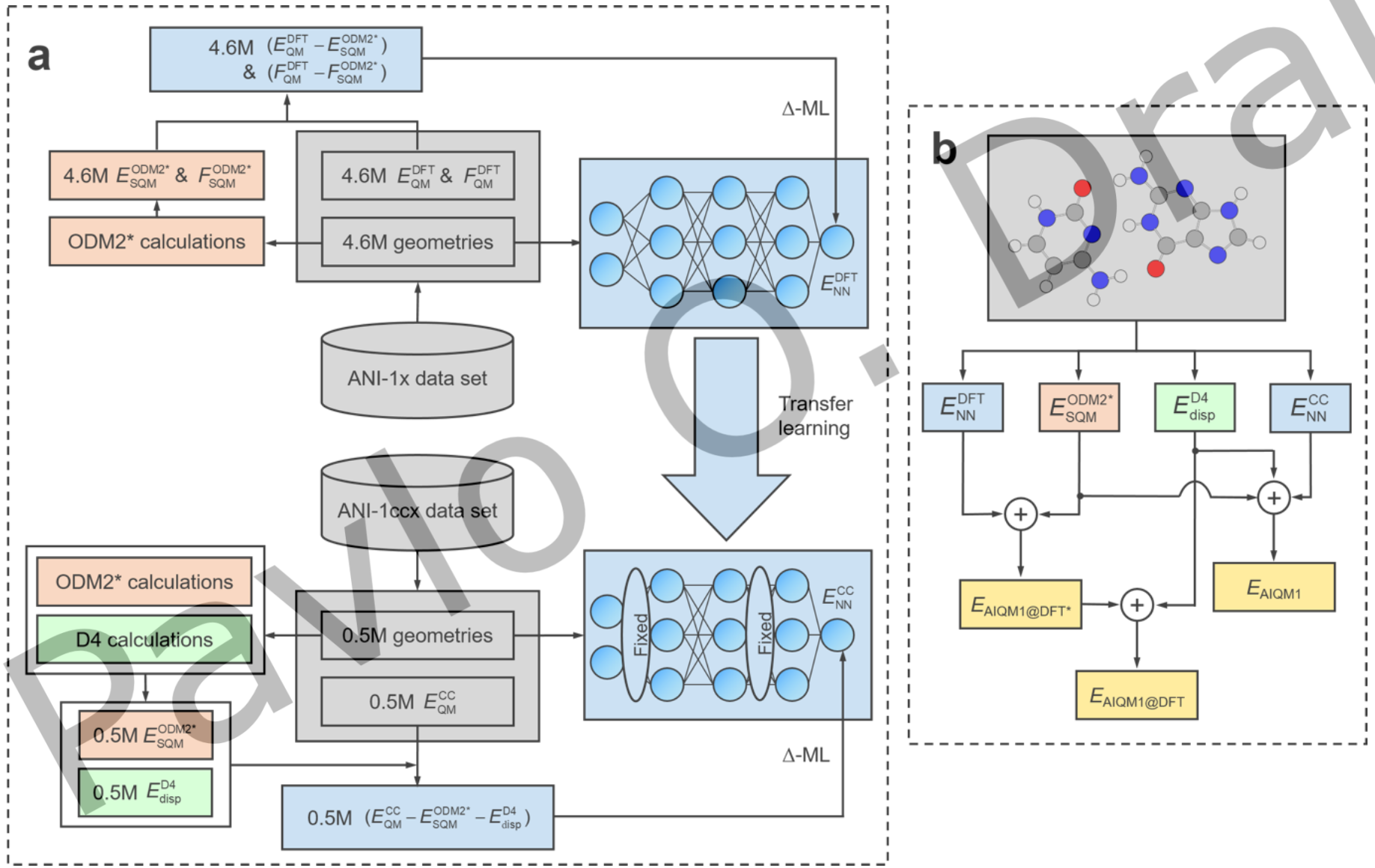
- **Accuracy of CCSD(T)**
- **Orders of magnitude faster than B3LYP**

AIQM1: P. Zheng, R. Zubatyuk, W. Wu, O. Isayev, P. O. Dral, *Nat. Commun.* **2021**, *12*, 7022

*CHNO elements only – extensions on the way

P. O. Dral, *J. Phys. Chem. Lett.* **2020**, *11*, 2336; *Chem. Commun.* **2024**, *60*, 3240

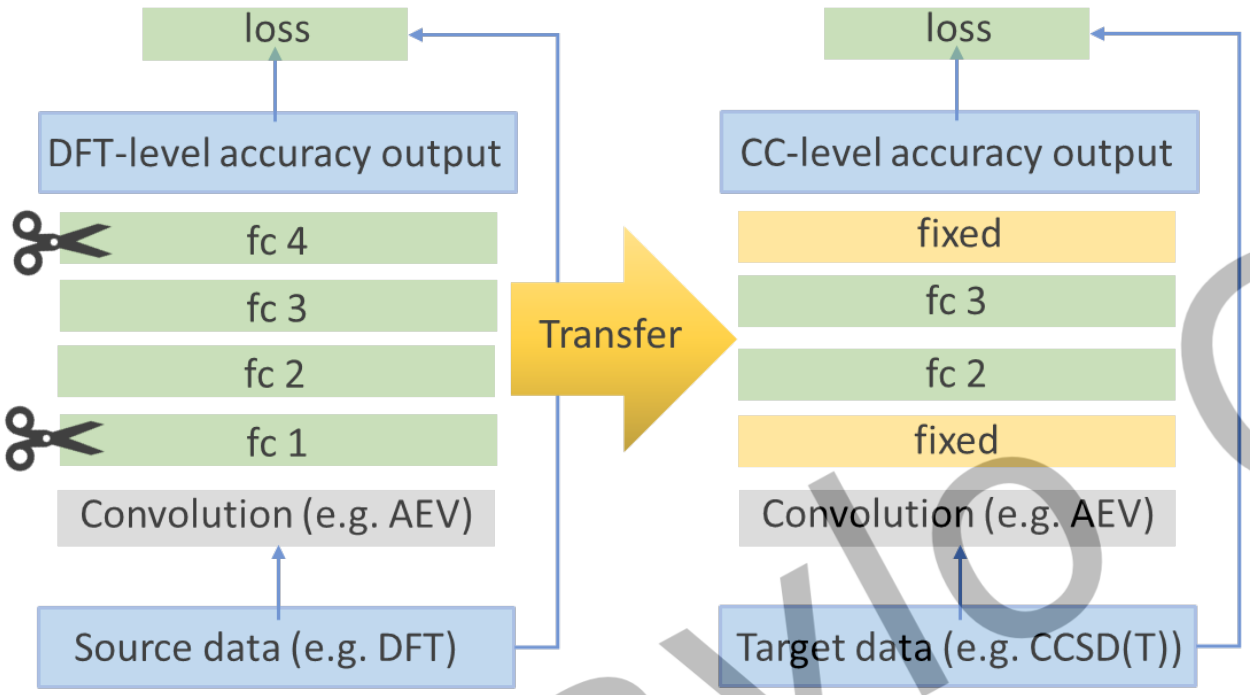
Design

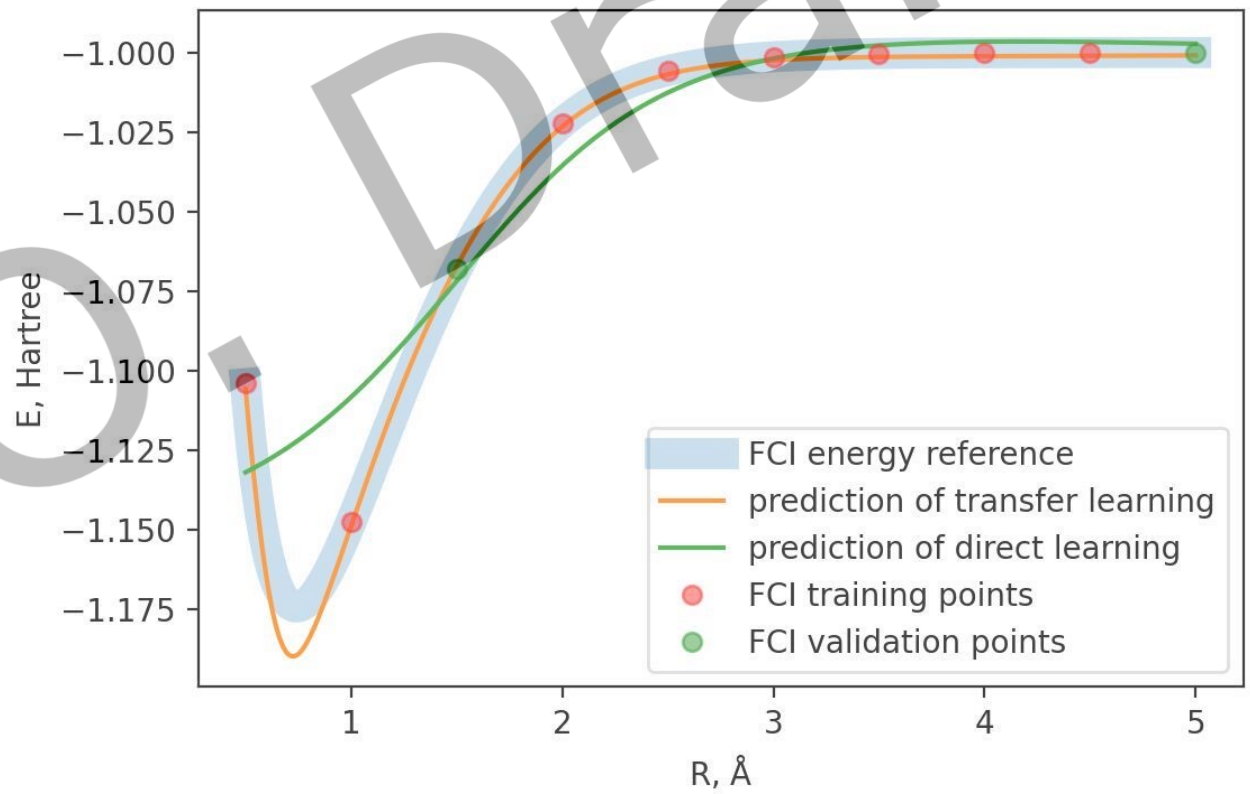


P. Zheng, R. Zubatyuk, W. Wu, O. Isayev, P. O. Dral, *Nat. Commun.* **2021**, *12*, 7022

Dealing with multiple levels

Multi-fidelity data: Transfer learning

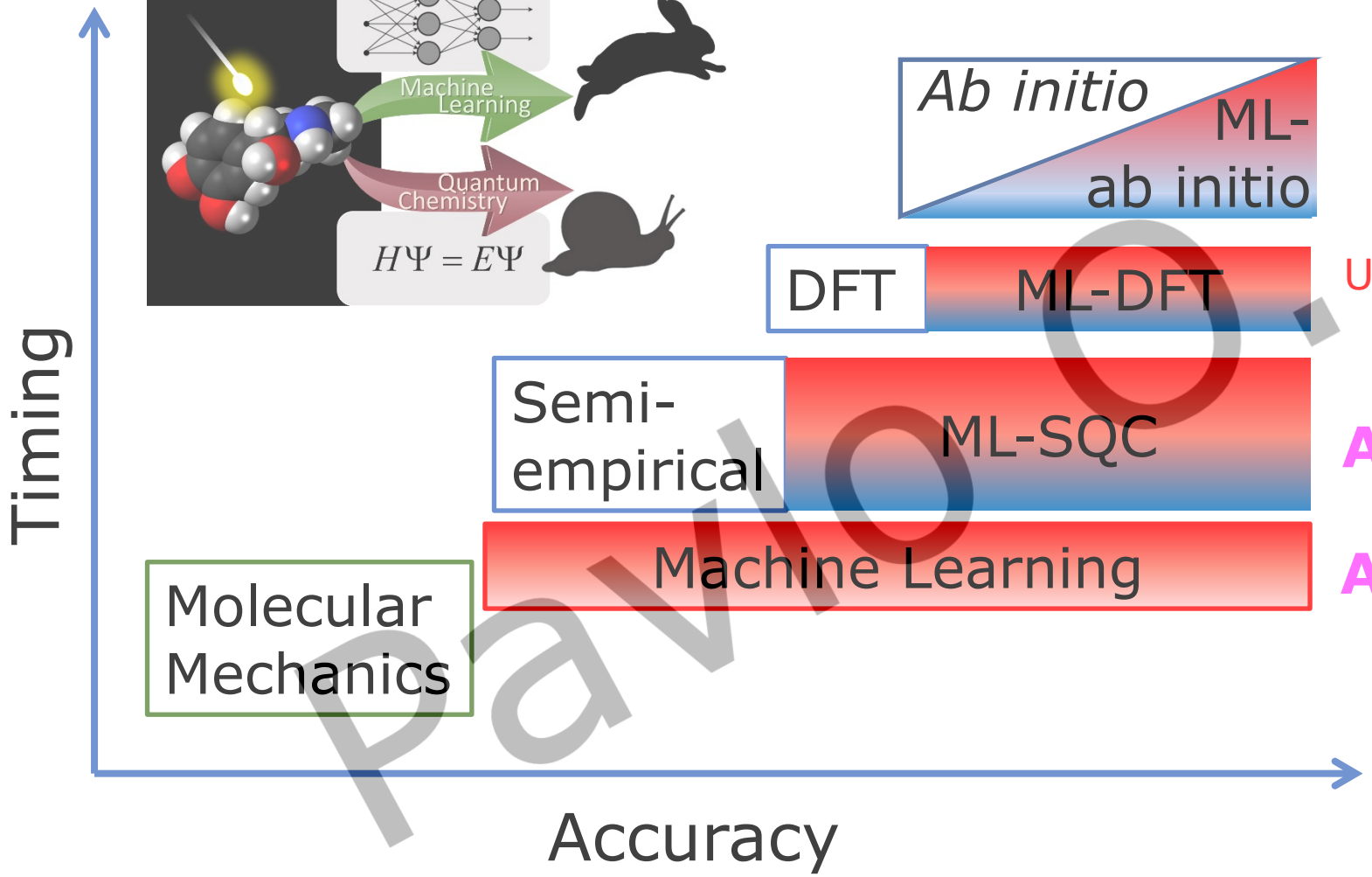




Figures: Pavlo O. Dral, Tetiana Zubatiuk, Bao-Xin Xue, Learning from multiple quantum chemical methods: Δ-learning, transfer learning, co-kriging, and beyond. In *Quantum Chemistry in the Age of Machine Learning*, Pavlo O. Dral, Ed. Elsevier: **2023**. Paperback ISBN: 9780323900492

Machine learning in quantum chemistry

P. O. Dral, M. Barbatti, *Nat. Rev. Chem.* **2021,** *5*, 388



Universal models (no training required):

AIQM1 AIQM1: P. Zheng, R. Zubatyuk, W. Wu, O. Isayev, P. O. Dral, *Nat. Commun.* **2021**, *12*, 7022

ANI-1ccx ANI-1ccx: J. S. Smith, *et al*. *Nat. Commun.* **2019,** *10*, 2903

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

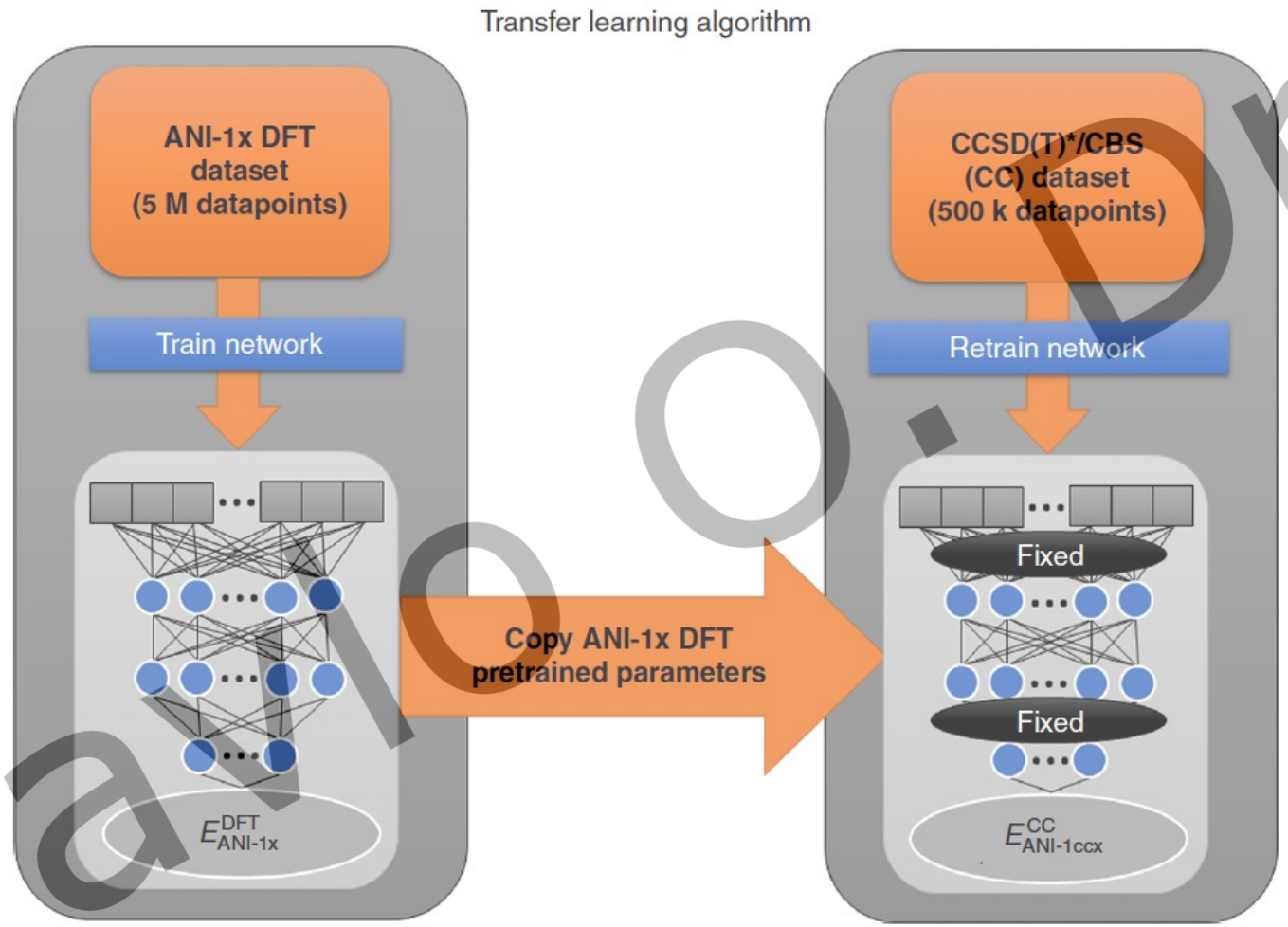
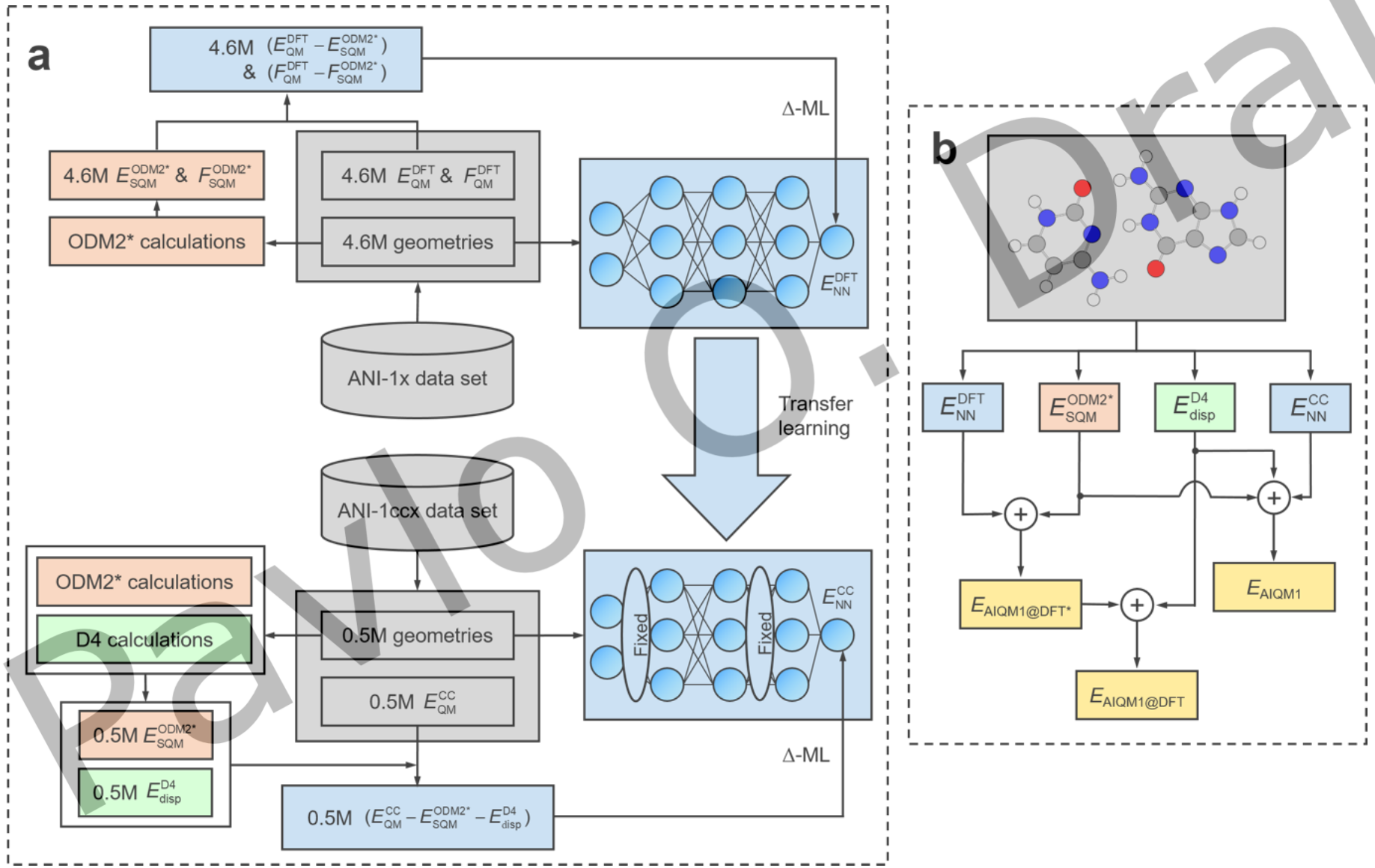


Fig. 4 Diagram of the transfer learning technique evaluated in this work. Transfer learning starts from a pretrained ANI-1x DFT model, then retrains to higher accuracy CCSD(T)*/CBS data with some parameters fixed during training

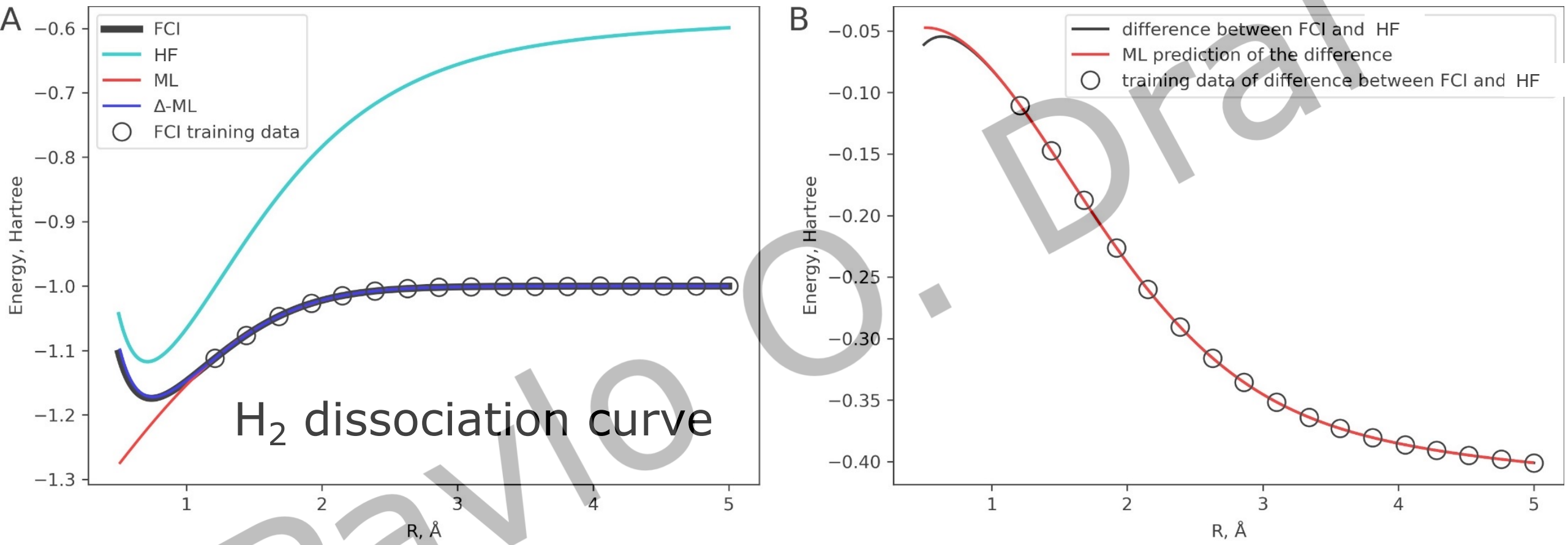
ANI-1ccx: J. S. Smith, *et al*. *Nat. Commun.* **2019,** *10*, 2903

Design



P. Zheng, R. Zubatyuk, W. Wu, O. Isayev, P. O. Dral, *Nat. Commun.* **2021**, *12*, 7022

Multi-fidelity data: Δ-learning



Δ-learning: R. Ramakrishnan, P. O. Dral, M. Rupp, O. A. von Lilienfeld, *J. Chem. Theory Comput.* **2015**, *11*, 2087

Figure: Pavlo O. Dral, Tetiana Zubatiuk, Bao-Xin Xue, Learning from multiple quantum chemical methods: Δ-learning, transfer learning, co-kriging, and beyond. In *Quantum Chemistry in the Age of Machine Learning*, Pavlo O. Dral, Ed. Elsevier: **2023**. Paperback ISBN: 9780323900492

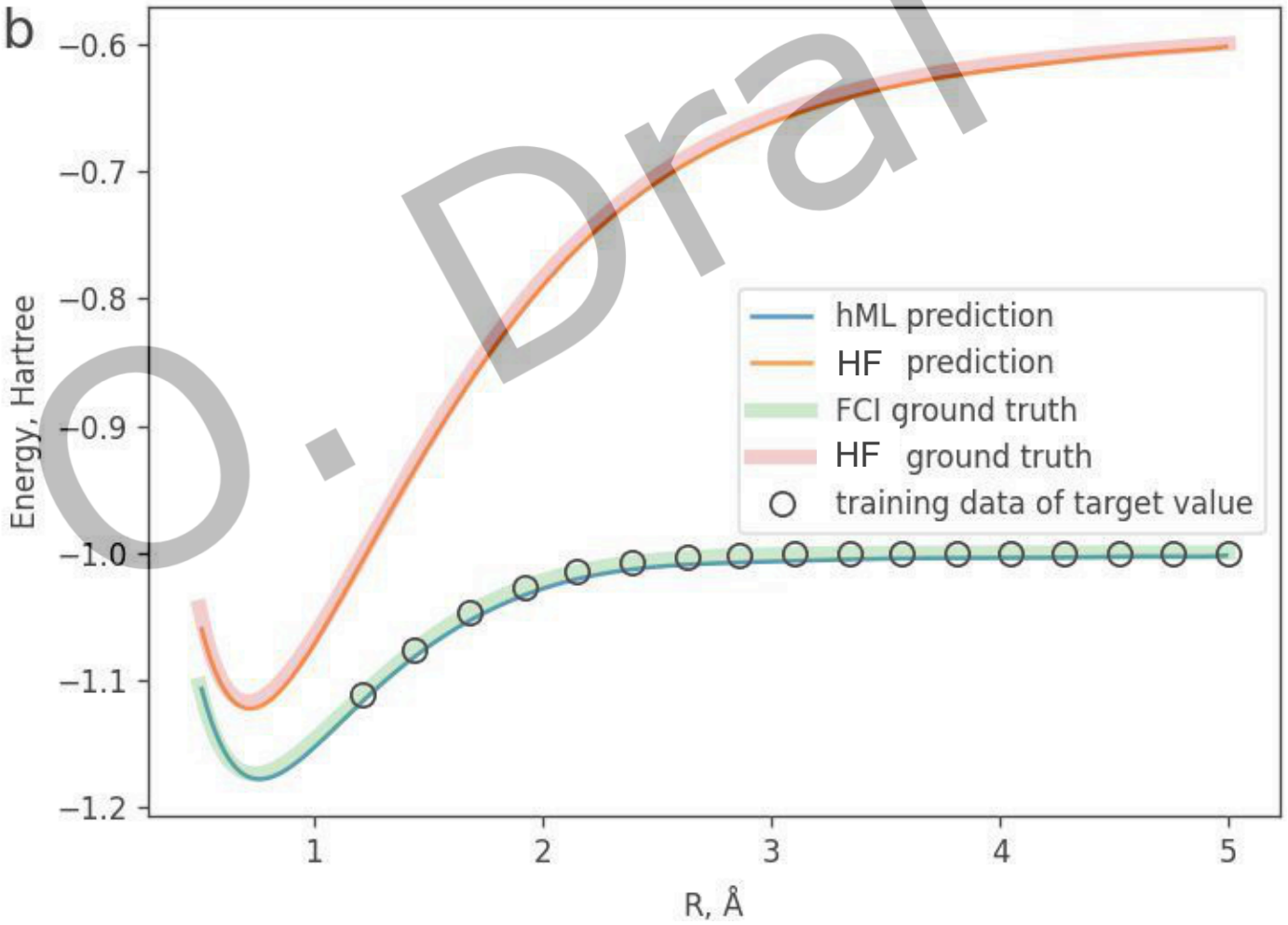
Hierarchical ML (hML): generalized Δ-learning

Automatic procedure to find optimal training points for each delta-model

$$\hat{y}_{\Sigma,i} = \sum_M \hat{y}_{M,i}(N_{\mathrm{tr},M})$$

hML for H_2

$$\mathrm{hML} = \Delta_0^{\mathrm{HF}} + \Delta_{\mathrm{HF}}^{\mathrm{FCI}}$$



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

Figure: Pavlo O. Dral, Tetiana Zubatiuk, Bao-Xin Xue, Learning from multiple quantum chemical methods: Δ-learning, transfer learning, co-kriging, and beyond. In *Quantum Chemistry in the Age of Machine Learning*, Pavlo O. Dral, Ed. Elsevier: **2023**. Paperback ISBN: 9780323900492

Hierarchical ML (hML) Combining Δ-ML with Sampling

Automatic *procedure to find optimal training points for each Δ-model*

$$\hat{y}_{\Sigma,i} = \sum_M \hat{y}_{M,i}(N_{\mathrm{tr},M})$$

$$\frac{e_\Sigma}{e_{\Sigma,\mathrm{th}}} + \frac{t_\Sigma / t_{\Sigma,\max}}{1 - s_{\mathrm{th}}} + \frac{p}{e_{\Sigma,\mathrm{th}}}$$

$$\frac{\hat{y}_{M,i}(N_{\mathrm{tr},M})}{\hat{y}_{1,i}(N_{\mathrm{tr},M})} \approx r_{M/1,i} = \frac{y_{M,i}}{y_{1,i}}$$

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

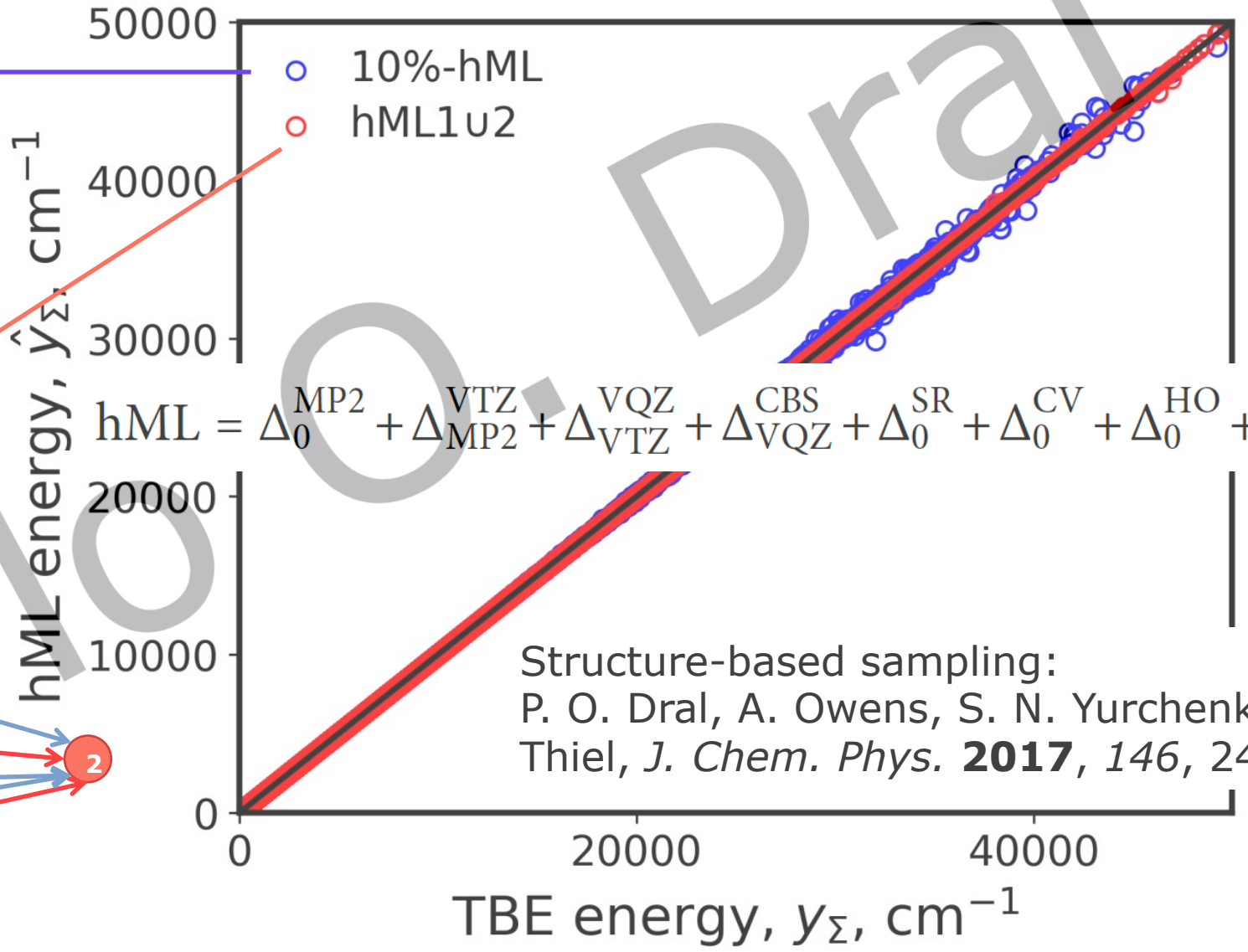
P. O. Dral, A. Owens, A. Dral, G. Csányi, *J. Chem. Phys.* **2020**, *152*, 204110

Hierarchical ML for CH_3Cl PES

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

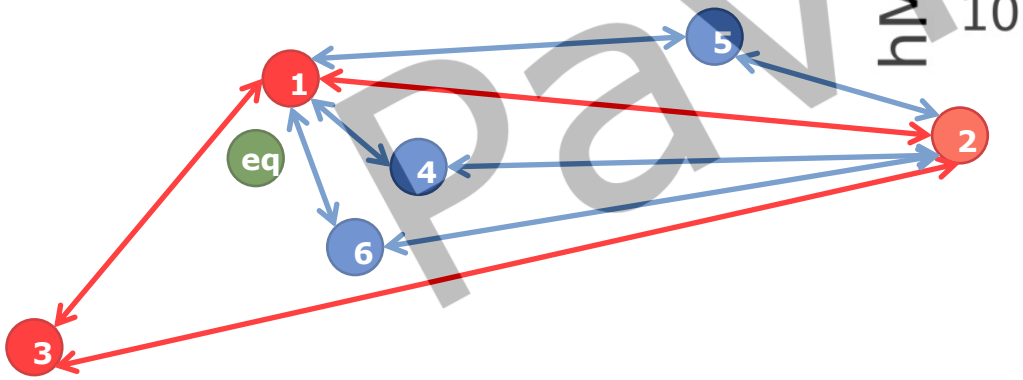
Weighted RMSE:
3.49 cm⁻¹ = 0.01 kcal/mol

Δ-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



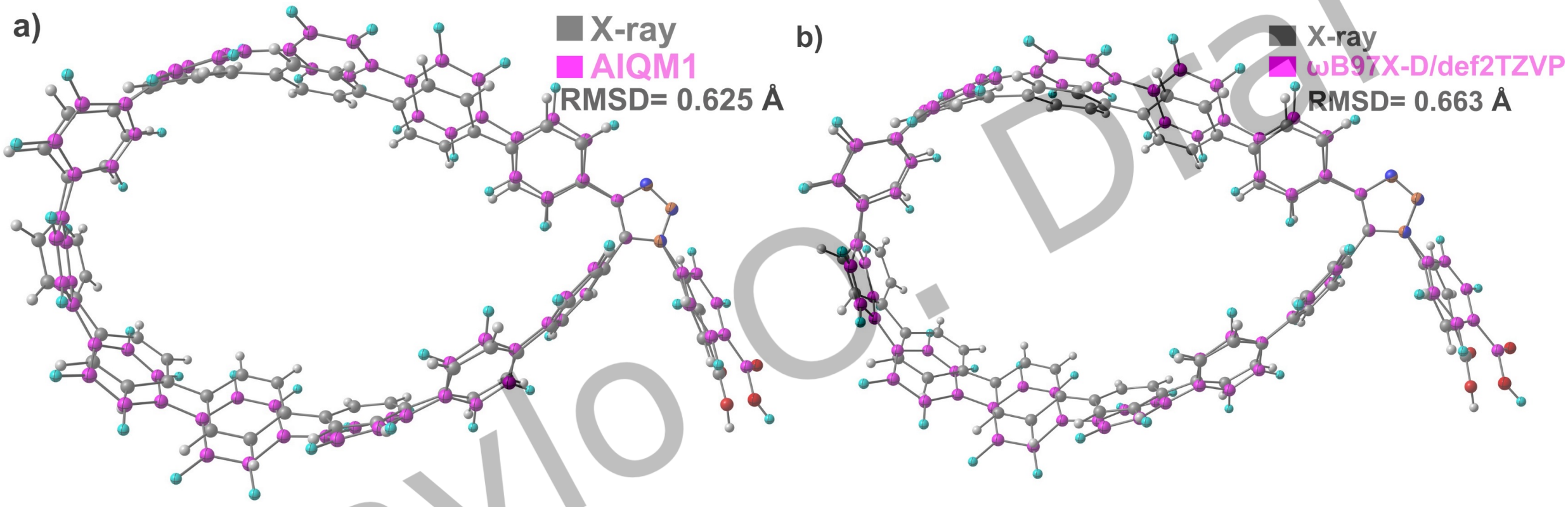
P. O. Dral, A. Owens, A. Dral, G. Csányi, *J. Chem. Phys.* **2020**, *152*, 204110

data set	ODM2	B3LYP/ 6-31G*	ωB97X/ 6-31G*	ωB97X-D/ 6-31G*	ωB97X/ def2-TZVPP	ωB97X-D4/ def2-TZVPP	ANI-1ccx	AIQM1 @DFT*	AIQM1 @DFT	AIQM1	CCSD(T)* /CBS
	energies, kcal/mol										
CHNO	2.64	6.71	4.10	3.84	3.21	2.76	—	2.49	2.12	0.87	—
G3/99	3.04	8.53	3.46	3.22	4.18	3.20	—	2.83	2.06	0.88	—
ISOMERS44 (ΔH_f)	1.16	8.08	3.57	3.53	4.52	3.78	—	3.00	2.27	0.42	—
ISOMERS44 (ΔH_r)	0.70	2.29	1.45	1.31	1.19	1.10	1.68	0.95	0.89	0.50	—
IsoL6/11	1.48	5.26	3.83	3.36	1.75	1.64	1.46	1.65	1.55	0.62	0.47
HC7/11	5.37	6.44	16.90	13.98	6.83	7.10	2.53	8.89	9.16	1.43	1.57
Torsion	0.74	0.55	0.30	0.29	0.20	0.19	0.23	0.23	0.23	0.19	0.05
	bond lengths, Å										
CHNO	0.015	0.006	0.008	0.007	0.010	0.010	0.011	0.010	0.010	0.007	—
MGHBL9	0.023	0.007	0.006	0.005	0.002	0.002	0.047	0.011	0.011	0.004	—
MGNHBL11	0.026	0.006	0.003	0.002	0.008	0.008	0.004	0.008	0.008	0.002	—
	bond angles, °										
CHNO	2.04	0.70	0.68	0.64	0.68	0.68	1.00	0.77	0.77	0.70	—
	dihedral angles, °										
CHNO	4.07	5.20	4.68	6.10	7.12	7.11	5.86	2.14	2.14	2.31	—

Ground-state properties of neutral, closed-shell compounds
(heats of formation, reaction enthalpies, and ZPVE-exclusive reaction energies)

P. Zheng, R. Zubatyuk, W. Wu, O. Isayev, P. O. Dral, *Nat. Commun.* **2021**, *12*, 7022

AIQM1 investigation of cycloparaphenylenes

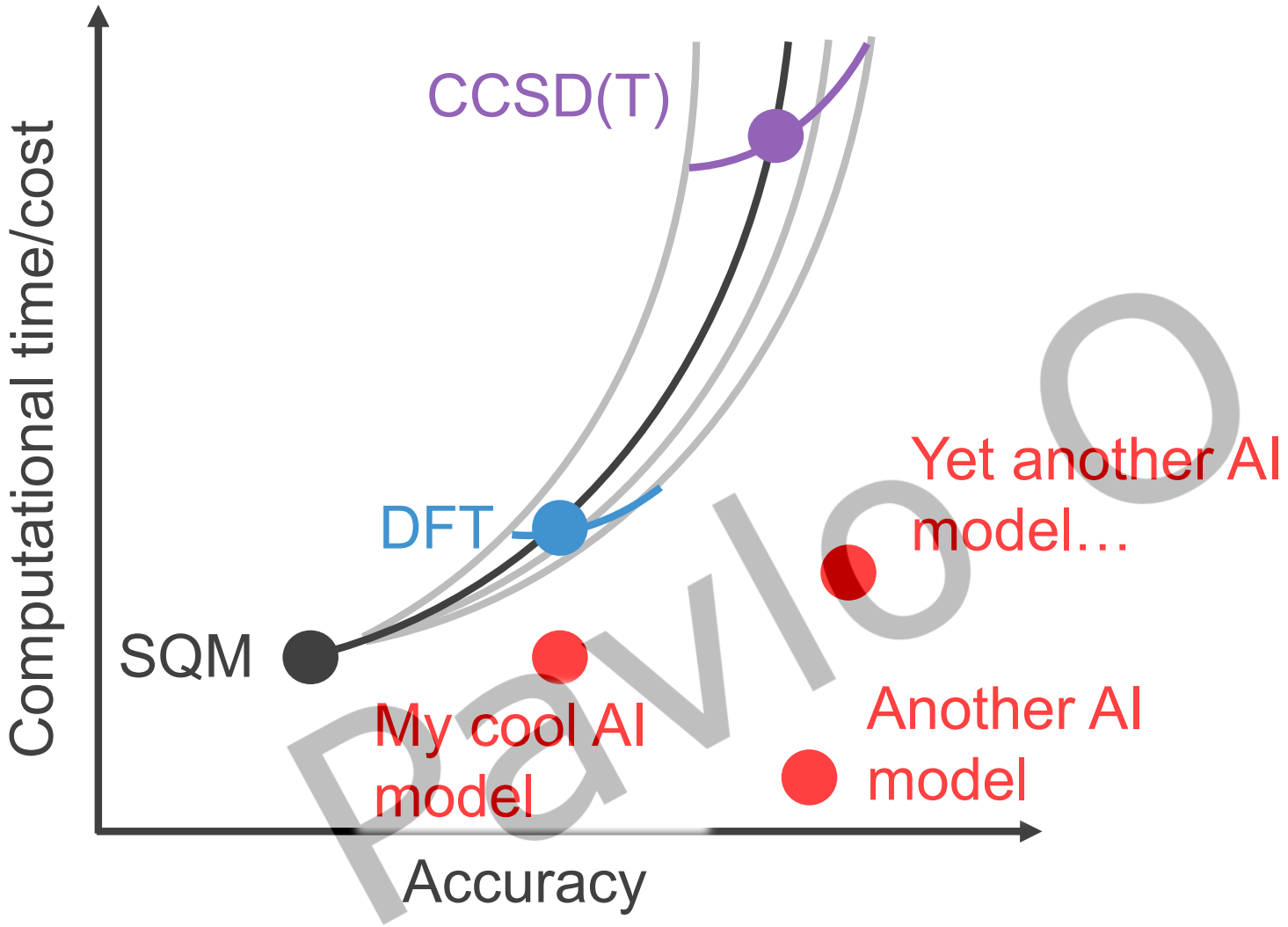


2 CPU-minutes

1200 CPU-minutes

T. A. Schaub, A. Zieleniewska, R. Kaur, M. Minameyer, W. Yang, C. M. Schüßlbauer, L. Zhang, M. Freiberger, L. N. Zakharov, T. Drewello, P. O. Dral, D. Guldi, R. Jasti. Tunable Macrocyclic Polyparaphenylene Nanolassos via Copper-Free Click Chemistry. *Chem. Eur. J.* **2023**, *29*, e202300668

UAIQM: Universal and Updatable AI-QM models

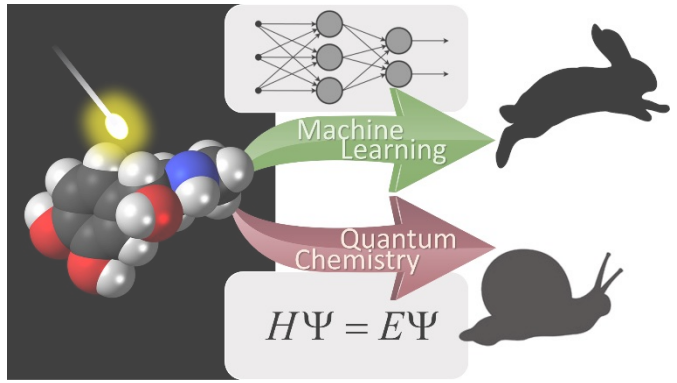


Try to create more better models…

Y. Chen, Y.-F. Hou, O. Isayev, P. O. Dral. *Universal and Updatable Artificial Intelligence-Enhanced Quantum Chemical Foundational Models*. **2024**, *submitted*. https://doi.org/10.26434/chemrxiv-2024-604wb.

Can we do better?

P. O. Dral, M. Barbatti, *Nat. Rev. Chem.* **2021,** *5*, 388



Lots of models!
How to choose?

Ab initio
ML-ab initio

DFT
ML-DFT
DM21, CF22D, DENS24…

Semi-empirical
ML-SQC
AIQM1

Machine Learning
ANI-1ccx, AIMnet-2, ANI-1xnr, …

Molecular Mechanics

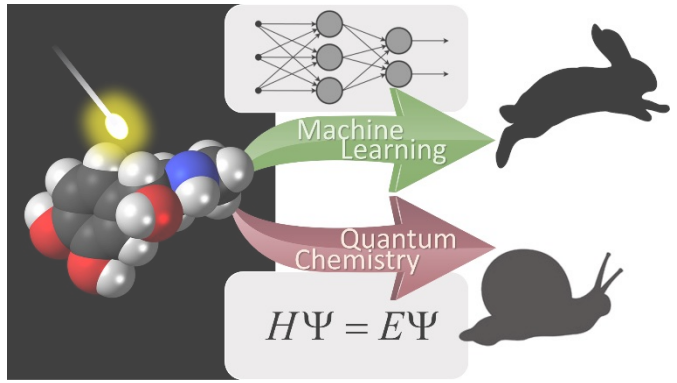
Timing

Accuracy

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

Can we do better?

P. O. Dral, M. Barbatti, *Nat. Rev. Chem.* **2021,** *5*, 388



Every model (DFT or AI …) has some problems!

Ab initio
ML-ab initio

DFT
ML-DFT
DM21, CF22D, DENS24…

Semi-empirical
ML-SQC
AIQM1

Machine Learning
ANI-1ccx, AIMnet-2, ANI-1xnr, …

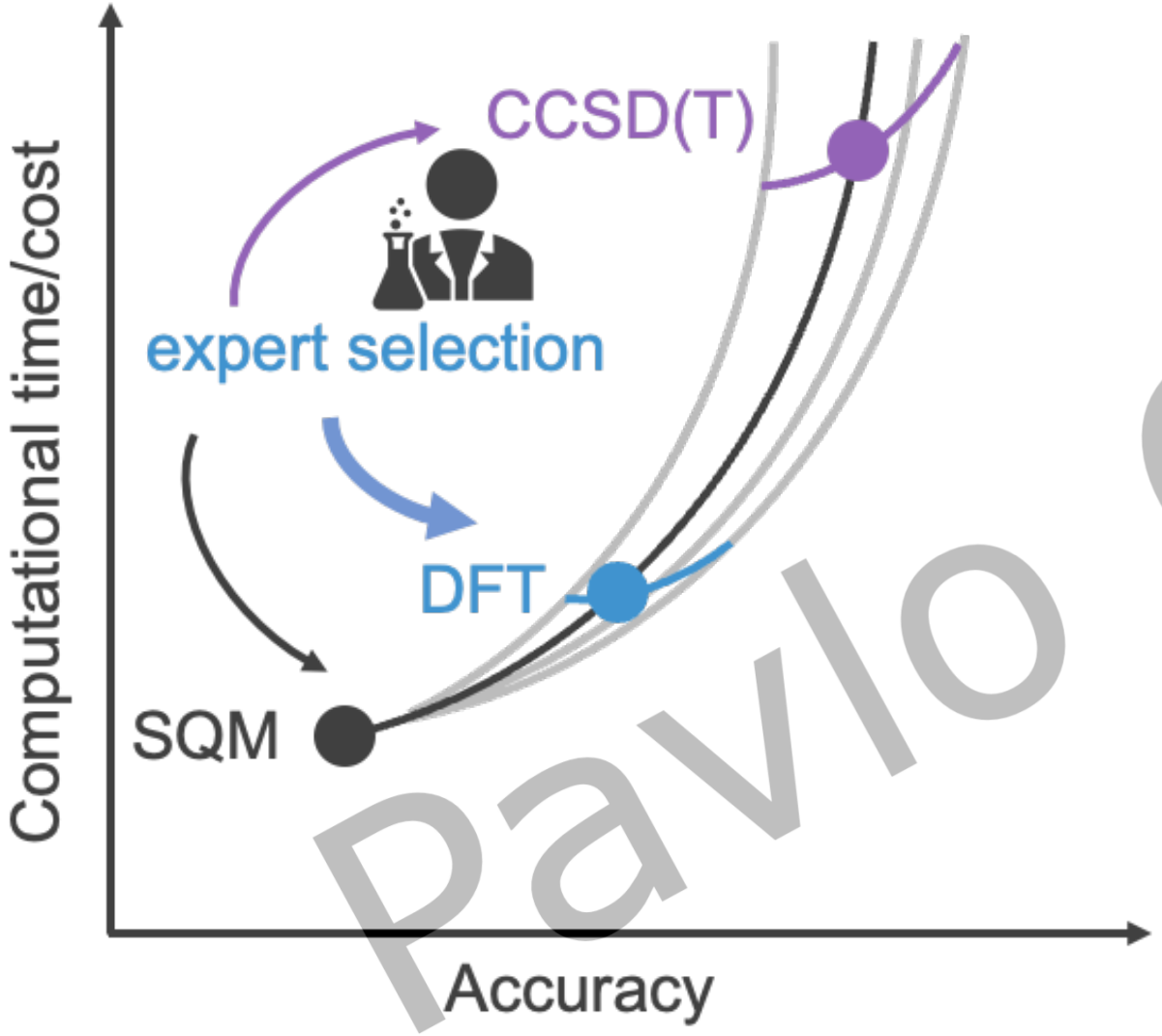
Molecular Mechanics

Timing

Accuracy

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

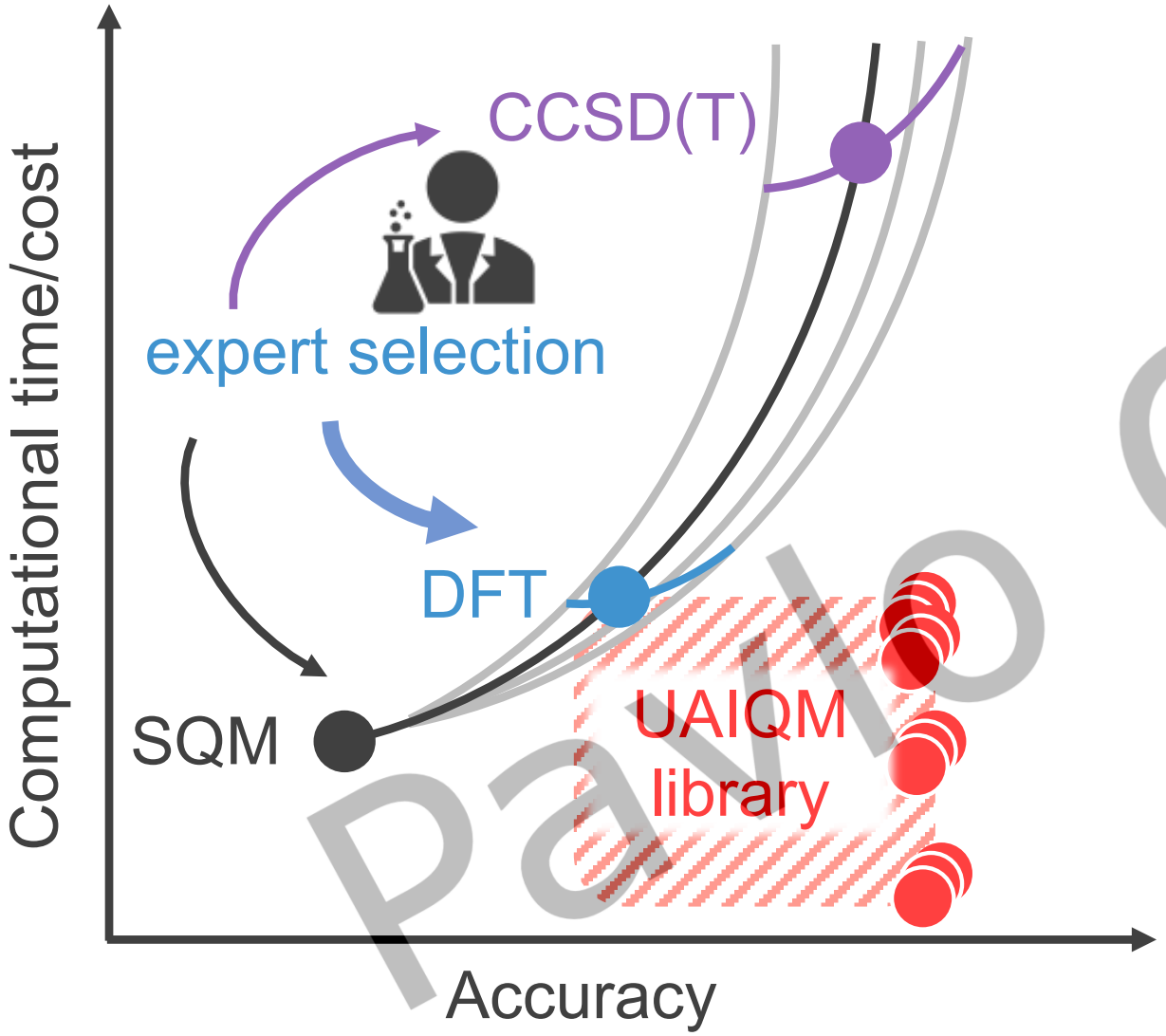
UAIQM: Universal and Updatable AI-QM models



We can use human expertise to select models

Y. Chen, Y.-F. Hou, O. Isayev, P. O. Dral. *Universal and Updatable Artificial Intelligence-Enhanced Quantum Chemical Foundational Models*. **2024**, *submitted*. https://doi.org/10.26434/chemrxiv-2024-604wb.

UAIQM: Universal and Updatable AI-QM models



We put together a dozen of state-of-the-art models in a library and just give new models version numbers!

Y. Chen, Y.-F. Hou, O. Isayev, P. O. Dral. *Universal and Updatable Artificial Intelligence-Enhanced Quantum Chemical Foundational Models*. **2024**, *submitted*. https://doi.org/10.26434/chemrxiv-2024-604wb.

UAIQM: Universal and Updatable AI-QM models

nature communications

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature > nature communications > articles > article

Article | Open access | Published: 02 December 2021

Artificial intelligence-enhanced quantum chemical method with broad applicability

Peikun Zheng, Roman Zubatyuk, Wei Wu, Olexandr Isayev & Pavlo O. Dral

Nature Commun

15k Accesses

Journal of XXX

AIQM2 : the 2nd generation of AI models

By Dral's group

Journal of XXX

AIQM3 : the 3rd generation of AI models

By Dral's group

…… ……

Reviewers 🤨

* We might end up doing this...

UAIQM: Universal and Updatable AI-QM models

nature communications

Explore content · About the journal · Publish with us

nature > nature communications > articles > article

Article | Open access | Published: 02 December 2021

Artificial intelligence-enhanced quantum chemical method with broad applicability

Peikun Zheng, Roman Zubatyuk, Wei Wu, Olexandr Isayev & Pavlo O. Dral

Nature Commun

15k Accesses

Journal of XXX

AIQM2 : the 2nd generation of AI models

By Dral's group

Journal of XXX

AIQM3 : the 3rd generation of AI models

By Dral's group

....

Pinned

Pavlo Dral @PavloDral · Jul 12

It seems that putting a dozen of state of the art #AI methods in one big paper is not appreciated by some journals. Slice or Not to Slice: That is the Question 😅 youtube.com/watch?v=wvYDPq...
Preprint: doi.org/10.26434/chemr...

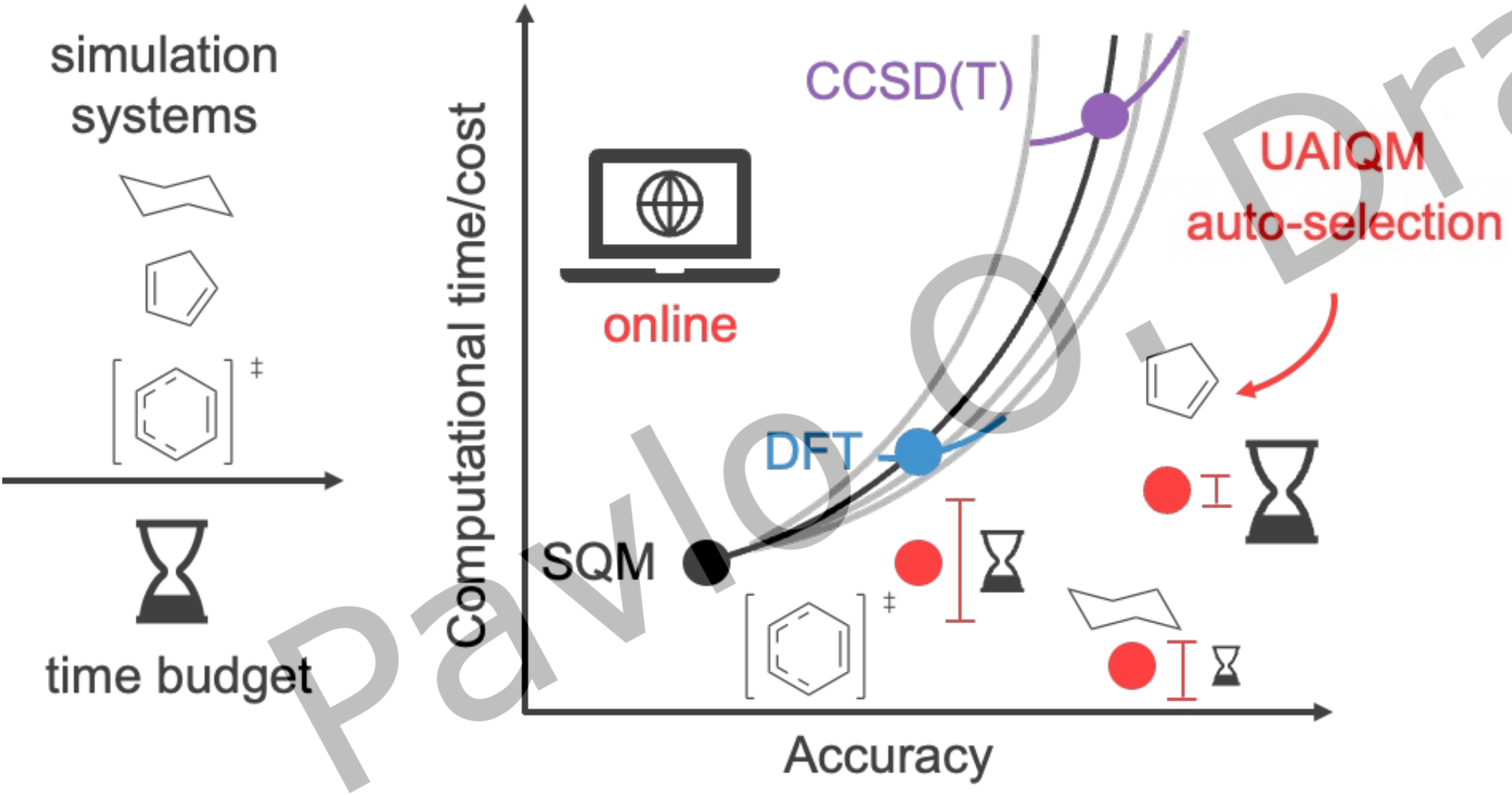
Slice	50%
Not slice	50%

12 votes · Final results

3 · 3 · 1K

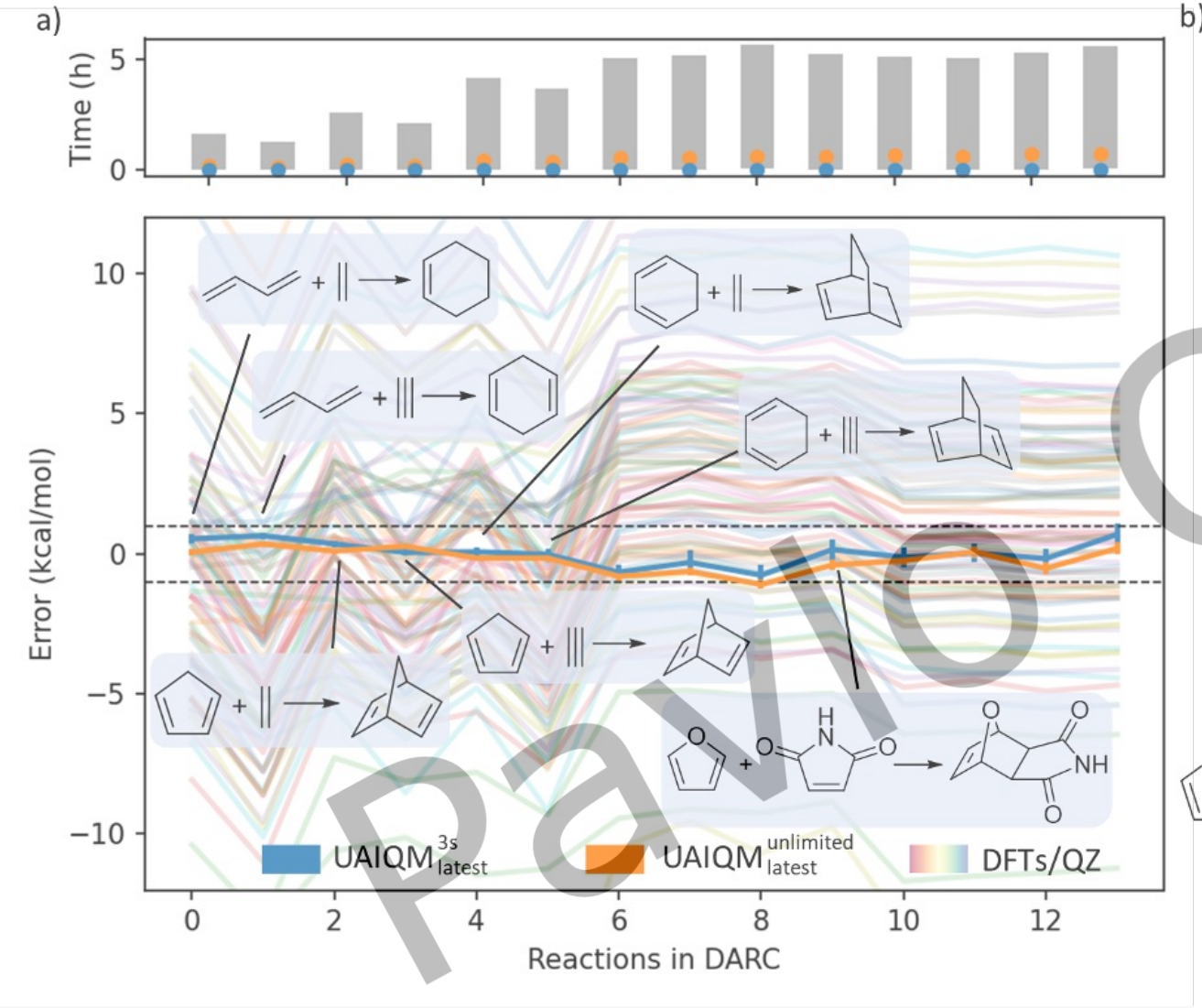
* We might end up doing this...

UAIQM: Universal and Updatable AI-QM models



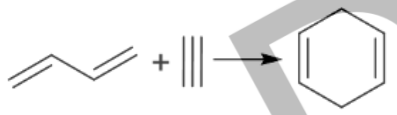
Y. Chen, Y.-F. Hou, O. Isayev, P. O. Dral. *Universal and Updatable Artificial Intelligence-Enhanced Quantum Chemical Foundational Models*. **2024**, *submitted*. https://doi.org/10.26434/chemrxiv-2024-604wb.

UAIQM: Universal and Updatable AI-QM models

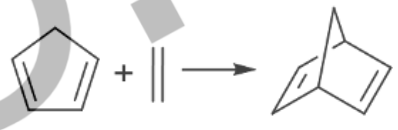


b)

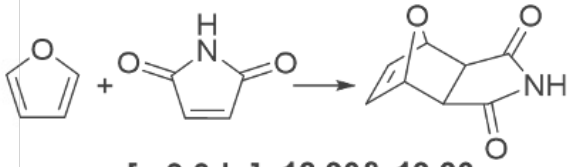
W1-F12 & CCSD(T)/CBS
DSD-BLYP-D3(BJ)/def2-QZVP
UAIQM (unlimited)
UAIQM (3 s time budget)



[~3.5 h] -60.80&-60.45
[~1.0 h] -59.67
[~0.1 h] -60.16±0.11
[~0.1 s] -60.44±0.10



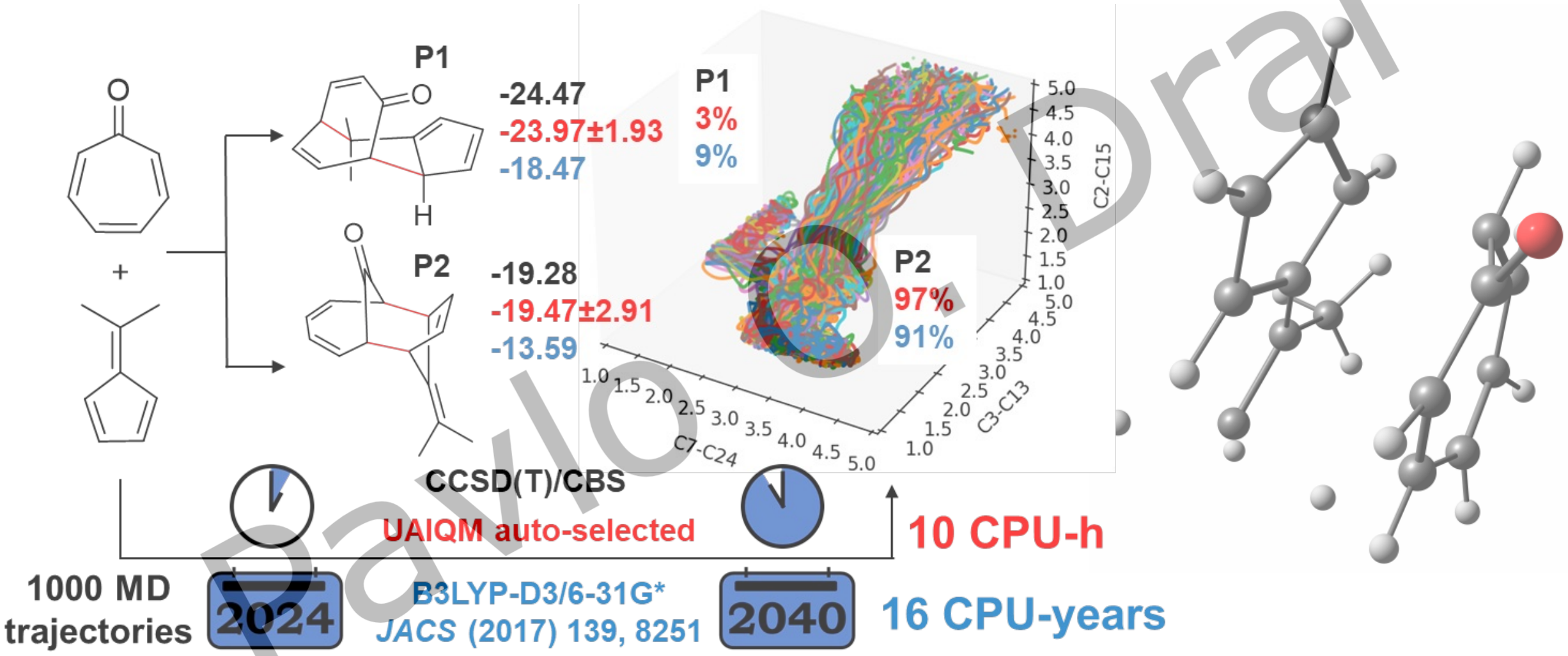
[~4.5 h] -29.9&-29.69
[~2.5 h] -27.93
[~0.3 h] -29.81±0.09
[~0.1 s] -29.55±0.13



[~9.0 h] -18.90&-19.30
[~5.0 h] -15.18
[~0.6 h] -19.31±0.19
[~0.1 s] -18.75±0.37

Y. Chen, Y.-F. Hou, O. Isayev, P. O. Dral. *Universal and Updatable Artificial Intelligence-Enhanced Quantum Chemical Foundational Models*. **2024**, *submitted*. https://doi.org/10.26434/chemrxiv-2024-604wb.

UAIQM: Universal and Updatable AI-QM models



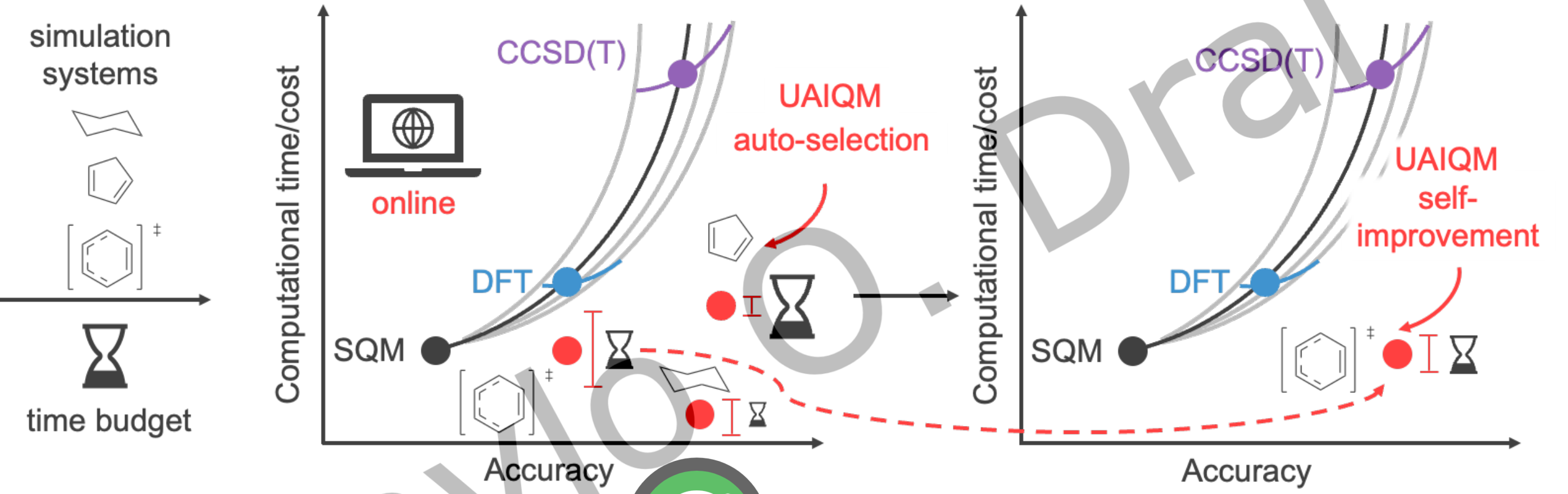
Y. Chen, Y.-F. Hou, O. Isayev, P. O. Dral. *Universal and Updatable Artificial Intelligence-Enhanced Quantum Chemical Foundational Models*. **2024**, *submitted*. https://doi.org/10.26434/chemrxiv-2024-604wb.

UAIQM: Universal and Updatable AI-QM models

```
================================================================================
WARNING: Uncertainty is too high for selected UAIQM method



Properties of molecule 1

Selected UAIQM method: uaiqm_gfn2xtbstar@cc
Selected version: 20240106

Standard deviation of ML contribution    :      0.00546625 Hartree        3.43012 kcal/mol
Baseline contribution                    :     -2.73848388 Hartree
NN contribution                          :     -0.27210359 Hartree
D4 contribution                          :     -0.00007046 Hartree
Total energy                             :     -3.01065793 Hartree

================================================================================
```

Y. Chen, Y.-F. Hou, O. Isayev, P. O. Dral. *Universal and Updatable Artificial Intelligence-Enhanced Quantum Chemical Foundational Models*. **2024**, *submitted*. https://doi.org/10.26434/chemrxiv-2024-604wb.

UAIQM: Universal and Updatable AI-QM models



Updating UAIQM Library...

Y. Chen, Y.-F. Hou, O. Isayev, P. O. Dral. *Universal and Updatable Artificial Intelligence-Enhanced Quantum Chemical Foundational Models*. **2024**, *submitted*. https://doi.org/10.26434/chemrxiv-2024-604wb.

Element availability

Excellent ~CCSD(T) level

Supported max. DFT level

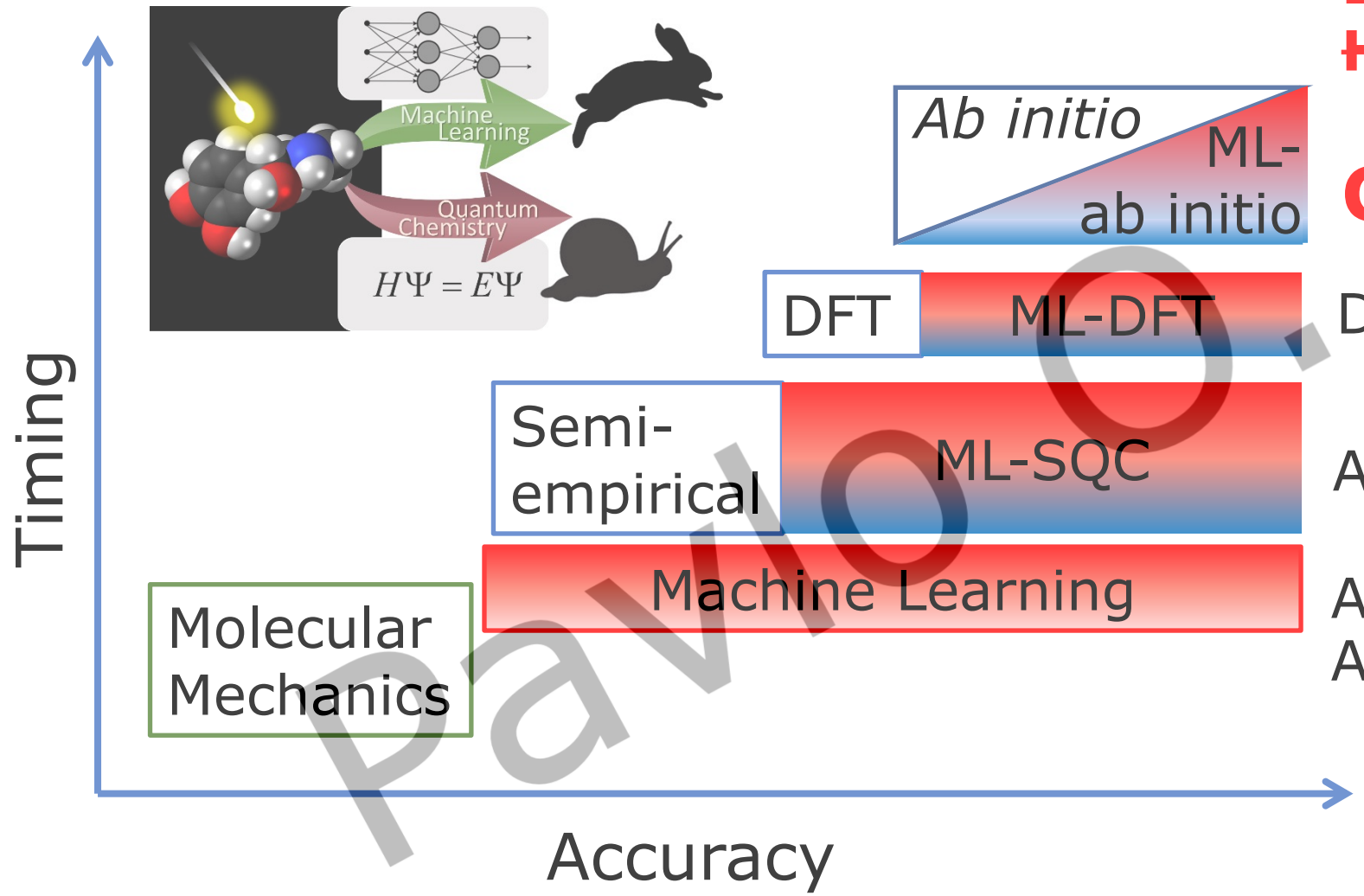
Unsupported

IA	IIA	IIIB	IVB	VB	VIB	VIIB	VIIIB	VIIIB	VIIIB	IB	IIB	IIIA	IVA	VA	VIA	VIIA	VIIIA
1 H Hydrogen 1.01																	2 He Helium 4.00
3 Li Lithium 6.94	4 Be Beryllium 9.01											5 B Boron 10.81	6 C Carbon 12.01	7 N Nitrogen 14.01	8 O Oxygen 16.00	9 F Fluorine 19.00	10 Ne Neon 20.18
11 Na Sodium 22.99	12 Mg Magnesium 24.31											13 Al Aluminum 26.98	14 Si Silicon 28.09	15 P Phosphorus 30.97	16 S Sulfur 32.06	17 Cl Chlorine 35.45	18 Ar Argon 39.95
19 K Potassium 39.10	20 Ca Calcium 40.08	21 Sc Scandium 44.96	22 Ti Titanium 47.87	23 V Vanadium 50.94	24 Cr Chromium 52.00	25 Mn Manganese 54.94	26 Fe Iron 55.85	27 Co Cobalt 58.93	28 Ni Nickel 58.69	29 Cu Copper 63.55	30 Zn Zinc 65.38	31 Ga Gallium 69.72	32 Ge Germanium 72.63	33 As Arsenic 74.92	34 Se Selenium 78.97	35 Br Bromine 79.90	36 Kr Krypton 83.80
37 Rb Rubidium 85.47	38 Sr Strontium 87.62	39 Y Yttrium 88.91	40 Zr Zirconium 91.22	41 Nb Niobium 92.91	42 Mo Molybdenum 95.95	43 Tc Technetium (98)	44 Ru Ruthenium 101.07	45 Rh Rhodium 102.91	46 Pd Palladium 106.42	47 Ag Silver 107.87	48 Cd Cadmium 112.41	49 In Indium 114.82	50 Sn Tin 118.71	51 Sb Antimony 121.76	52 Te Tellurium 127.60	53 I Iodine 126.90	54 Xe Xenon 131.29
55 Cs Cesium 132.91	56 Ba Barium 137.33	57 - 71 Lanthanides	72 Hf Hafnium 178.49	73 Ta Tantalum 180.95	74 W Tungsten 183.84	75 Re Rhenium 186.21	76 Os Osmium 190.23	77 Ir Iridium 192.22	78 Pt Platinum 195.08	79 Au Gold 196.97	80 Hg Mercury 200.59	81 Tl Thallium 204.38	82 Pb Lead 207.20	83 Bi Bismuth 208.98	84 Po Polonium (209)	85 At Astatine (210)	86 Rn Radon (222)
87 Fr Francium (223)	88 Ra Radium (226)	89 - 103 Actinides	104 Rf Rutherfordium (265)	105 Db Dubnium (268)	106 Sg Seaborgium (271)	107 Bh Bohrium (270)	108 Hs Hassium (277)	109 Mt Meitnerium (276)	110 Ds Darmstadtium (281)	111 Rg Roentgenium (280)	112 Cn Copernicium (285)	113 Nh Nihonium (284)	114 Fl Flerovium 289	115 Mc Moscovium (288)	116 Lv Livermorium (293)	117 Ts Tennessine (294)	118 Og Oganesson (294)

57 La Lanthanum 138.91	58 Ce Cerium 140.12	59 Pr Praseodymium 140.91	60 Nd Neodymium 144.24	61 Pm Promethium (145)	62 Sm Samarium 150.36	63 Eu Europium 151.96	64 Gd Gadolium 157.25	65 Tb Terbium 158.93	66 Dy Dysprosium 162.50	67 Ho Holmium 164.93	68 Er Erbium 167.26	69 Tm Thulium 168.93	70 Yb Ytterbium 173.05	71 Lu Lutetium 174.97
89 Ac Actinium (227)	90 Th Thorium 232.04	91 Pa Protactinium 231.04	92 U Uranium 238.03	93 Np Neptunium (237)	94 Pu Plutonium (244)	95 Am Americium (243)	96 Cm Curium (247)	97 Bk Berkelium (247)	98 Cf Californium (251)	99 Es Einsteinium (252)	100 Fm Fermium (257)	101 Md Mendelevium (258)	102 No Nobelium (259)	103 Lr Lawrencium (262)

Can we do better?

P. O. Dral, M. Barbatti, *Nat. Rev. Chem.* **2021,** *5*, 388



Lots of models!
~~How to choose?~~

Choose them all!

DM21, CF22D, DENS24…

AIQM1

ANI-1ccx, AIMnet-2, ANI-1xnr, …

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

Basic properties and reaction energies of small systems

Reaction barrier heights

Intermolecular noncovalent interactions

Intramolecular noncovalent interactions

Reaction energies for large systems and isomerization reactions

?

MN12SX

BLYP

B3LYP

SCAN

DSD-BLYP

PBE0

revPBE

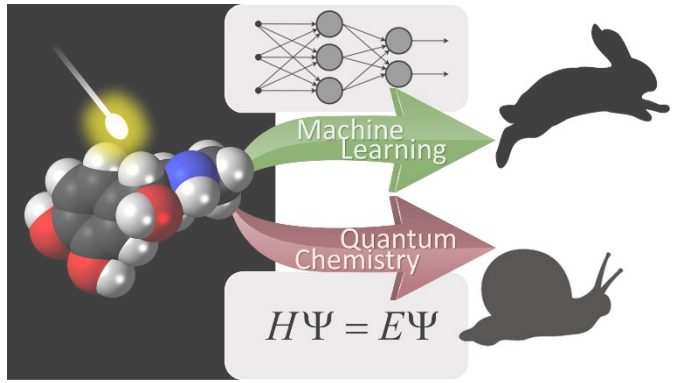
ωB97X-V

.....

Which one is the best?

Universal models (no training needed)

P. O. Dral, M. Barbatti, *Nat. Rev. Chem.* **2021,** *5*, 388



Timing

Ab initio

ML-ab initio

DFT

ML-DFT

DM21, CF22D, DENS24…

Semi-empirical

ML-SQC

AIQM1

AIQM1: P. Zheng, R. Zubatyuk, W. Wu, O. Isayev, P. O. Dral, *Nat. Commun.* **2021**, *12*, 7022

Machine Learning

ANI-1ccx, AIMnet-2, ANI-1xnr, …

Molecular Mechanics

Accuracy

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

DENS24 series – ML density functionals

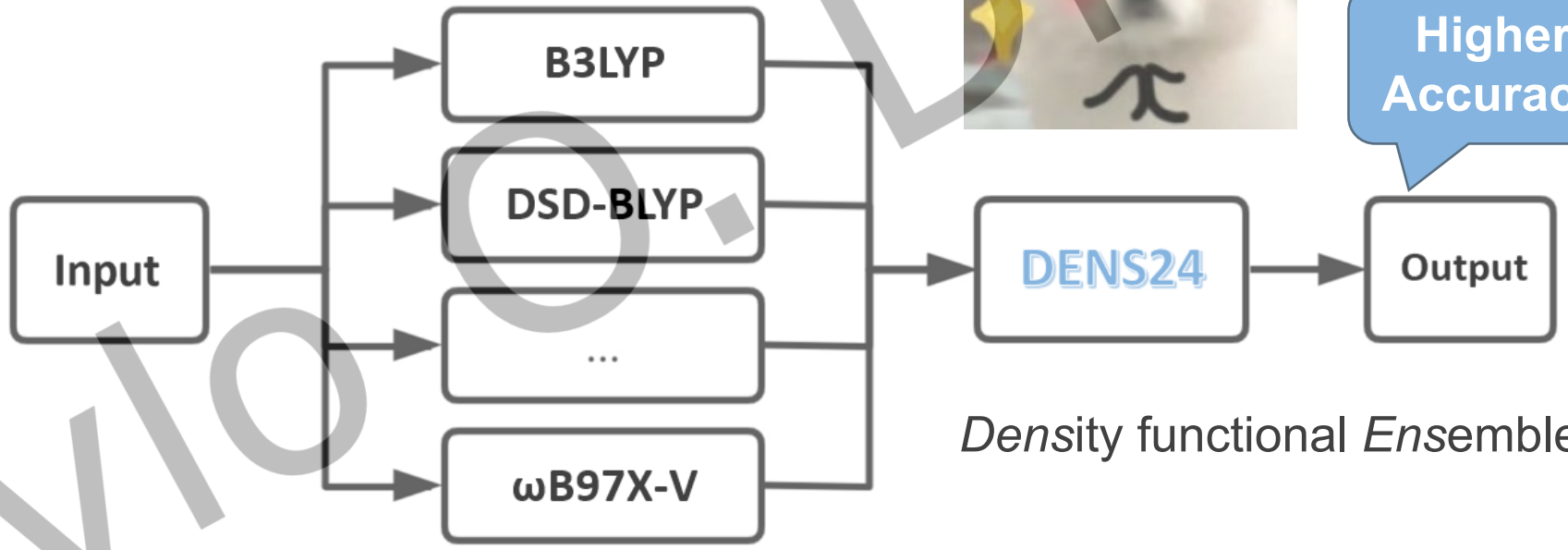
GMTKN55 benchmark:

- Double-hybrid variant: 1.62 kcal/mol
- Hybrid variant: 2.86 kcal/mol

Y. Rui, Y. Chen, E. Ivanova, I. Grabowski, P. O. Dral. The best DFT functional is the ensemble of functionals. Preprint on *ChemRxiv*, **2024**. https://doi.org/10.26434/chemrxiv-2024-2g7zr

DFT ensembles

Choose them all!



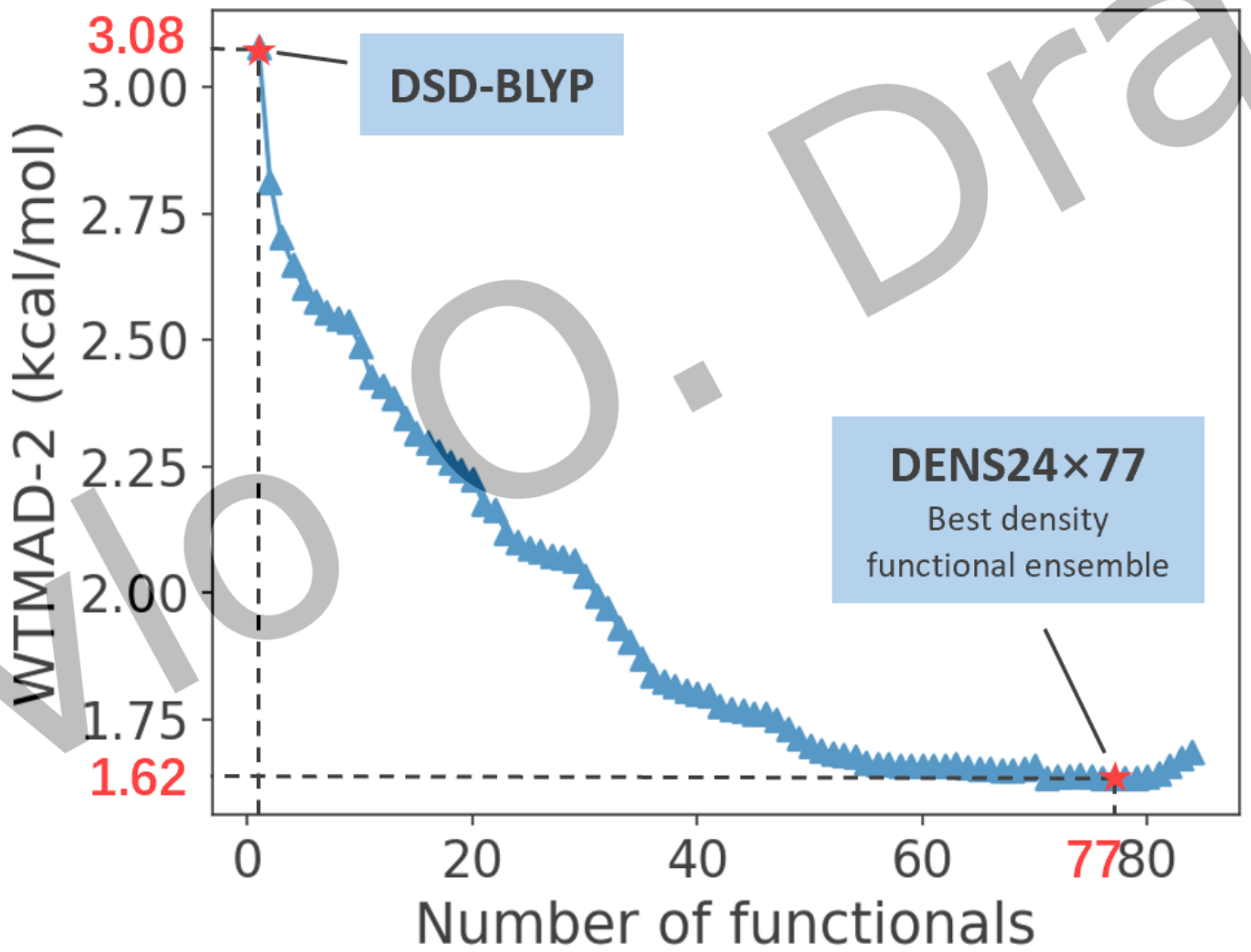
Ensemble

*Den*sity functional *Ens*embles

Weak learner → Strong learner

Y. Rui, Y. Chen, E. Ivanova, I. Grabowski, P. O. Dral. The best DFT functional is the ensemble of functionals. Preprint on *ChemRxiv*, **2024**. https://doi.org/10.26434/chemrxiv-2024-2g7zr

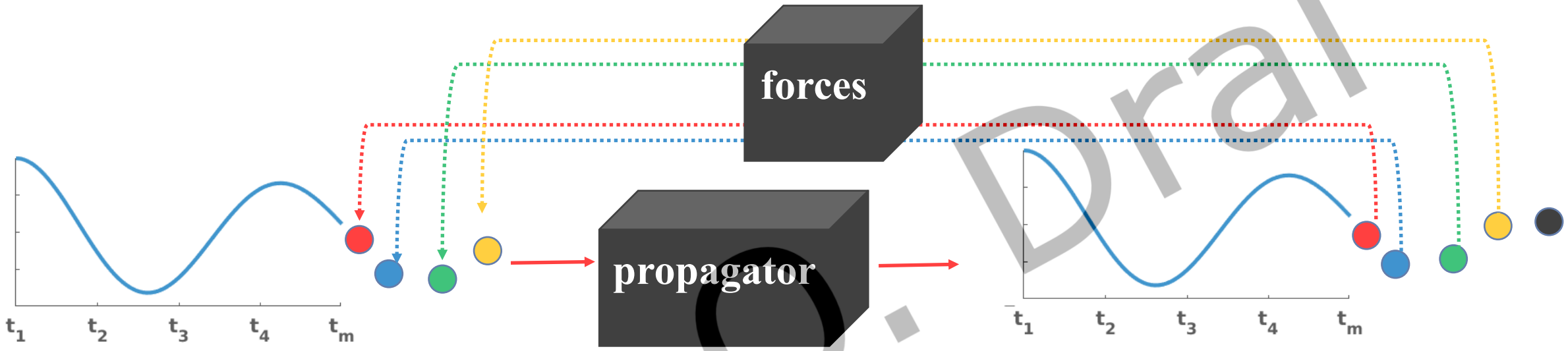
More functionals – better accuracy



Y. Rui, Y. Chen, E. Ivanova, I. Grabowski, P. O. Dral. The best DFT functional is the ensemble of functionals. Preprint on *ChemRxiv*, **2024**. https://doi.org/10.26434/chemrxiv-2024-2g7zr

What else can we do differently in computational chemistry?

MD



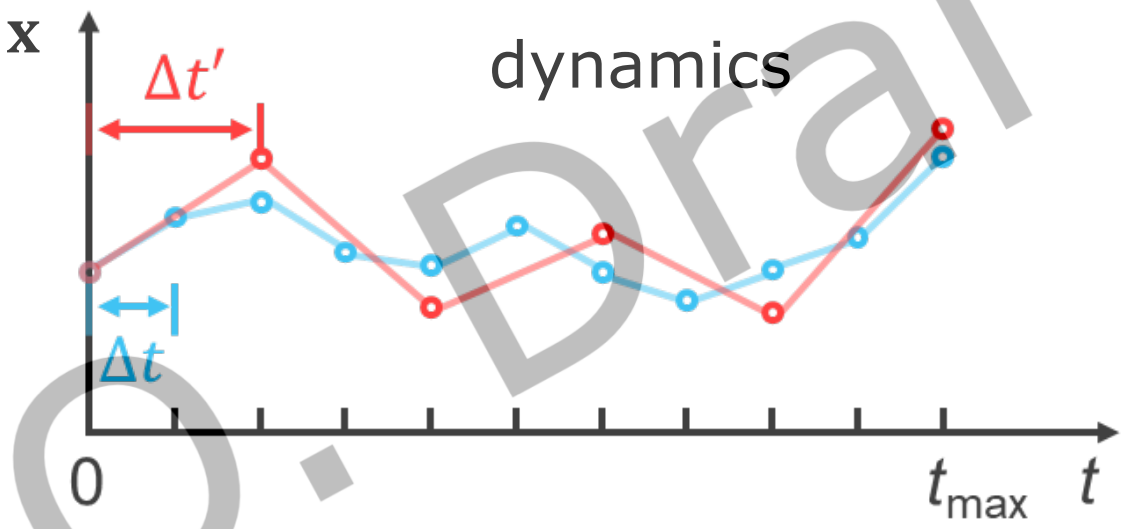
$$F_{A,d} = -\frac{\partial E}{\partial x_{A,d}} = -\frac{\partial E^{\text{machine learning potential}}(\mathbf{x})}{\partial x_{A,d}}$$

Adapted from animation by Arif Ullah

Can we do better?

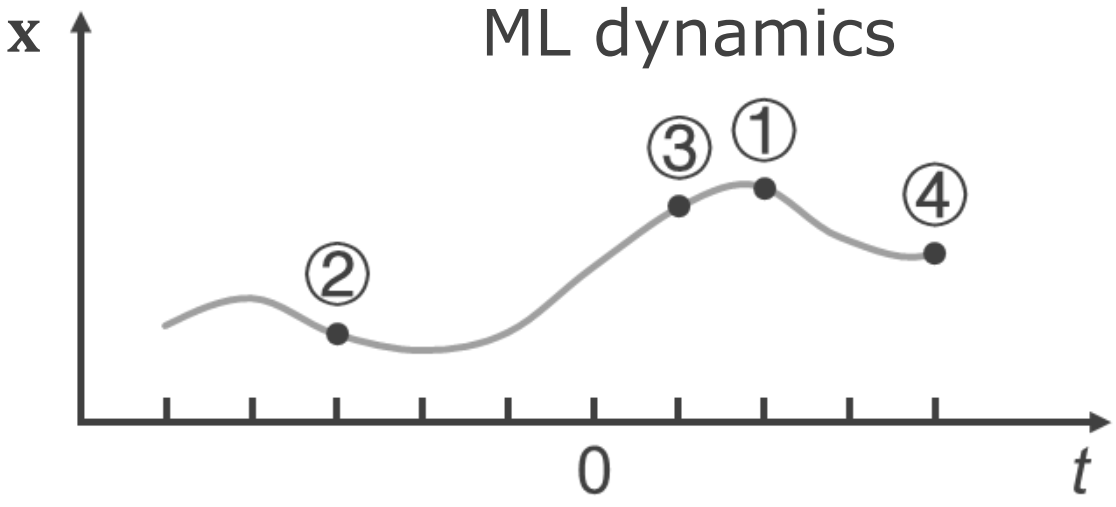
~~dynamics propagation is~~

- ~~Iterative (non-parallelizable)~~
- ~~Depends on time step~~
- ~~Discrete~~



Directly learning dynamics

$$\mathbf{x}_t = f(\mathbf{x}_0, \mathbf{v}_0, t)$$

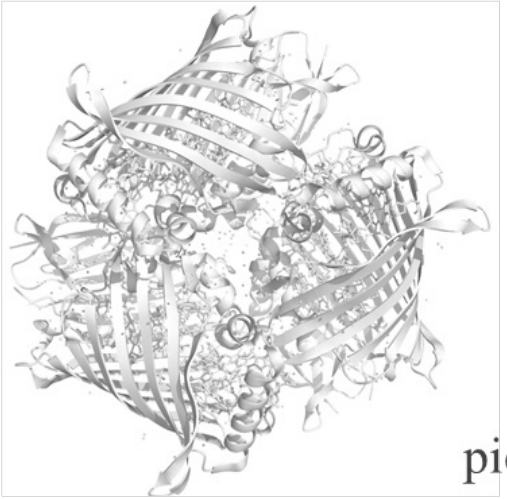


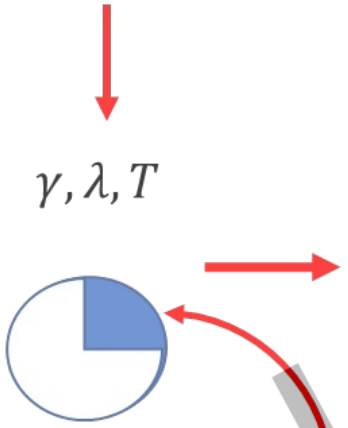
AI-QD (artificial intelligence-based quantum dynamics)

γ = characteristic frequency
λ = reorganization energy
T = temperature

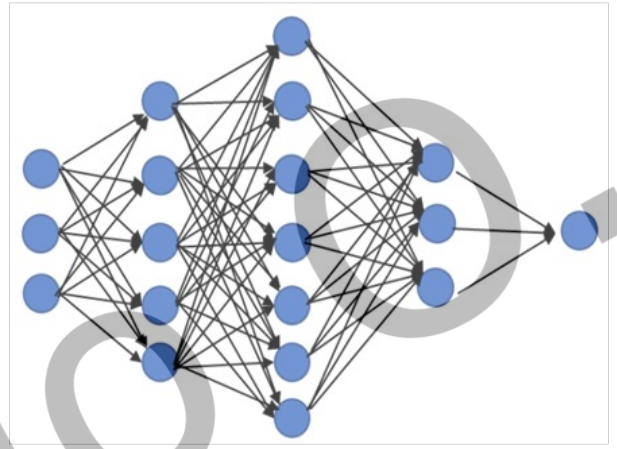
$$\boldsymbol{\rho}(\text{time}) = f[\text{time}; \text{simulation parameters}]$$

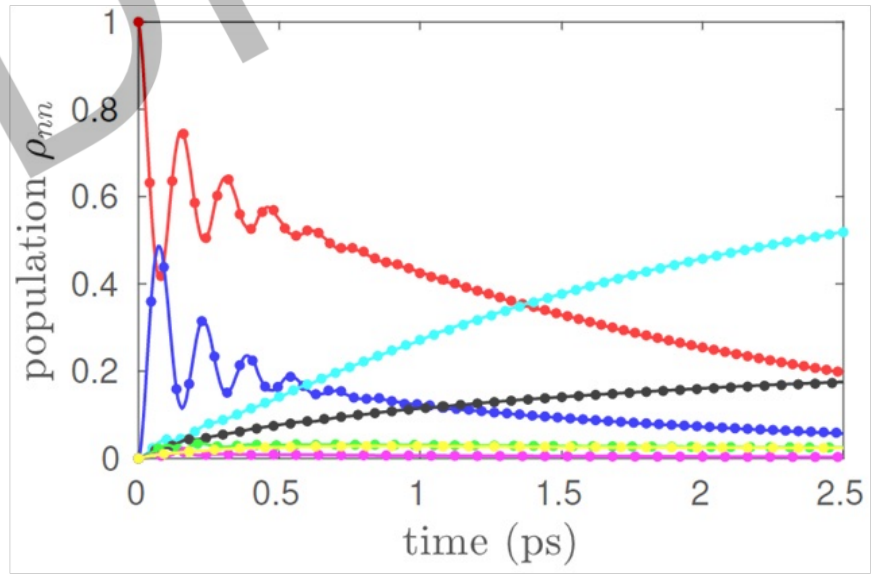
PDB code: 3ENI





2.5ps



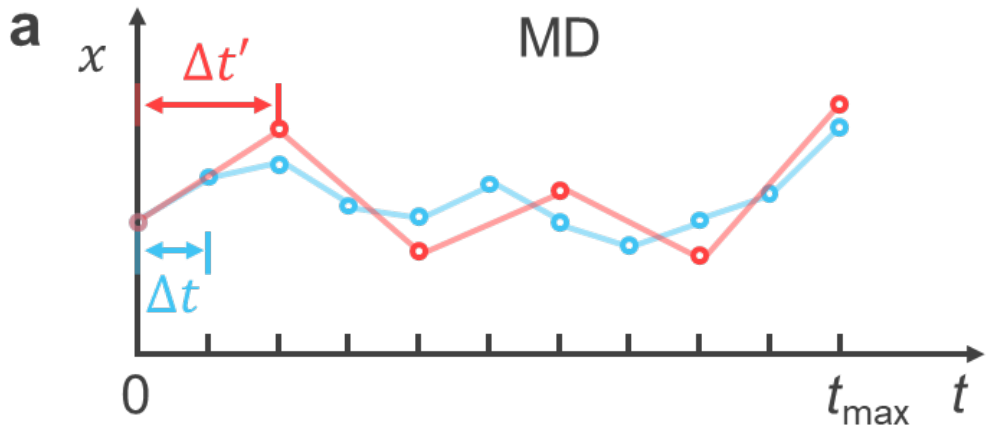


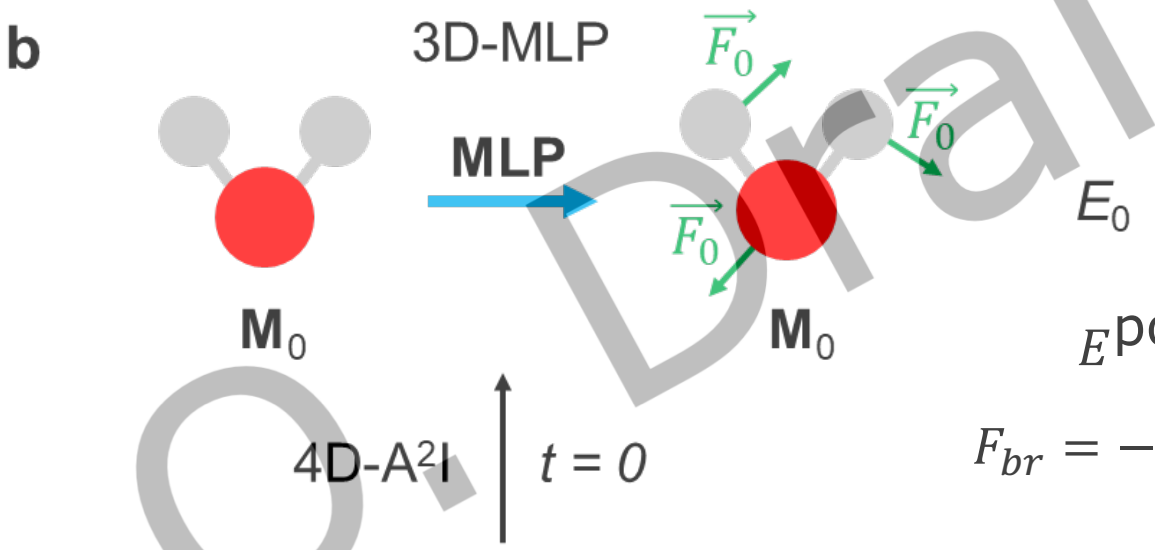
Dots: Reference
Line: AI-QD

7-sites Fenna–Matthews–Olson (FMO) complex

A. Ullah, P. O. Dral. Predicting the future of excitation energy transfer in light-harvesting complex with artificial intelligence-based quantum dynamics. *Nat. Commun.* **2022**, *13*, 1930

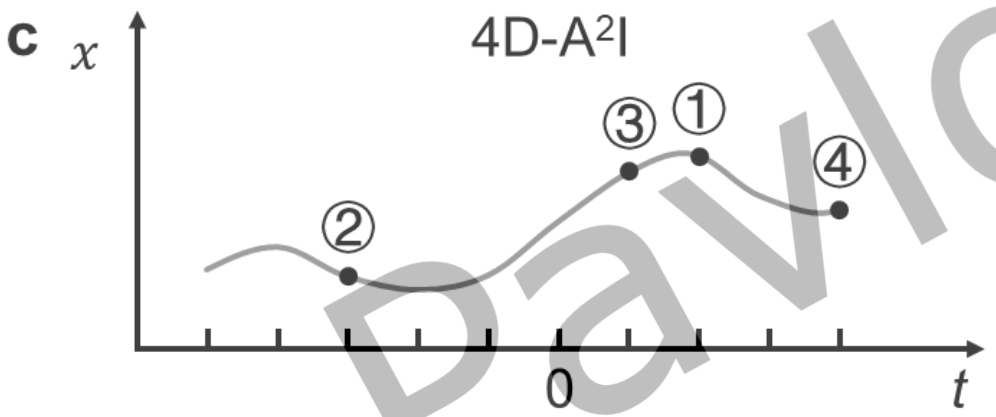
Learning dynamics: 4D-spacetime models

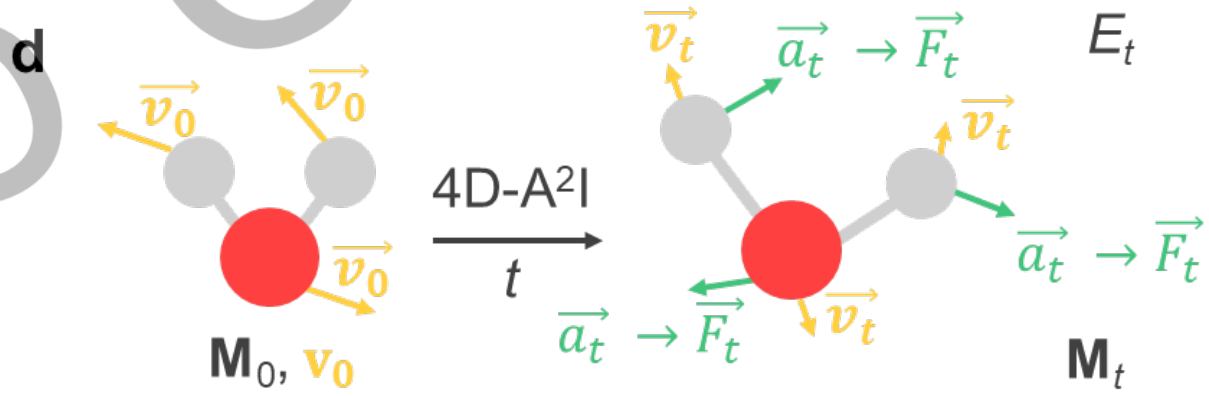




$$E^{\text{pot}} = f(\mathbf{M})$$

$$F_{br} = -\partial E^{\text{pot}}/\partial x_{br}$$

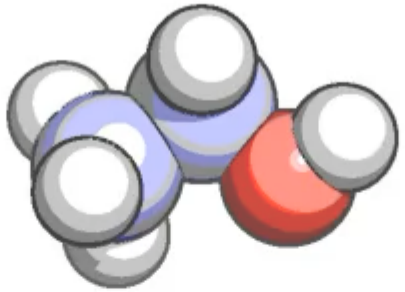




$$\mathbf{x}_t = f(\mathbf{x}_0, \mathbf{v}_0, t) \longrightarrow v_{br,t} = \frac{dx_{br,t}}{dt} \longrightarrow a_{br,t} = \frac{dv_{br,t}}{dt} \longrightarrow F_{br,t} = m_b a_{br,t}$$

F. Ge, L. Zhang, Y.-F. Hou, Y. Chen, A. Ullah, P. O. Dral. *J. Phys. Chem. Lett.* **2023**, *14*, 7732

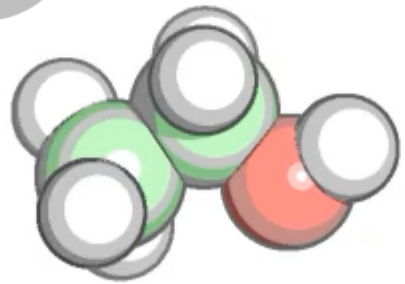
Ref



Very slow

Stable dynamics for > 1 ns

GICnet

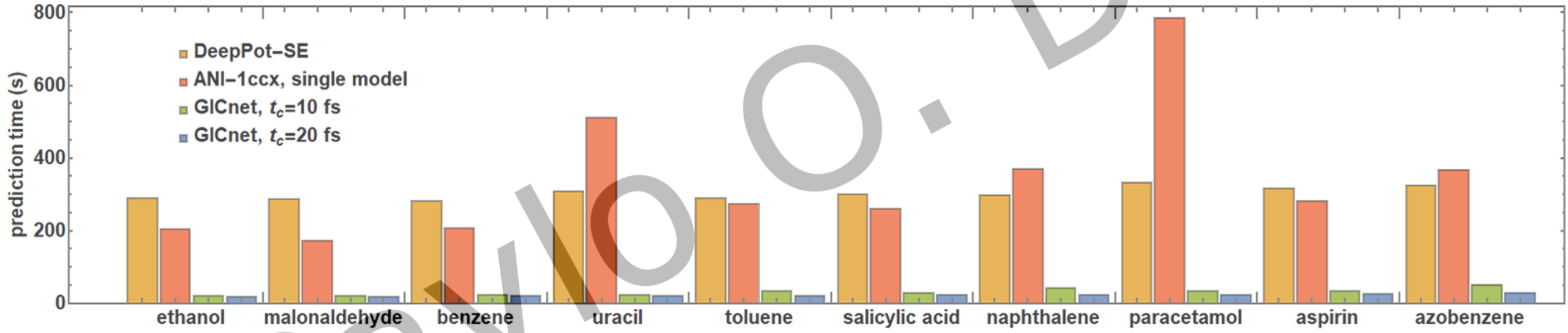


Very fast, e.g., 1 ps trajectory (time step 0.05 fs) within 1 minute

0.0 fs

F. Ge, L. Zhang, Y.-F. Hou, Y. Chen, A. Ullah, P. O. Dral. *J. Phys. Chem. Lett.* **2023**, *14*, 7732

Direct learning of molecular dynamics



F. Ge, L. Zhang, Y.-F. Hou, Y. Chen, A. Ullah, P. O. Dral. *J. Phys. Chem. Lett.* **2023**, *14*, 7732

Can we do even better?

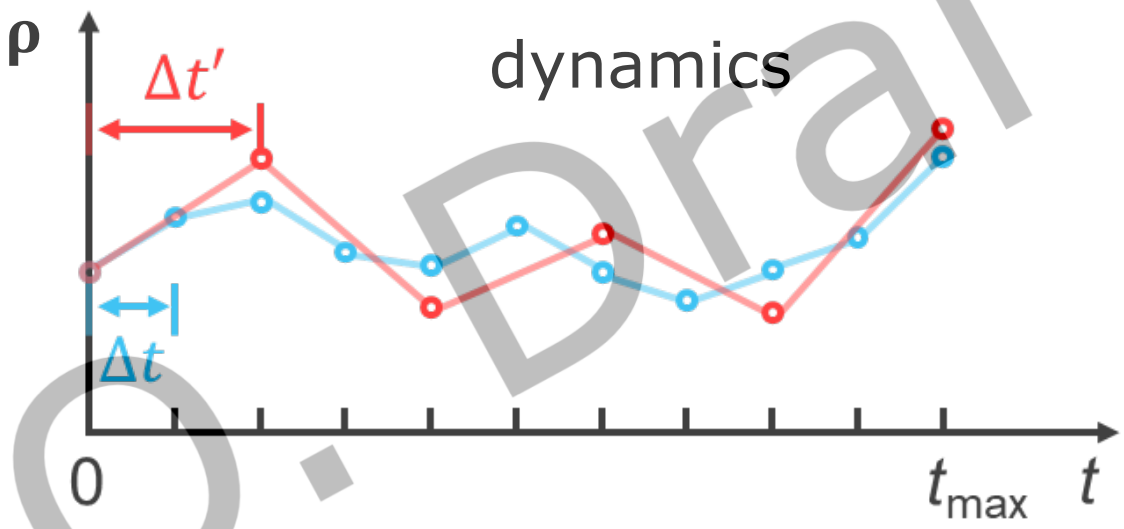
$$\cancel{\boldsymbol{\rho}(t) = f[\boldsymbol{\rho}(t - \Delta t)]}$$

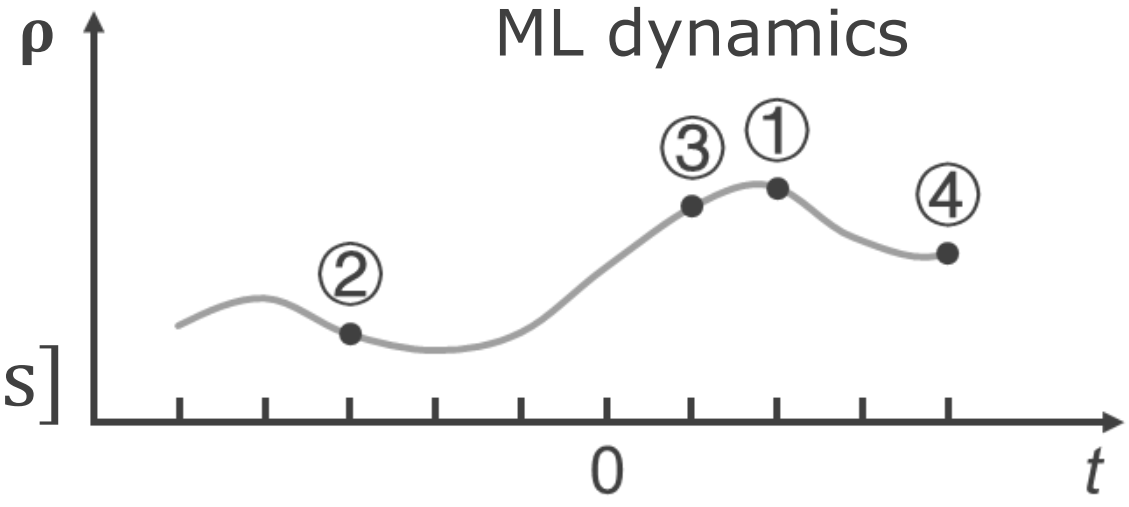
dynamics propagation is

- ~~computationally expensive~~
- ~~recursive (iterative)~~

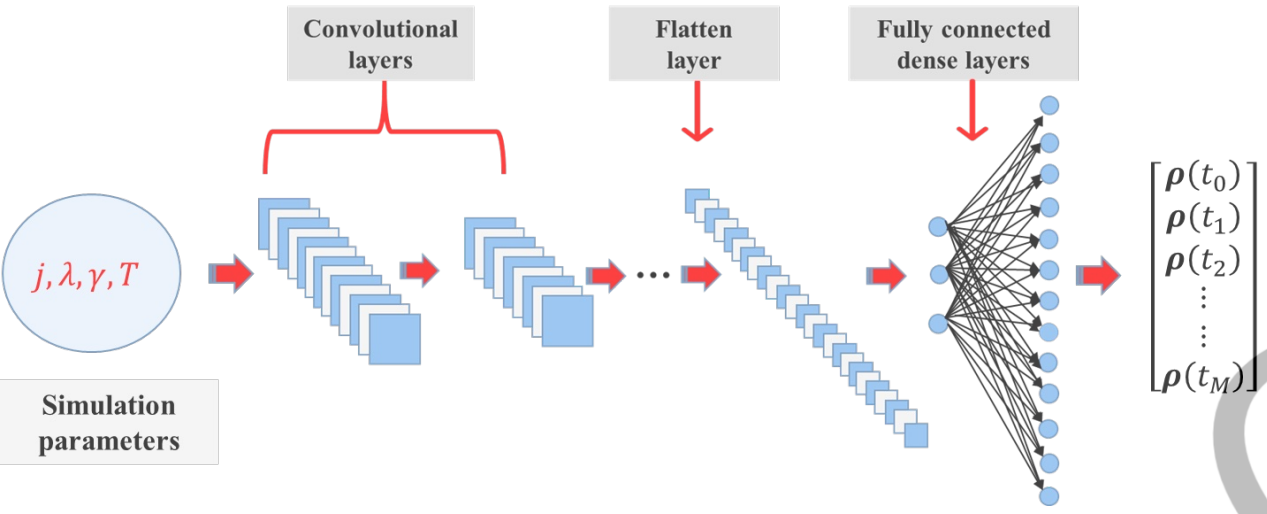
$$\cancel{\boldsymbol{\rho}(t) = f[t; \text{other parameters}]}$$

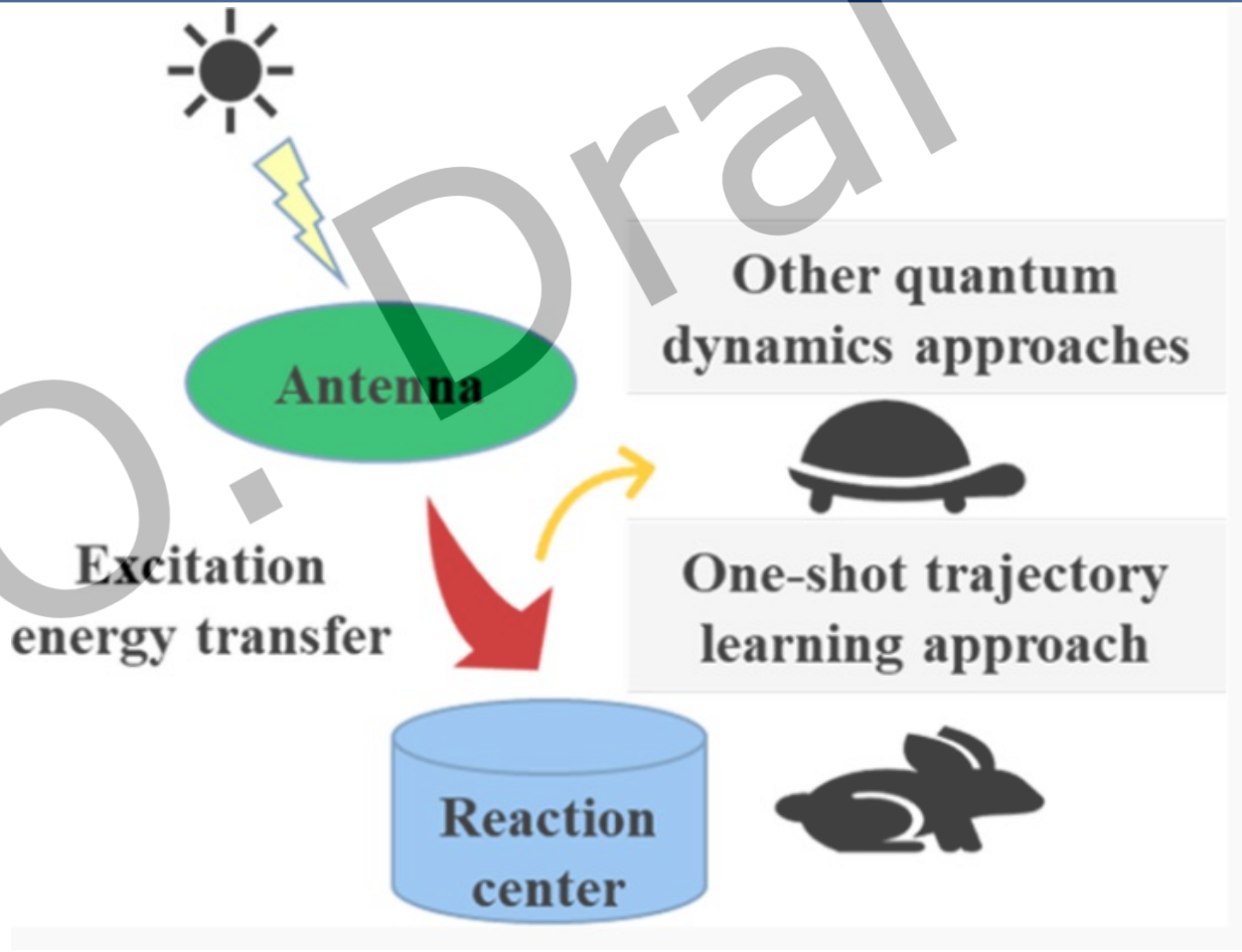
$$\{\boldsymbol{\rho}(t_k)\}_{k=0}^{M} = f[\text{other parameters}]$$





One-Shot Trajectory Learning (OSTL)





$$\{\boldsymbol{\rho}(t_k)\}_{k=0}^{M} = f[\text{other parameters}]$$

- **10 ps long dynamics in just 70 ms**
- good for massive simulation in parameter space

A. Ullah, P. O. Dral. *J. Phys. Chem. Lett.* **2022**, 6037

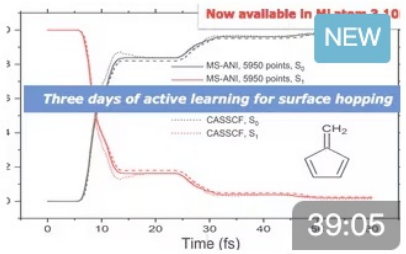
XACS团队 动态 投稿 60 合集和列表 1 搜索视频、动态

关注数	粉丝数	获赞数	播放数
5	1018	842	2.6万

TA的视频 36

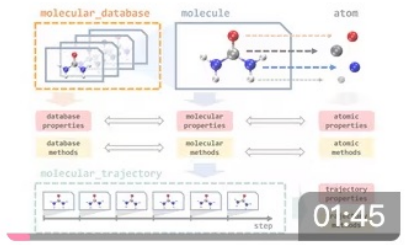
最新发布 最多播放 最多收藏

播放全部 更多



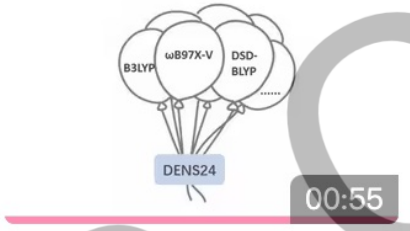
面跳跃动力学的主动学习

98 21小时前



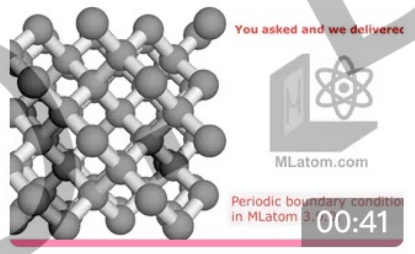
MLatom@XACS中的Python数据对象详解

179 8–14



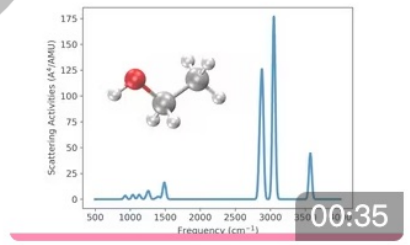
选DFT的泛函？不如全都要！

797 8–7



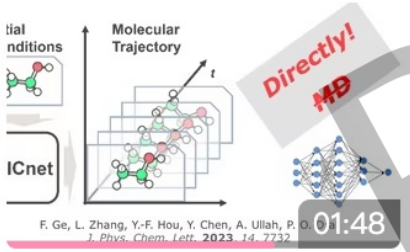
终于！MLatom@XACS支持周期性边界条件啦！

559 7–31



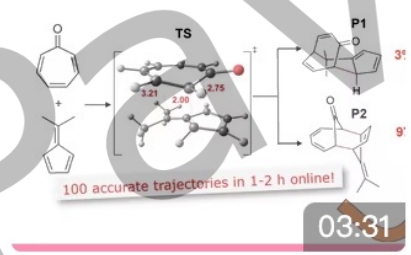
分子拉曼光谱模拟现在上线啦！

2435 7–24



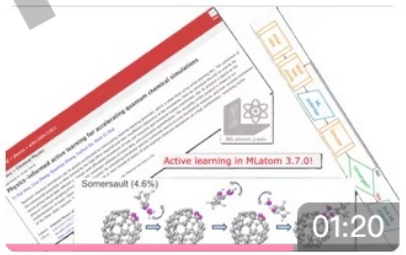
直接学习分子动力学轨迹！

2923 7–17



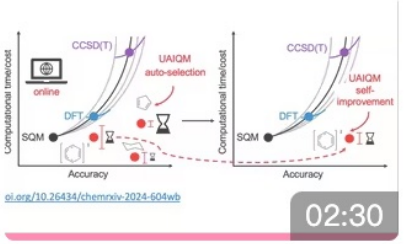
在线模拟两可反应！

571 7–10



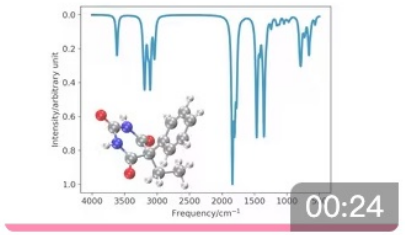
主动学习：数据高效的机器学习势构建

549 7–3



UAIQM：通用且可更新的AI模型让你火力全开！

1122 6–27



分子红外光谱模拟现在上线啦！

3543 6–20

XACS

@XACScloud · 184 subscribers · 41 videos

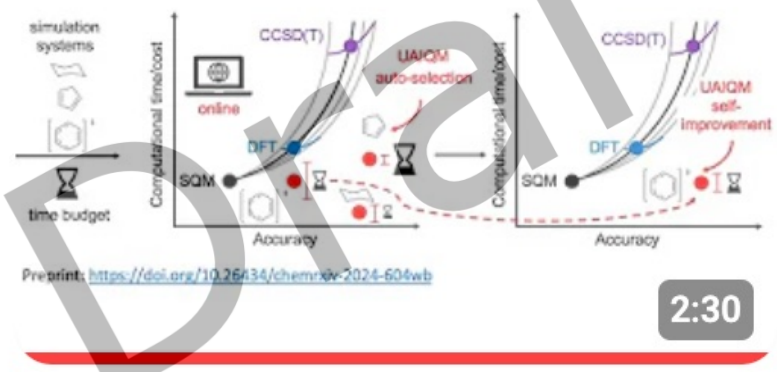
Online resources with AI, computational chemistry & chemical bonding analysis ...more

xacscloud.com

Subscribed

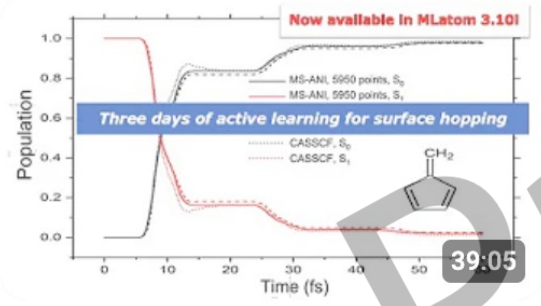
Home Videos Shorts Live Playlists

Latest Popular Oldest



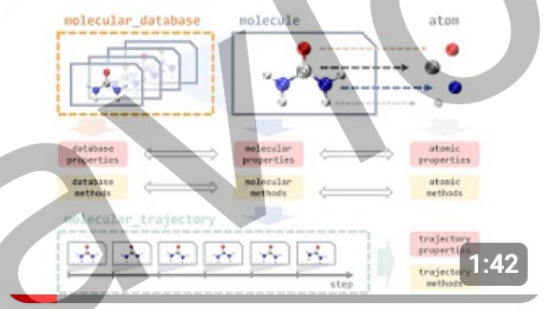
Supercharge your computational chemistry with the universal and...

165 views • 1 month ago



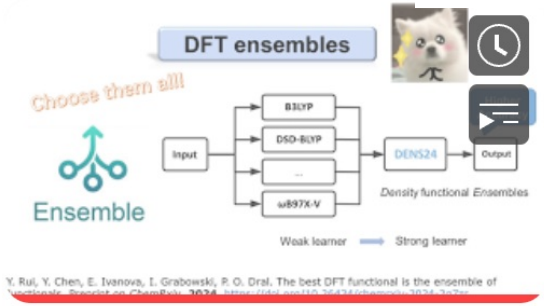
Active learning for surface hopping dynamics

20 views • 16 hours ago



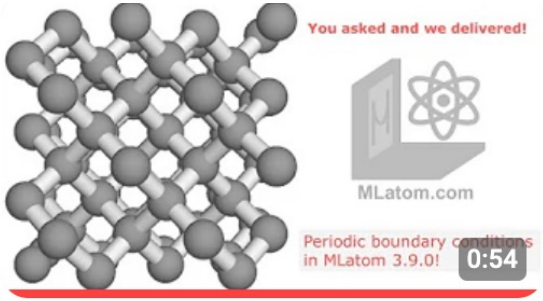
Data in MLatom's Python API

53 views • 7 days ago



Not sure which DFT functional to choose? Choose them all!

84 views • 2 weeks ago



Finally! Periodic boundary conditions in MLatom

92 views • 3 weeks ago

Subscribe to updates

Wechat

Follow
XACS account

QQ Group

Answering your
questions in real time

Slack

Answering your
questions in real time

Facebook

Join
the XACS group

Our other channels:
Twitter/X: XACSprogram
Threads: XACSprogram
YouTube: XACScloud
bilibili: XACS团队

Tutorial's Jupyter Notebooks

Please check the link
[and register on XACS cloud – optional]

before lunch break

Questionary

https://github.com/JakubMartinka/karlsruhe2024

XACS
Xiamen Atomistic Computing Suite
XACScloud.com

MLatom.com

Tutorial: ML potentials I

Pavlo O. Dral

Xiamen University, P.R. China

Visiting Professor in
Nicolaus Copernicus University, Poland

10 September 2024

Tutorial's Jupyter Notebooks

Please check the link
[and register on XACS cloud – optional]

before lunch break

Questionary

https://github.com/JakubMartinka/karlsruhe2024

Open source (on github, pip install…); also run on XACScloud.com

Machine Learning

Simulations

MLatom

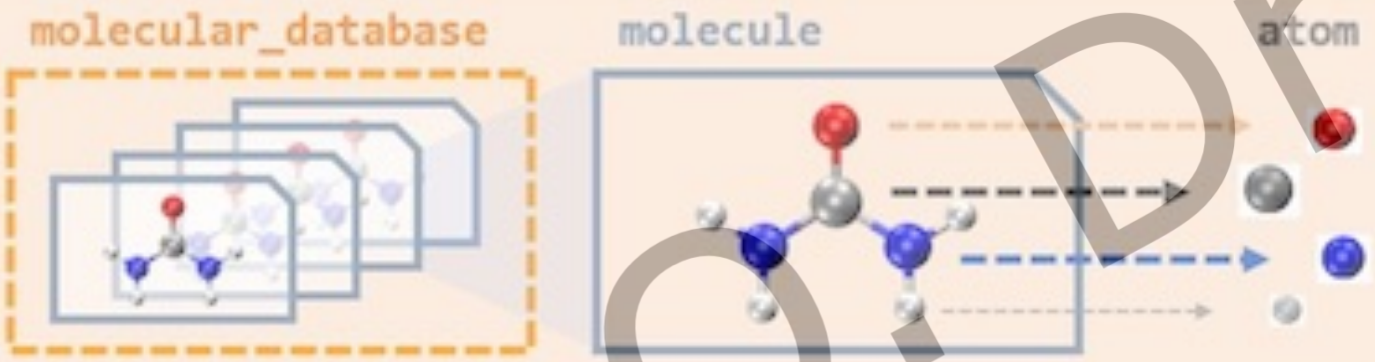
MLatom@GitHub
≠
MLatom@XACS

Quantum Chemistry

Data

Have it your way…

- Data in different formats and types V3



- Python API V3

```
import mlatom as ml
```

- Input file

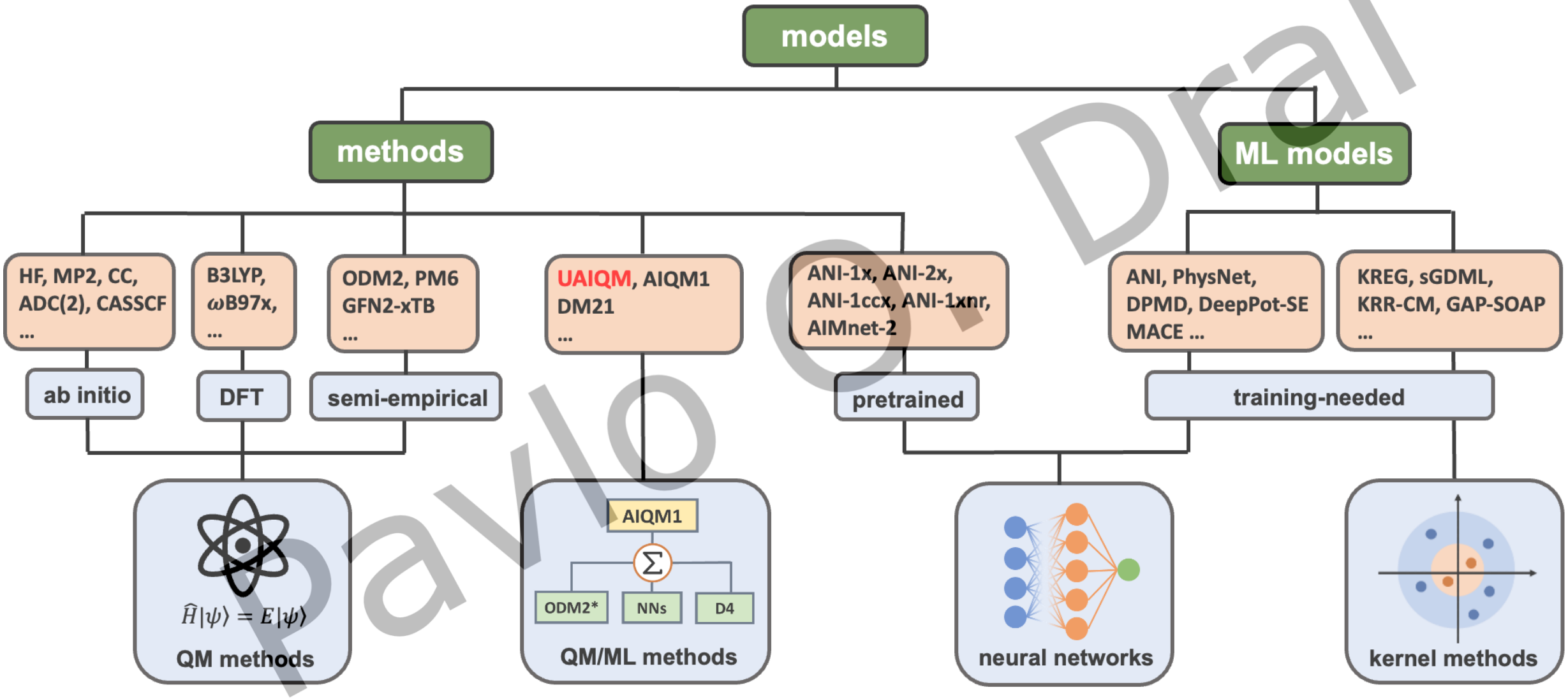
```
ANI-1ccx
geomopt
xyzfile=init.xyz
optxyz=opt.xyz
```

- Command line

```
> $mlatom ANI-1ccx geomopt xyzfile=init.xyz optxyz=opt.xyz
```


P. O. Dral, F. Ge, Y.-F. Hou, P. Zheng, Y. Chen, M. Barbatti, O. Isayev, C. Wang, B.-X. Xue, M. Pinheiro Jr, Y. Su, Y. Dai, Y. Chen, S. Zhang, L. Zhang, A. Ullah, Q. Zhang, Y. Ou. *J. Chem. Theory Comput.* **2024,** *20*, 1193

AI-enhanced computational chemistry

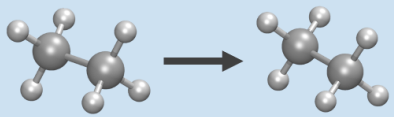


Updated based on: P. O. Dral, F. Ge, Y.-F. Hou, P. Zheng, Y. Chen, M. Barbatti, O. Isayev, C. Wang, B.-X. Xue, M. Pinheiro Jr, Y. Su, Y. Dai, Y. Chen, S. Zhang, L. Zhang, A. Ullah, Q. Zhang, Y. Ou. *JCTC* **2024,** *20*, 1193

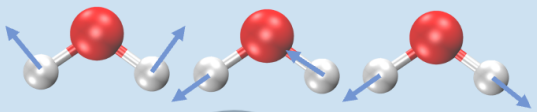
AI-enhanced computational chemistry

Single point calculations — Energies, forces, Hessian matrix…

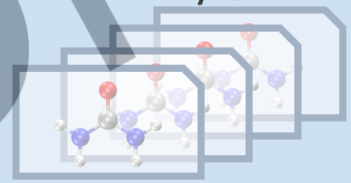
Geometry optimizations



Frequency calculations



Molecular dynamics



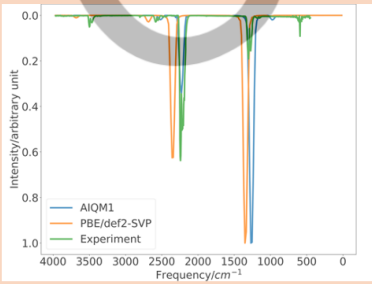
+PIMD coming soon…

Thermochemistry calculations

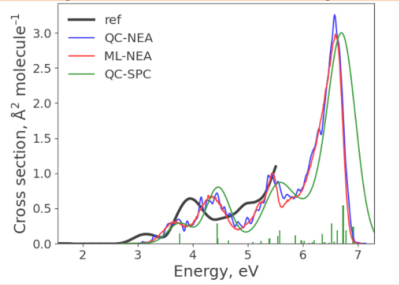
Heat of formation:

$$\Delta H_{\mathrm{at},T} = \left[\sum_A H_T(A)\right] - H_T$$

(Ro)vibrational spectra



One-photon UV/vis spectra



Properties & spectra

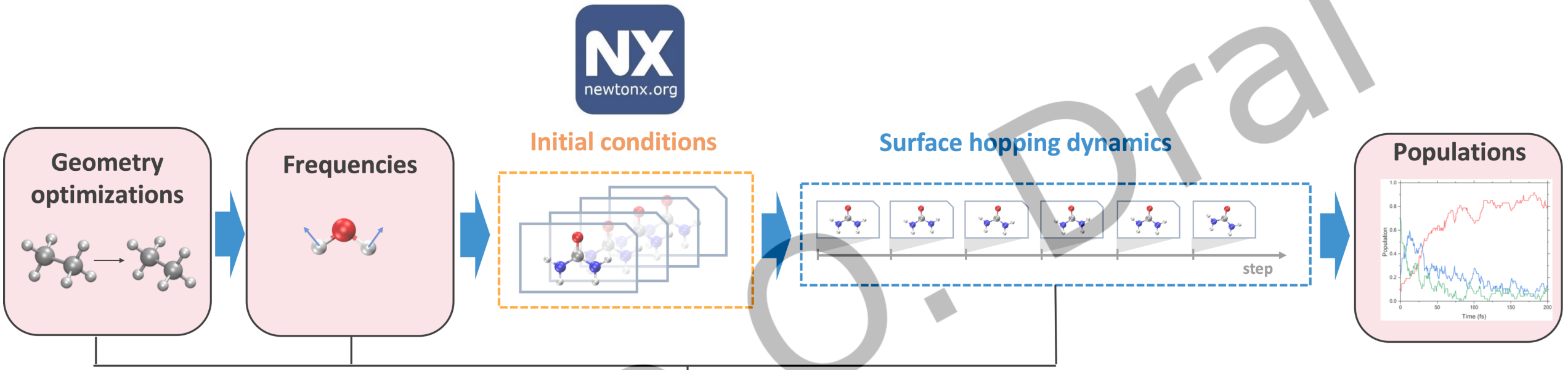
Simulations

Quantum dissipative dynamics

Two-photon absorption spectra

P. O. Dral, F. Ge, Y.-F. Hou, P. Zheng, Y. Chen, M. Barbatti, O. Isayev, C. Wang, B.-X. Xue, M. Pinheiro Jr, Y. Su, Y. Dai, Y. Chen, S. Zhang, L. Zhang, A. Ullah, Q. Zhang, Y. Ou. *J. Chem. Theory Comput.* **2024,** *20*, 1193

Surface-hopping dynamics



Single point calculations

$$E, \frac{\partial E}{\partial \mathbf{x}}, \frac{\partial^2 E}{\partial \mathbf{x}^2}$$

```
aiqm1=mlatom.models.methods(method='AIQM1')
geomopt=mlatom.optimize_geometry(model=aiqm1, ...)
...
mlatom.namd.surface_hopping_md(model=model, ...)
```

Models

- HF, MP2, CC; ADC(2), CASSCF; ...
- B3LYP; ωB97x; ...
- PM6, GFN2-xTB; AM1, PM3, OMx; ...
- AIQM1; AIQM1@DFT; AIQM1@DFT*
- ANI-1x, ANI-2x; ANI-1ccx; ...
- ANI, PhysNet, DPMD; DeepPot-SE, MACE; ...
- KREG, sGDML; KRR-CM, GAP-SOAP; ...

L. Zhang, M. Martyka, ..., J. Jankowska, M. Barbatti, P. O. Dral. *JCTC*, **2024**, *20*, 5043–5057

MLatom's interfaces

Extras!

MLQD

A Package for Quantum Dissipative Dynamics with Machine Learning
by Arif Ullah, Anhui University

[MLQD: A. Ullah, P. O. Dral. *Comput. Phys. Commun.* **2024**, *294*, 108940]

Wechat
Follow
XACS account

Slack
Answering your
questions in real time

Quantum chemical programs:

Gaussian
ORCA
PySCF
Turbomole
COLUMBUS

Semi-empirical quantum chemical programs:

xTB
SPARROW
MNDO

Machine learning programs:

PhysNet
GAP
TORCHANI
SCOML
hyperopt

Dynamics and other atomistic simulation:

ASE
NX newtonx.org

Not everything is available on the cloud...

Python API

对于KREG模型，我们可以使用简单的网格搜索优化

```
model = ml.models.kreg(model_file=f'kreg.npz')

sub, val = molDB.split(number_of_splits=2, fract

model.hyperparameters['sigma'].minval = 2**-5 # 
model.optimize_hyperparameters(subtraining_molec
                               optimization_alg
                               hyperparameters=
                               training_kwargs=
                               prediction_kwarg
lmbd = model.hyperparameters['lambda'].value ; s
valloss = model.validation_loss
print('Optimized sigma:', sigma)
print('Optimized lambda:', lmbd)
print('Optimized validation loss:', valloss)
# Train the model with the optimized hyperparame
model.train(molecular_database=molDB, property_t
# Train the model with the optimized hyperparame
model.train(molecular_database=molDB, property_t
```

输出如下所示（它可能随子训练集和验证集的随机子

```
Optimized sigma: 0.10511205190671434
Optimized lambda: 2.910383045673381e-11
Optimized validation loss: 3.1550365181164988e-0
```

其他参数也是可用的，例如SciPy(`Nelder-Mead` , `BF` `krylov` , `trust-exact`)和hyperopt库(`TPE`)。

Cloud Computing

- Job Submitter
- Terminal
- File Manager
- Job Manager
- Jupyter Lab

Software

- Download

Learning

- Courses
- Workshops

Statistics

Total CPU time used

4756h 1m

Job Information

* Job Name 2024-05-30_0757

Job Location from_job_submitter

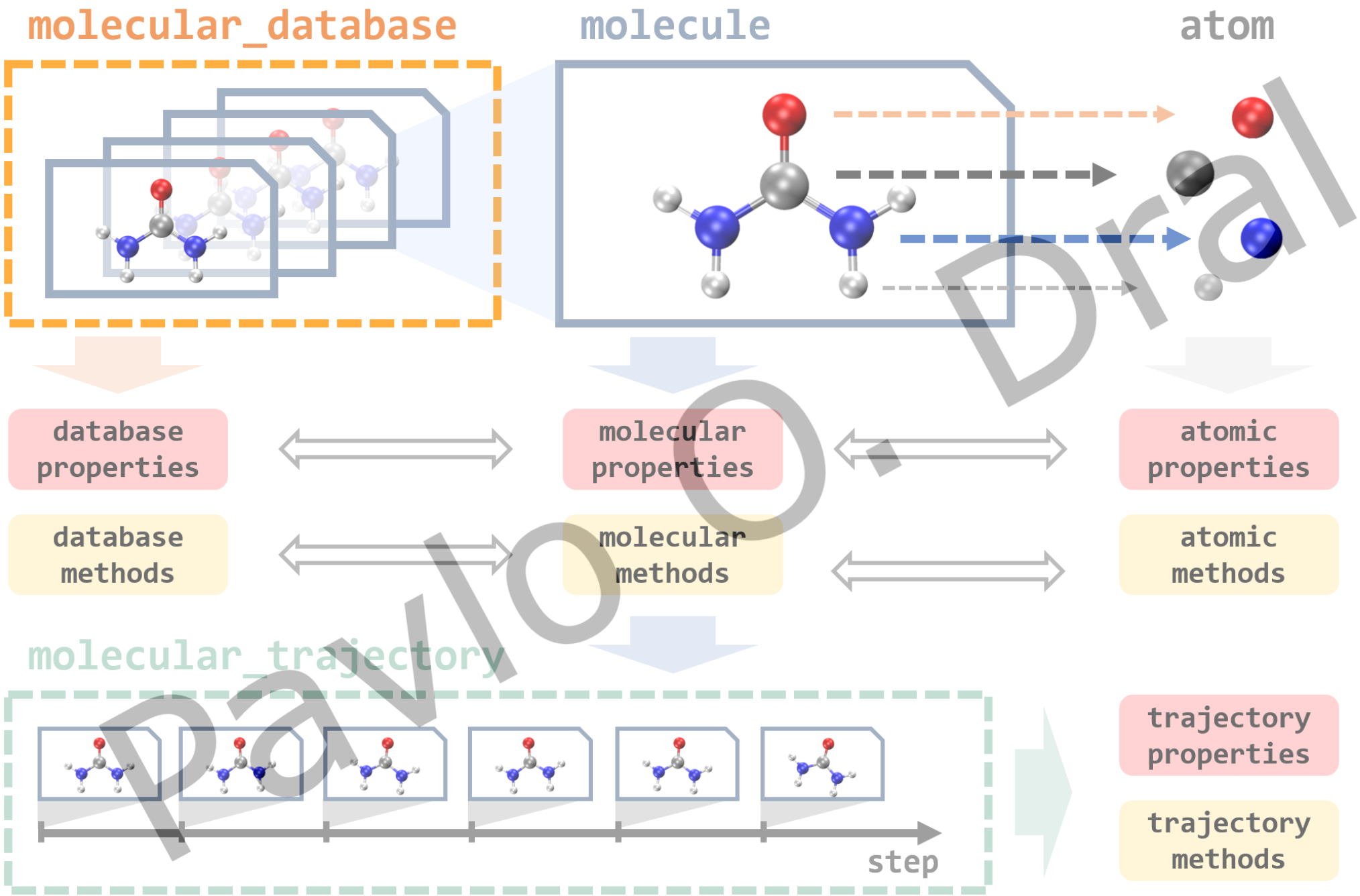
Job Type XACS (auto detect)

Gaussian Mlatom_d

Input File

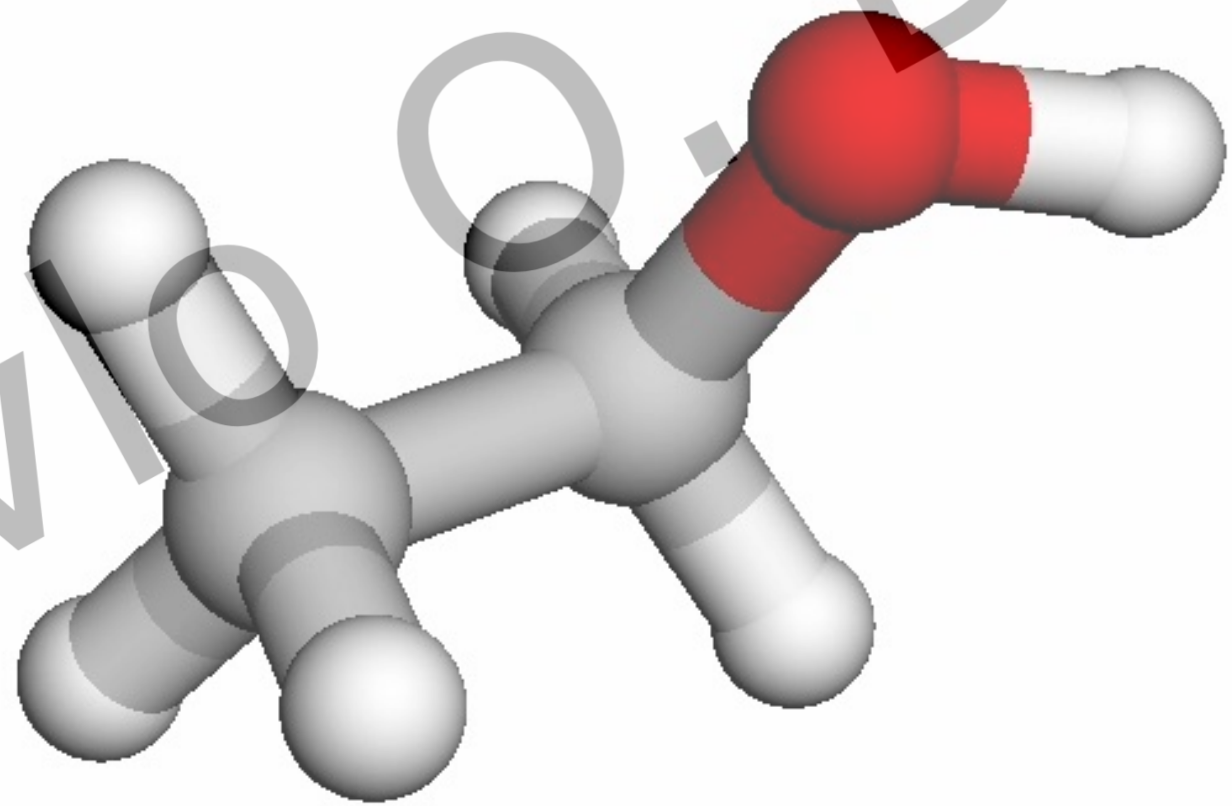
or edit XACS input file:

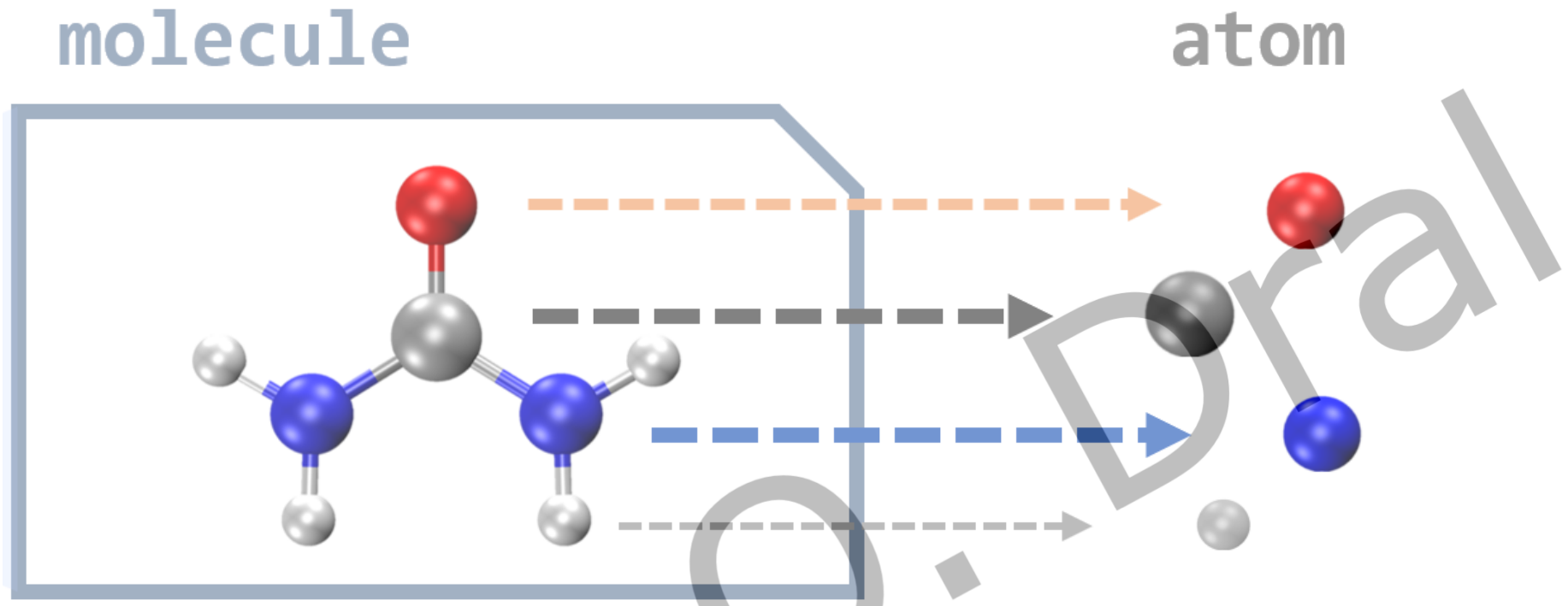
1



From MLatom 3 paper: *J. Chem. Theory Comput.* **2024,** *20*, 1193 (under CC-BY license)

```
molvibr.view(normal_mode=normal_mode)
```






```python
mol = ml.molecule()
mol.read_from_xyz_string('''2

H 0 0 0
H 0 0 0.74
''')
# or
mol.read_from_xyz_file('h2.xyz')
# or
import numpy as np
xyz = np.array([[0, 0, 0], [0, 0, 0.7]])
mol.read_from_numpy(coordinates=xyz, species=np.array([1, 1]))
```

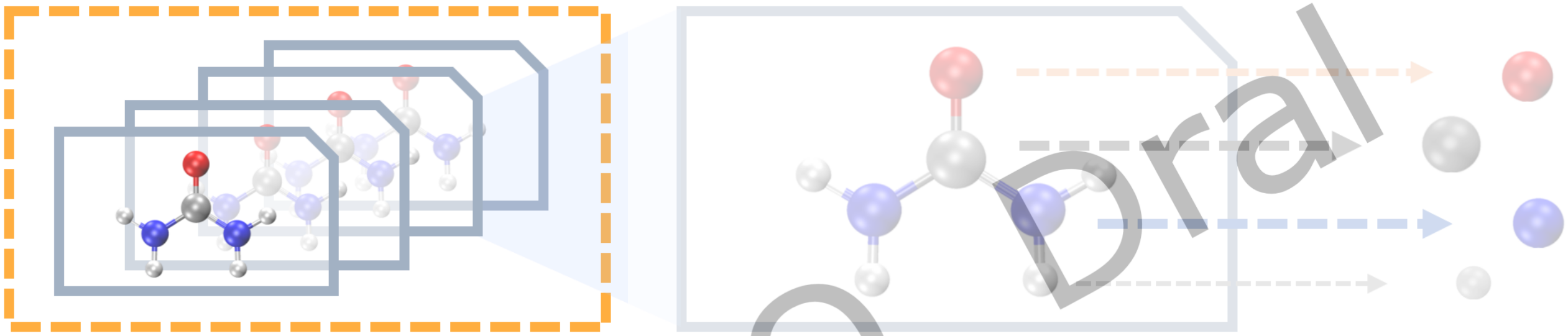
```python
mol.dump(filename='h2.json')
mol2.load(filename='h2.json')
```

```python
mol.charge = 1
mol.multiplicity = 2
```

molecular_database

molecule

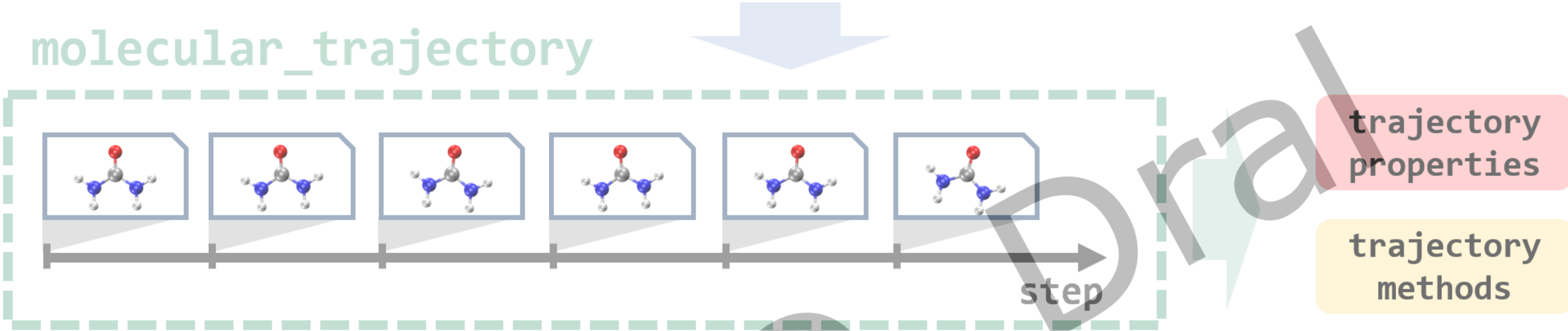
atom



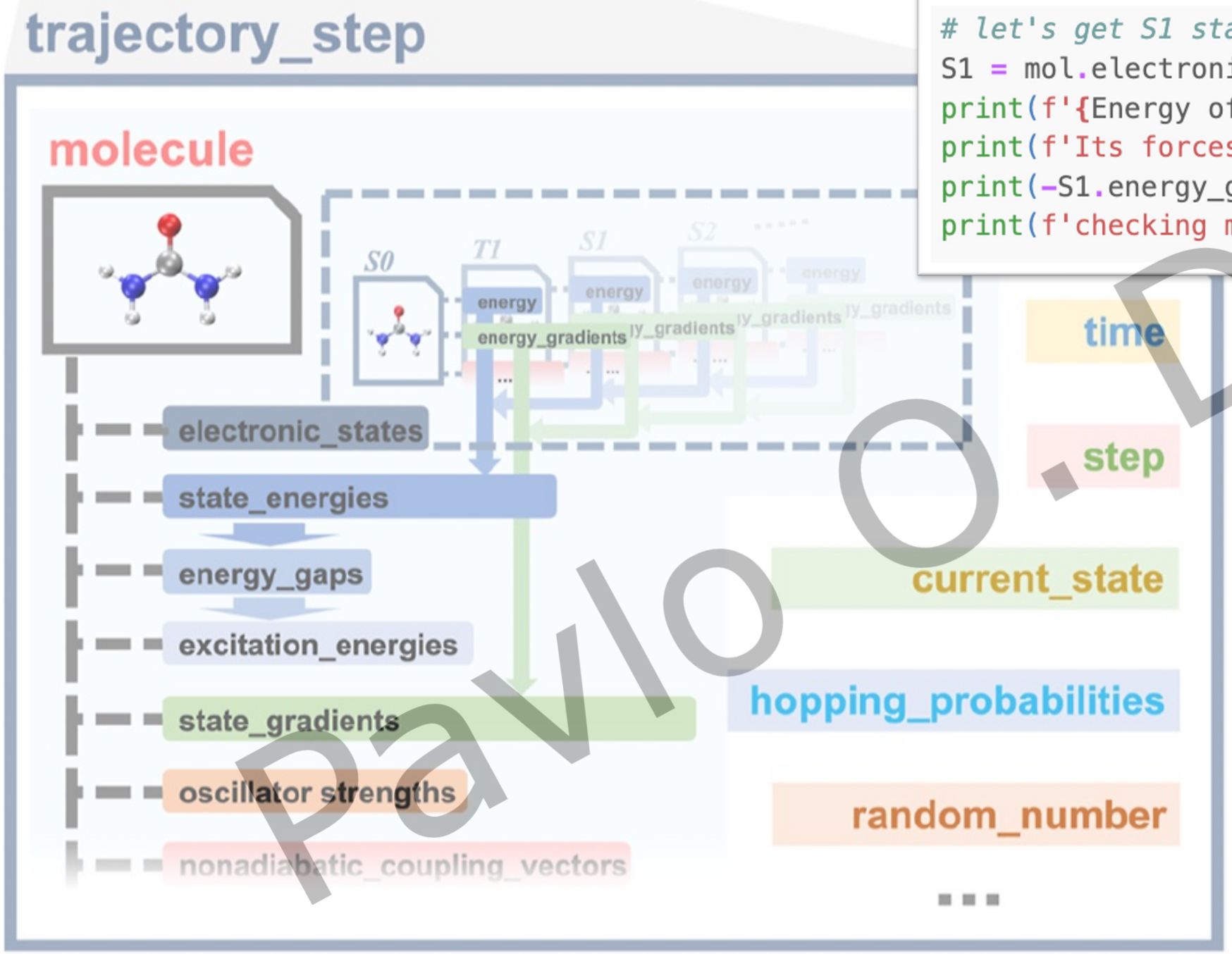
```python
# prepare H2 geometries with bond lengths ranging from 0.5 to 5.0 Å
xyz = np.zeros((451, 2, 3))
xyz[:, 1, 2] = np.arange(0.5, 5.01, 0.01)
z = np.ones((451, 2)).astype(int)
molDB = ml.molecular_database.from_numpy(coordinates=xyz, species=z)

molDB.dump(filename='h2.json')
molDB2 = ml.molecular_database.load(filename='h2.json')
```

```python
print(len(molDB))            # 451
print(len(molDB[:100:4]))    # 25
```

```python
# let's change the splitting to 9:1 instead of default 8:2
train, test = molDB.split(fraction_of_points_in_splits=[0.9, 0.1])
```




```
traj = ml.data.molecular_trajectory()
...
print(f'Info for MD trajectory step {traj.steps[5].step}'')
print(f'Molecule: {traj.steps[5].molecule}')
print(f'Time: {traj.steps[5].time}')
print(f'Kinetic energy: {traj.steps[5].molecule.kinetic_energy}')
```




```python
# let's get S1 state
S1 = mol.electronic_states[1]
print(f'{Energy of S1 state is {S1.energy}')
print(f'Its forces are')
print(-S1.energy_gradients)
print(f'checking multiplicity: {S1.multiplicity}')
```


See for more details MLatom's paper on nonadiabatic dynamics: *JCTC* **2024**, *20*, 5043–5057

Subscribe to updates and submit your papers to the AI/ML journals

MLatom.com

11:05 LTE 81

IOPscience

Ukraine: Click here to read IOP Publishing's statement

Machine Learning: Science and Technology

Home › Journals › Machine Learning: Science and Technology › Editorial board

Follow:

- Homepage
- Author guidelines
- About Machine Learning: Science and Technology
- Editorial board
- Track my article

AA shingsupport.iopscience.iop.org