JK

WORKSHOP

Part 2

JKML

- created 2021 and still evolving
- the script is working but it is not much efficient and very hard to read

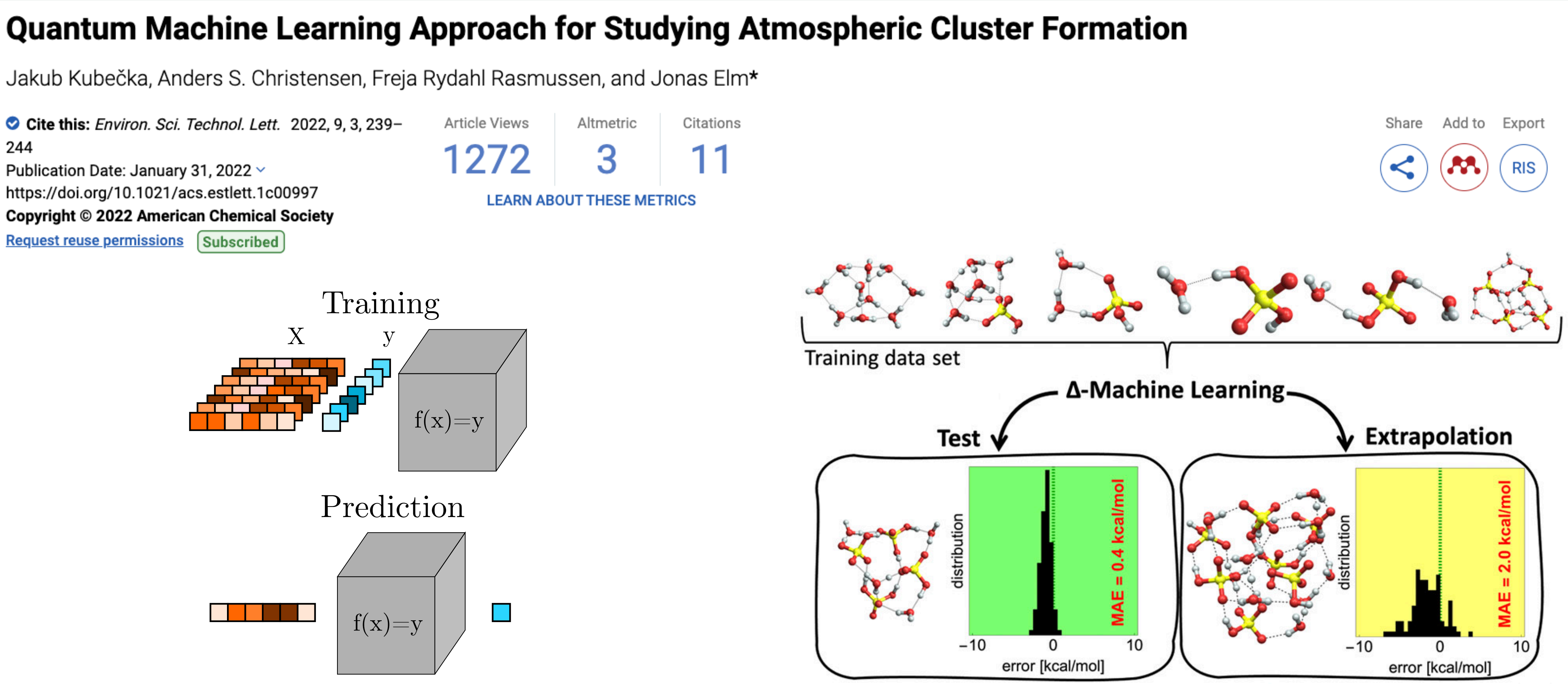
  … I have sometimes issues to understand it myself. <promise to rewrite it soon>
- Uses Python packages such as NumPy, (SciPy), Pandas, DScribe,

  PyTorch/Pythorch-Lightning, QML, SchNetPack, (hydra-core, sclearn …)
- as input uses the JKQC pickled files
- submittable to cluster via SLURM

  on this lecture, use **-loc** for teaching purposes //run on login computer

  THIS IS NOT ADVISE FOR YOUR FUTURE ACTIONS

SA-W 1. study case



Quantum Machine Learning Approach for Studying Atmospheric Cluster Formation

Jakub Kubečka, Anders S. Christensen, Freja Rydahl Rasmussen, and Jonas Elm*

Cite this: *Environ. Sci. Technol. Lett.* 2022, 9, 3, 239–244
Publication Date: January 31, 2022
https://doi.org/10.1021/acs.estlett.1c00997

Request reuse permissions Subscribed

Article Views	Altmetric	Citations
1272	3	11

LEARN ABOUT THESE METRICS

```
LECTURE_2/TASK_01/2sa2w_DFT.pkl
LECTURE_2/TASK_01/prepare.sh
```

direct-ML [KRR] for 2sa2am conformers

TODO: How many structures are in the 2sa2w_DFT.pkl file?

TODO: Understand prepare.sh file and then run it.

TODO: Study the JKML help.(Use: JKML -help)

Prepare the model by training on 2sa2w_DFT_train50.pkl:

```
JKML -loc -qml -train 2sa2w_DFT_train50.pkl
```

Use model.pkl to predict the 2sa2w_STR_test13.pkl energies:

```
JKML -loc -qml -trained model.pkl -eval 2sa2w_STR_test13.pkl
```

Try training and testing at one step:

```
JKML -loc -qml -train 2sa2w_DFT_train50.pkl -eval 2sa2w_DFT_test13.pkl
```

Closer look at KRR

Kernel (“similarity to training structures”) multiplied by fitted/regression coeffitients gives the modelled energy:

$$\vec{E} = \mathbf{K} \cdot \vec{\alpha}$$

The theory is based on the possibility to write energy as sum of atomic energies:

$$E(c) = \sum_{I \epsilon c} E_{\mathrm{local}}(\mathbf{q}_I) = \sum_{I \epsilon c} \sum_{c'} \sum_{J \epsilon c'} \mathcal{K}(\mathbf{q}_J, \mathbf{q}_I) \alpha_{c'}$$

Kernel (“similarity matrix”) definition:

$$\mathcal{K}(\mathbf{q}_J, \mathbf{q}_I) = \delta_{Z_J Z_I} \exp\left(- \frac{\|\mathbf{q}_J - \mathbf{q}_I\|_2^2}{2\sigma^2} \right)$$

Regression coef. training:

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

```
LECTURE_2/TASK_01/2sa2w_DFT_train50.pkl
LECTURE_2/TASK_01/2sa2w_DFT_test13.pkl
```

Hyperparameter optimization

The default hyperparameters do not have to be optimal for your study case!

TODO: Check JKML help and figure out how to fiddle with KRR hyperparameters (use: -help to figure out that you need –help_???)

TODO: What extra arguments do you need? Run them!

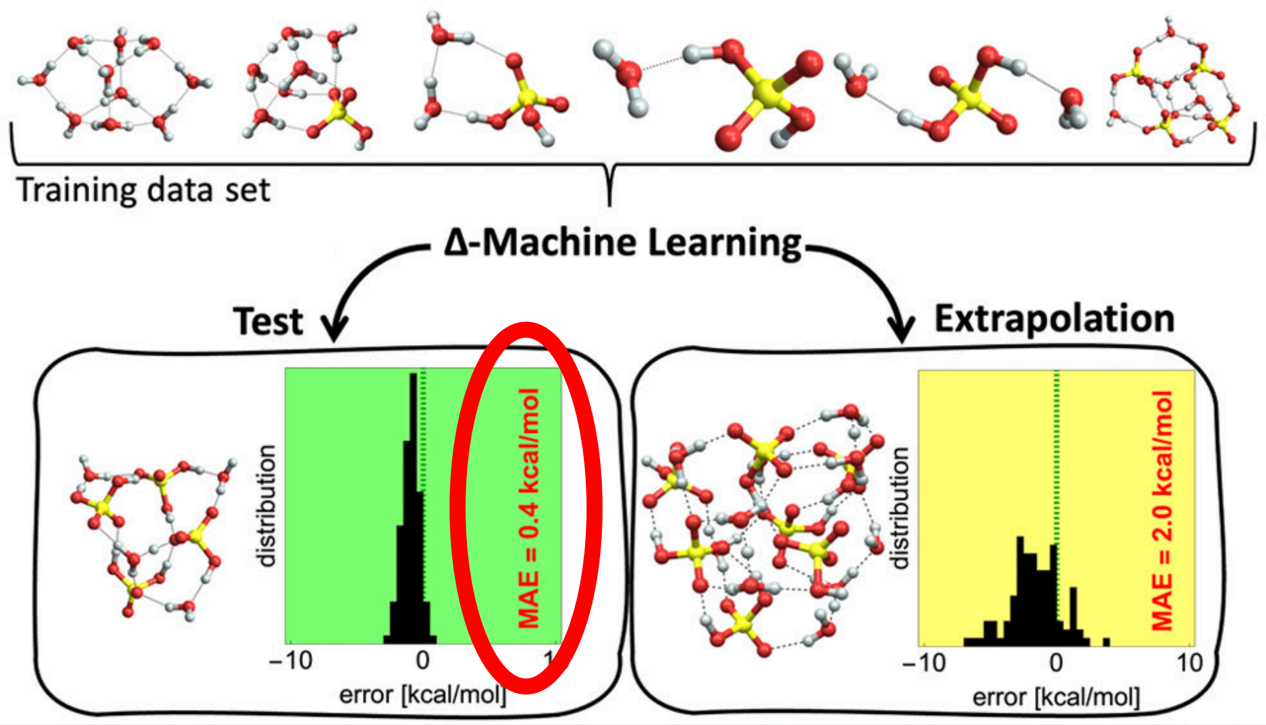
```
JKML -loc -qml -train 2sa2w_DFT_train50.pkl -test 2sa2w_DFT_test13.pkl <WHAT>
```

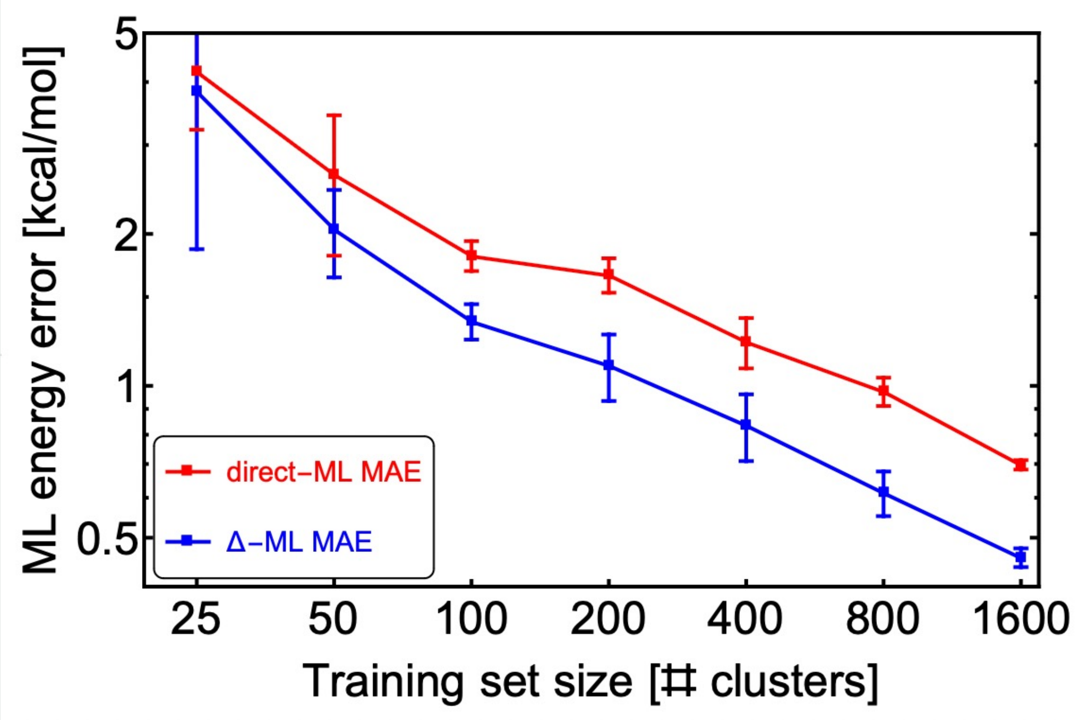
COMPETITION: Shout out the lowest MAE you were able to reach.

Note: If you accidentally submit a job, just kill it, e.g.: scancel 7846632778

```
LECTURE_2/TASK_02/sa-w_DFT.pkl
LECTURE_2/TASK_02/prepare.sh
```

SA-W system





TODO: Examine TASK_02 folder, perform the training and testing.

```
LECTURE_2/TASK_02/train.pkl
LECTURE_2/TASK_02/test.sh
LECTURE_2/TASK_02/mons.sh
```

Relative properties

To reduce the deviation of predicted properties, you often use properties relative to number of atoms/molecules, e.g. atomization energies or in out case binding energies:

$$\Delta E_{\mathrm{bind}} = E_{\mathrm{cluster}} - \sum_{i} E_{\mathrm{monomer}}(i)$$

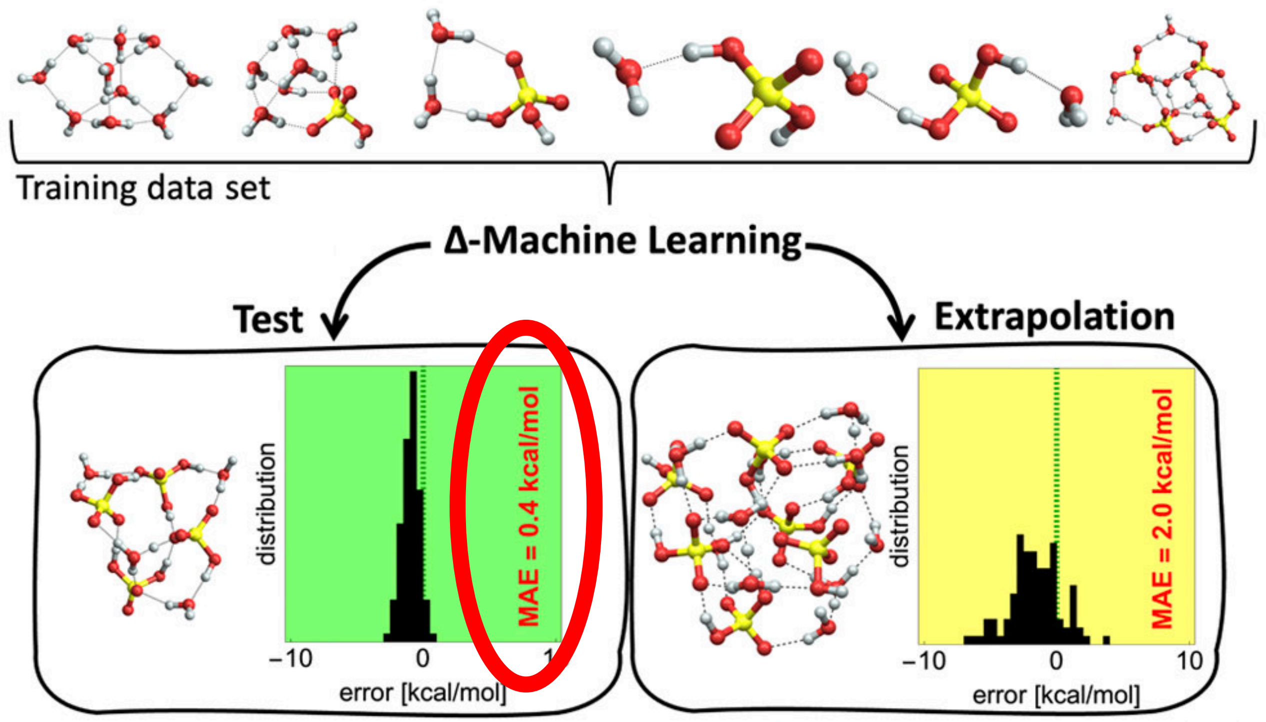
TODO: In this folder is a hidden file. Copy it to mons.pkl.

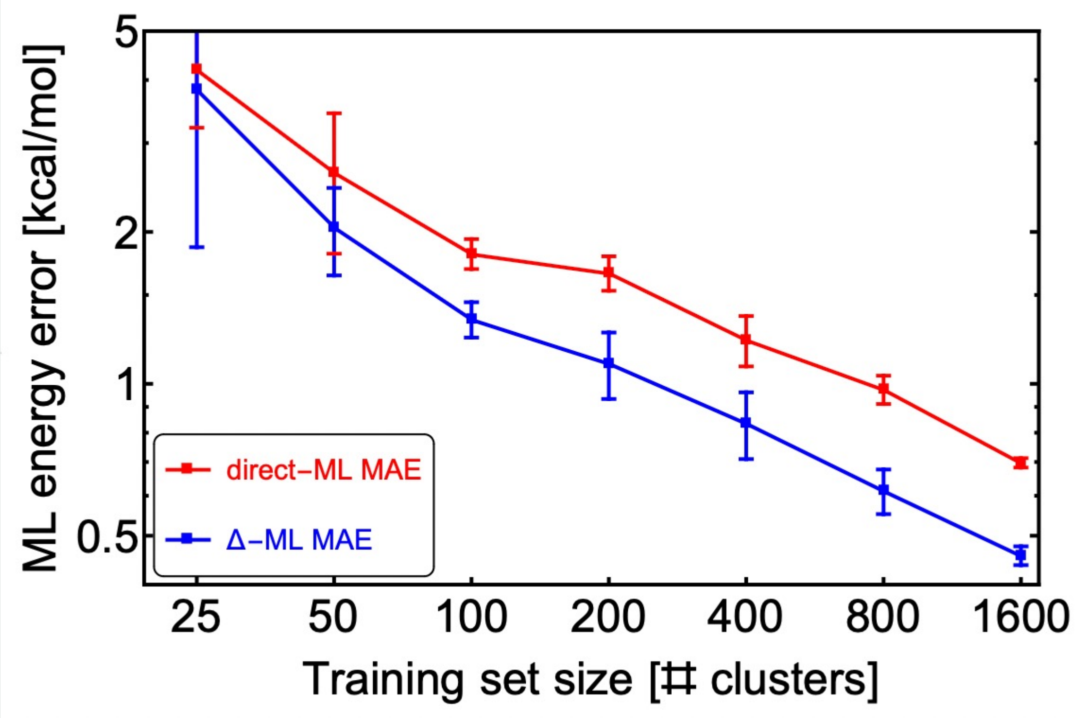
TODO: Train and test on binding energies. What MAE do you get?

```
JKML -loc -qml -train train.pkl -test test.pkl -monomers mons.pkl
```

```
LECTURE_2/TASK_02/sa-w_DFT.pkl
LECTURE_2/TASK_02/prepare.sh
```

SA-W system



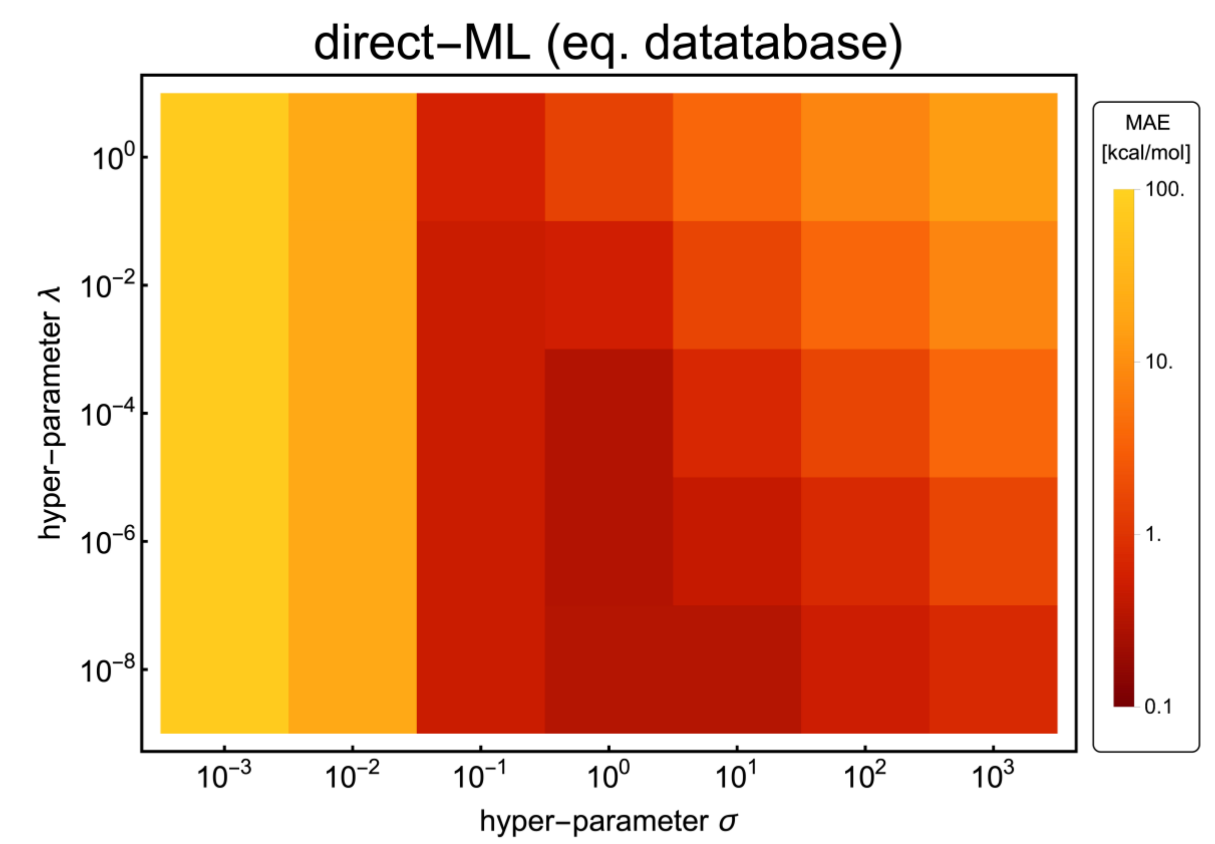


LECTURE_2/TASK_01/2sa2w_DFT_train50.pkl
LECTURE_2/TASK_01/2sa2w_DFT_test13.pkl

Training on binding energies

These are often the common cases when trained on el. binding energies:

(default JKML values)



```
LECTURE_2/TASK_02/train.pkl
LECTURE_2/TASK_02/test.sh
LECTURE_2/TASK_02/mons.sh
```

delta-ML [KRR] for 2sa2am conformers

We can learn some basic chemistry at a cheap/fast method and then use ML to only train on the difference:

$$\Delta\Delta E_{\mathrm{bind}} = \Delta E_{\mathrm{bind}}^{\mathrm{DFT}} - \Delta E_{\mathrm{bind}}^{\mathrm{PM7}}$$

TODO: What is faster, direct-ML or delta-ML?

TODO: Check prepare.sh (note the sorting), run it, compare:

```
JKML -loc -qml -train trainHIGH.pkl              -test testHIGH.pkl             <WHAT>
```

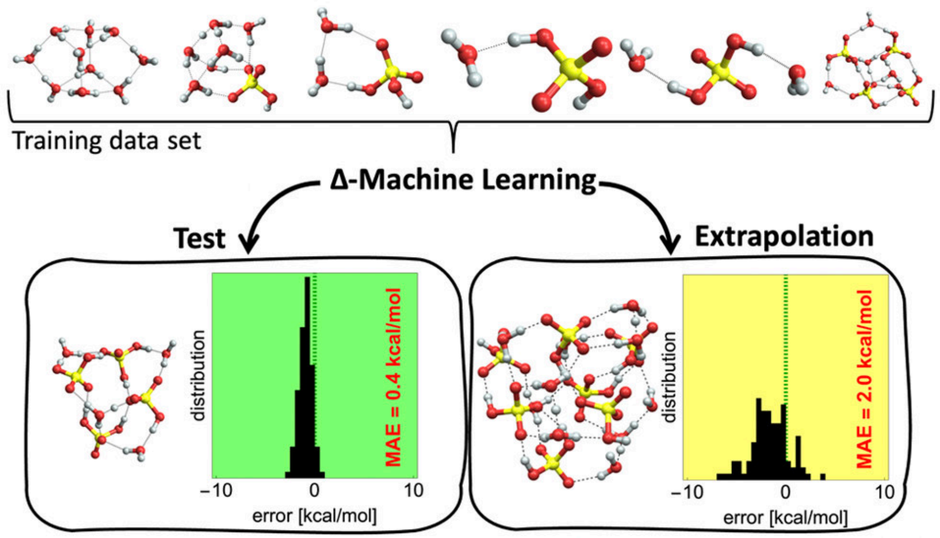
VS

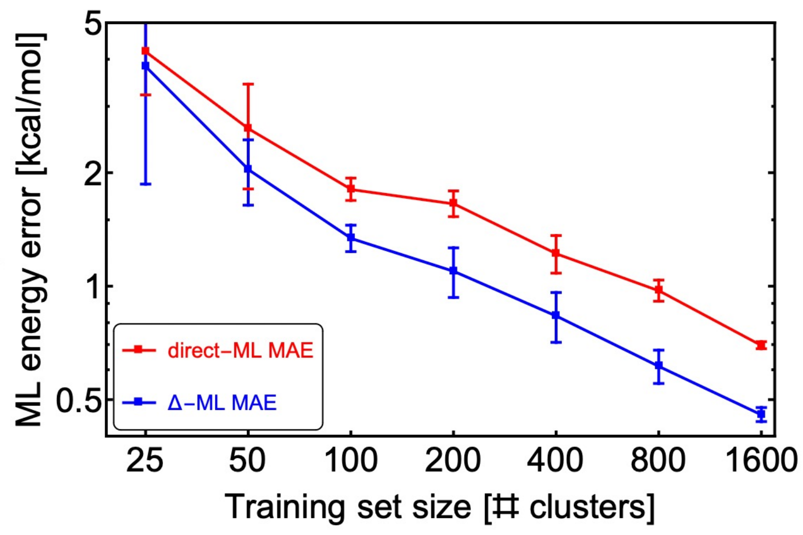
```
JKML -loc -qml -train trainHIGH.pkl trainLOW.pkl -test testHIGH.pkl testLOW.pkl <WHAT>
```

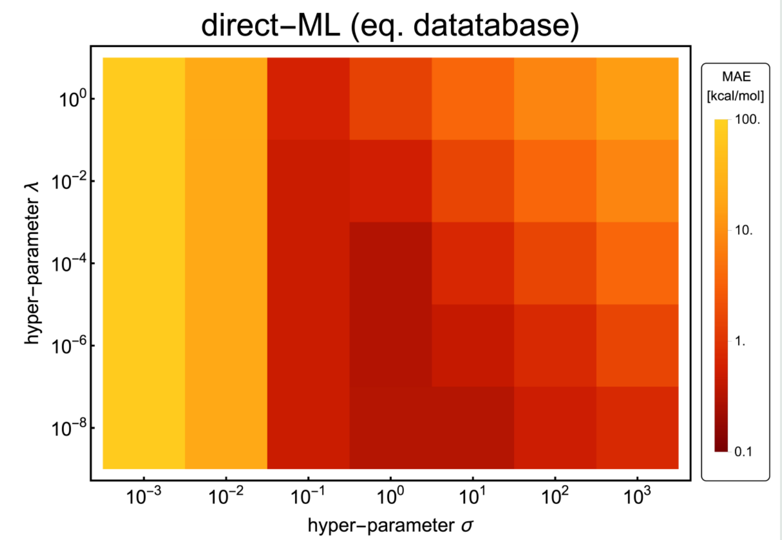
TODO: Examine/Think hyperparameters and accuracy. Why?!

LECTURE_2/TASK_04/*.pkl

SA-W system: delta-ML (combined knowledge from TASKS_01-3)







$$\Delta E_{\mathrm{bind}} = E_{\mathrm{cluster}} - \sum_{i} E_{\mathrm{monomer}}(i)$$

$$\Delta\Delta E_{\mathrm{bind}} = \Delta E_{\mathrm{bind}}^{\mathrm{DFT}} - \Delta E_{\mathrm{bind}}^{\mathrm{PM7}}$$

TODO: Can you get similar MAEs as in the figure?! Why yes/not?

```
LECTURE_2/TASK_05/testDFT.pkl
LECTURE_2/TASK_05/testXTB.pkl
```

Bit more practise

TODO: Use the model.pkl trained in TASK_04 to predict energies of testDFT.pkl

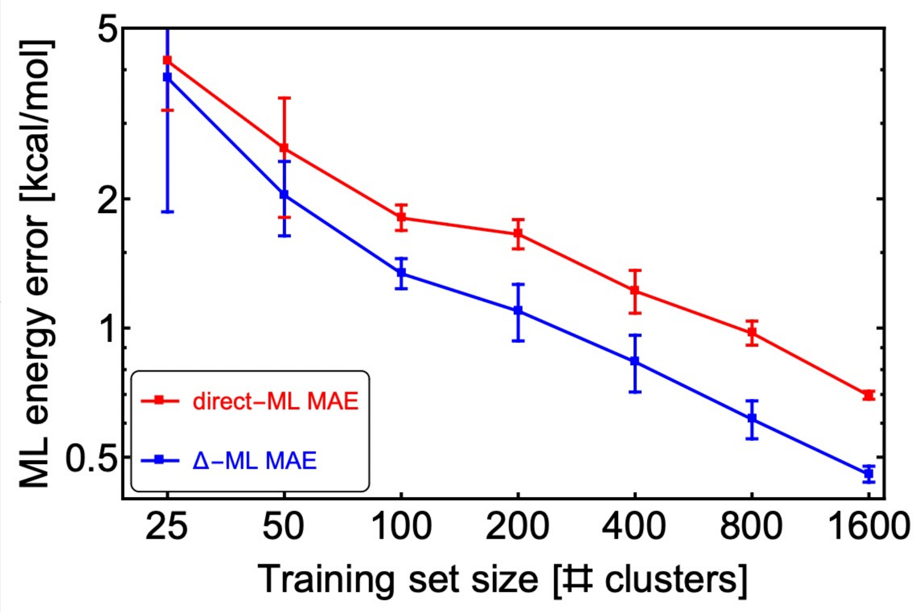
TODO: Can you get similar MAEs as in the figure?! Why yes/not?

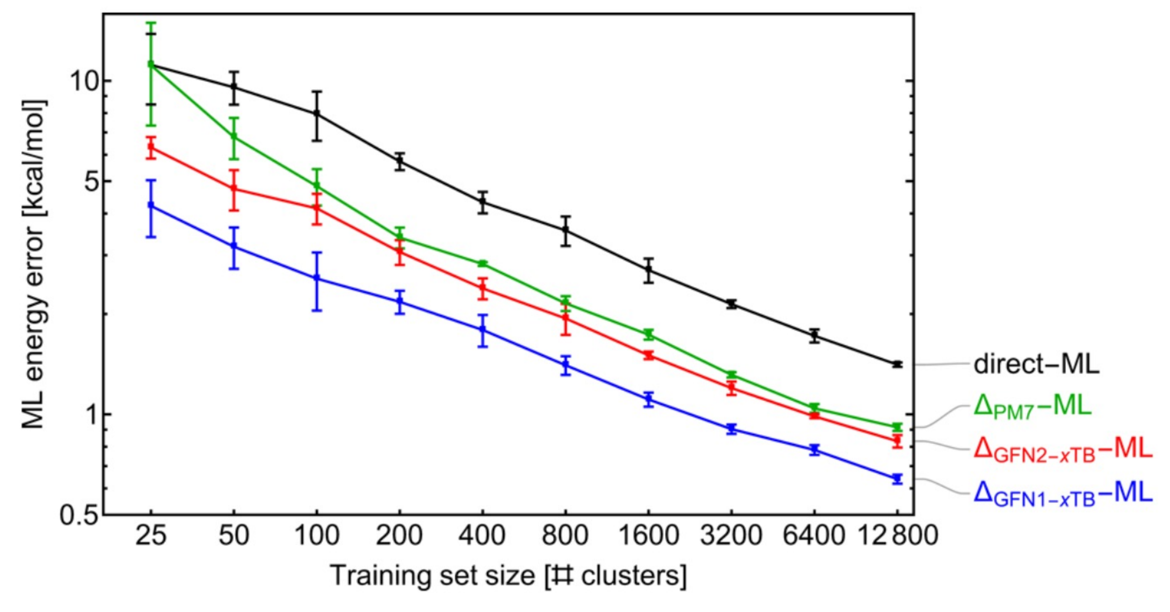
LECTURE_2/TASK_06/run.sh

Even more practise

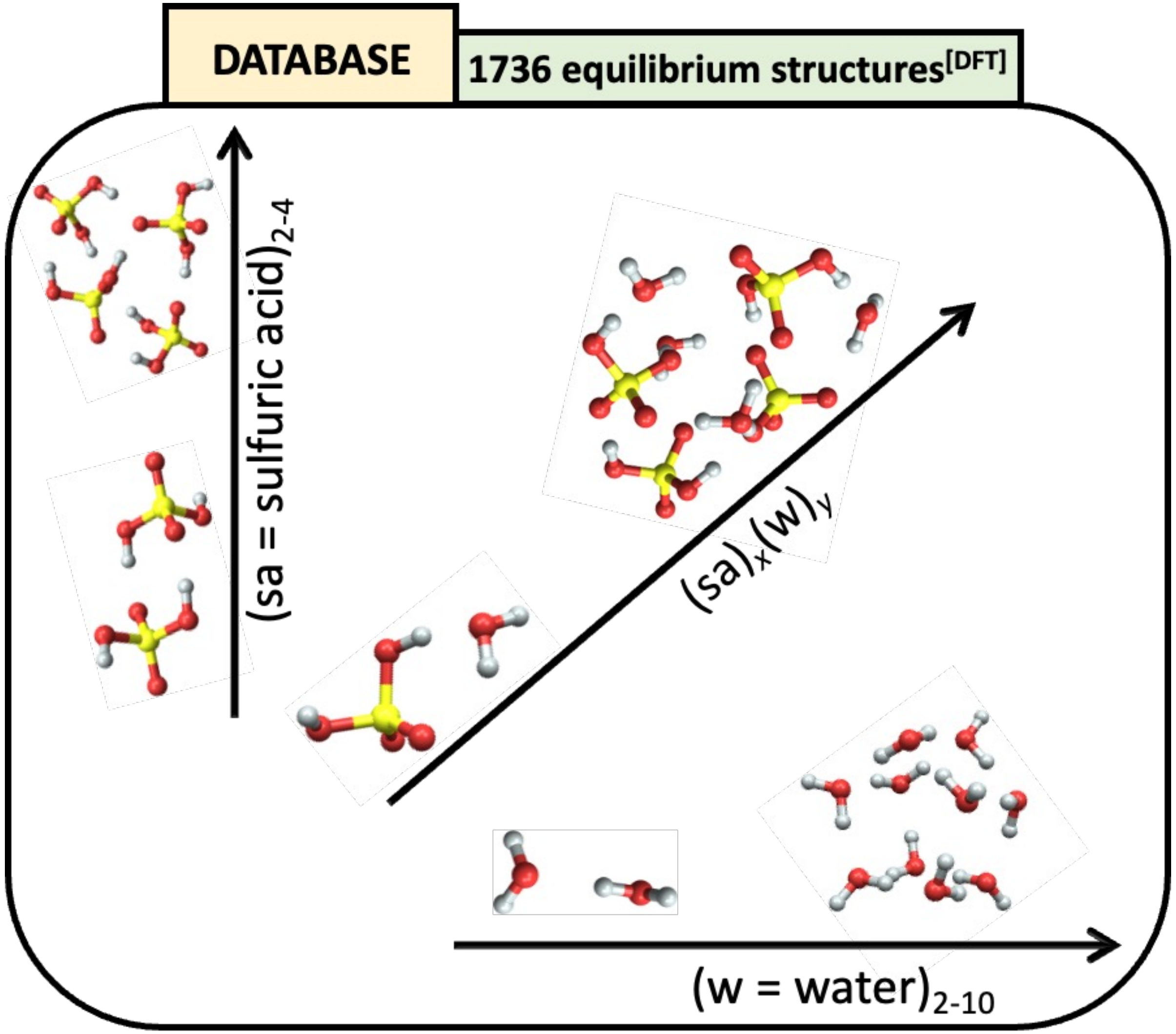
TODO: check TASK_06 folder, check file run.sh, run it. WHY?!

Pay attention to your methods, training/test/mons databases, program versions, whether you interpolate or extrapolate etc.





DATABASE

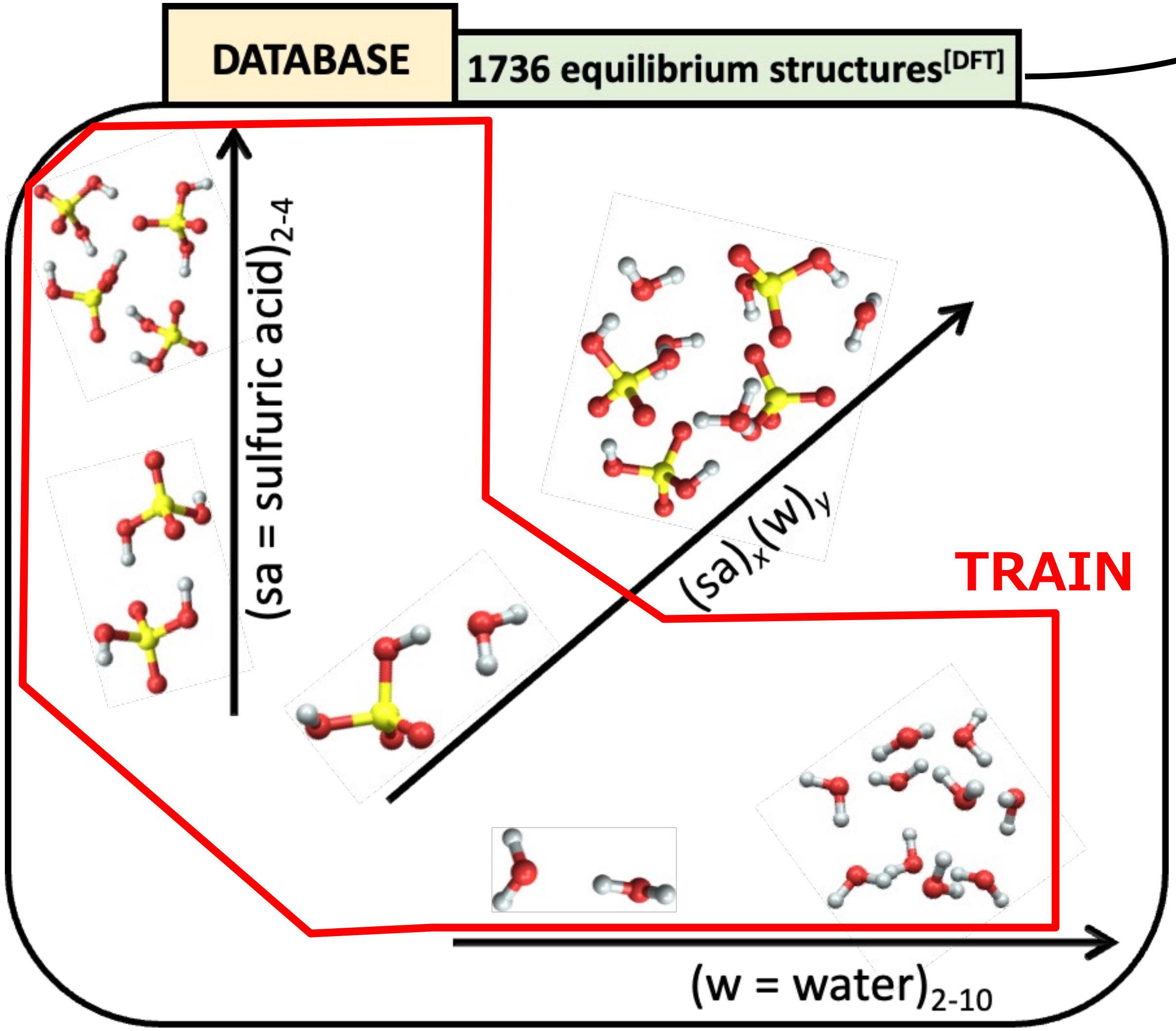


DATABASE
1736 equilibrium structures[DFT]
$(sa = sulfuric\ acid)_{2-4}$
$(sa)_x(w)_y$
$(w = water)_{2-10}$

OVERVIEW oF MACHINE LEARNING STEPS

1) DATABASE OF STRUCTURES (**TRAINING SET: x**)

DATABASE


DATABASE
1736 equilibrium structures[DFT]
$(sa = \text{sulfuric acid})_{2\text{-}4}$
$(sa)_x(w)_y$
TRAIN
$(w = \text{water})_{2\text{-}10}$

Train set = no ~~$(sa)_4(w)_5$~~ ⇒ 1684 strs

OVERVIEW oF MACHINE LEARNING STEPS

1) DATABASE OF STRUCTURES (**TRAIN SET: x**)

2) CALCULATE **FCHL** REPRESENTATIONS

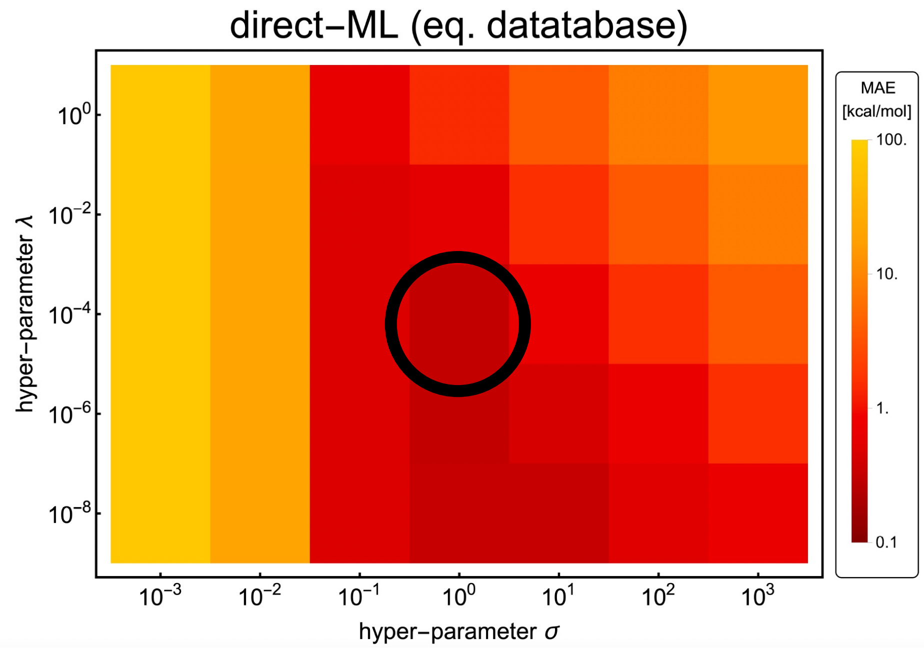
3) OPTIMIZE REGRESSION COEFFTIENTS:

$$\alpha = (\underset{\text{kernel matrix}}{\mathbf{K}}{}^{\overset{\text{kernel width}}{\sigma}}_{x,x} - \underset{\text{regularizer}}{\lambda} \cdot \underset{\text{unit matrix}}{\mathbf{I}})\overset{\text{energies of training structures}}{E_x}$$

4) **TEST SET: x'**

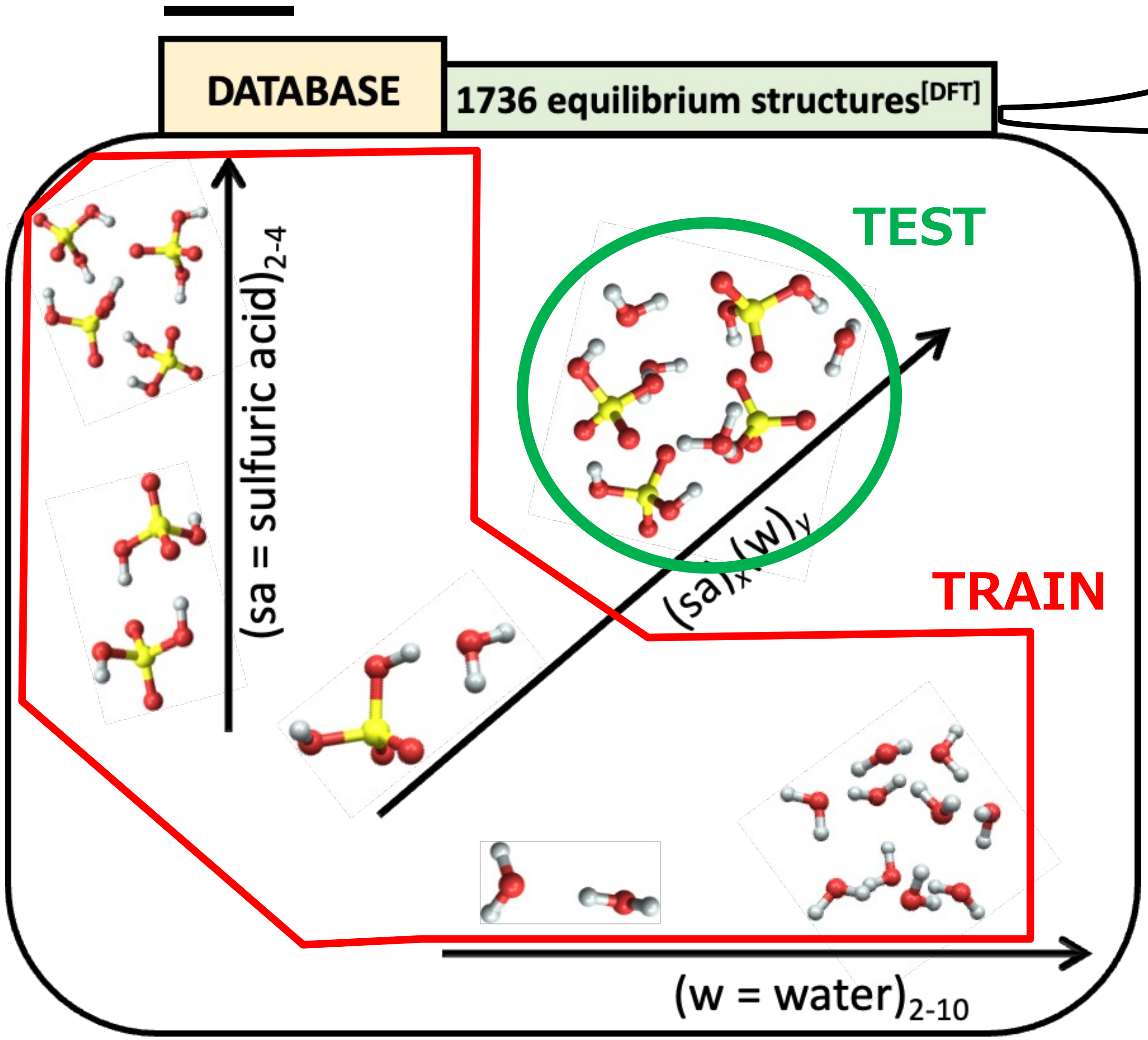
SIDE QUEST:

hyperparameter optimization



direct–ML (eq. datatabase)

DATABASE


DATABASE
1736 equilibrium structures[DFT]
TEST
TRAIN
(sa = sulfuric acid)$_{2\text{-}4}$
(sa)$_x$·(w)$_y$
(w = water)$_{2\text{-}10}$

Train set = no ~~$(sa)_4(w)_5$~~ ⇒ 1684 strs

Test set = only $(sa)_4(w)_5$ ⇒ 52 strs

OVERVIEW oF MACHINE LEARNING STEPS

1) DATABASE OF STRUCTURES (**TRAIN SET: x**)

2) CALCULATE **FCHL** REPRESENTATIONS

3) OPTIMIZE REGRESSION COEFFTIENTS:

$$\alpha = (\underset{\text{kernel matrix}}{\mathbf{K}}{}_{x,x}^{\overset{\text{kernel width}}{\sigma}} - \underset{\text{regularizer}}{\lambda} \cdot \underset{\text{unit matrix}}{\mathbf{I}}) \overset{\text{energies of training structures}}{E_x}$$

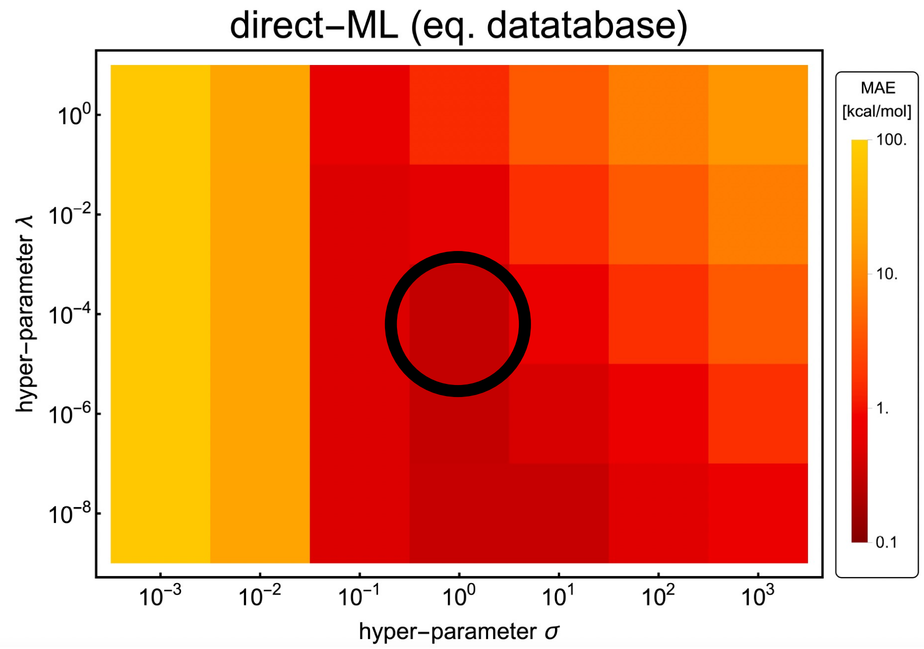
4) **TEST SET: x'**

5) EVALUATE:

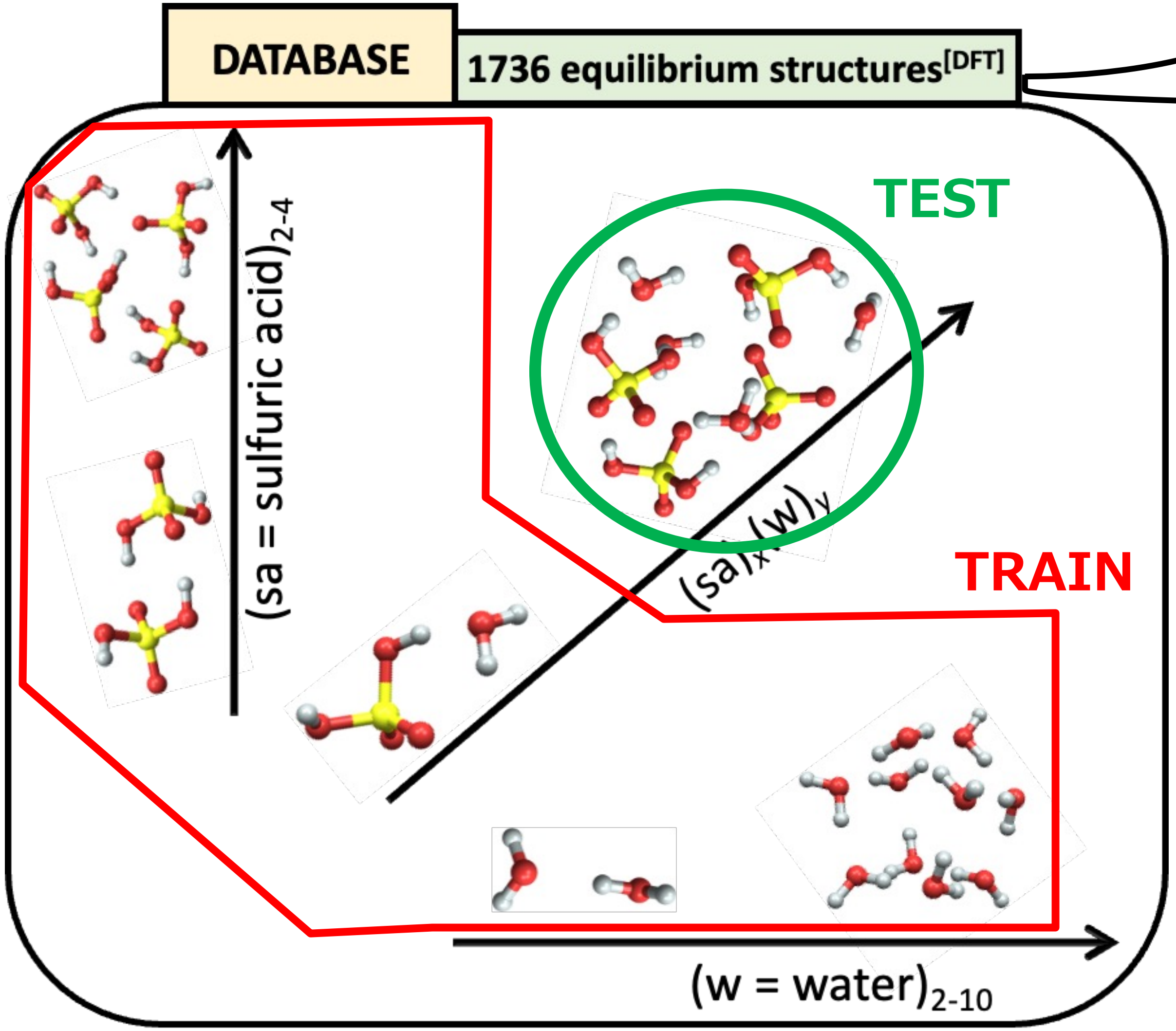
$$E_{x'} = \mathbf{K}_{x',x} \cdot \alpha$$

SIDE QUEST:

hyperparameter optimization

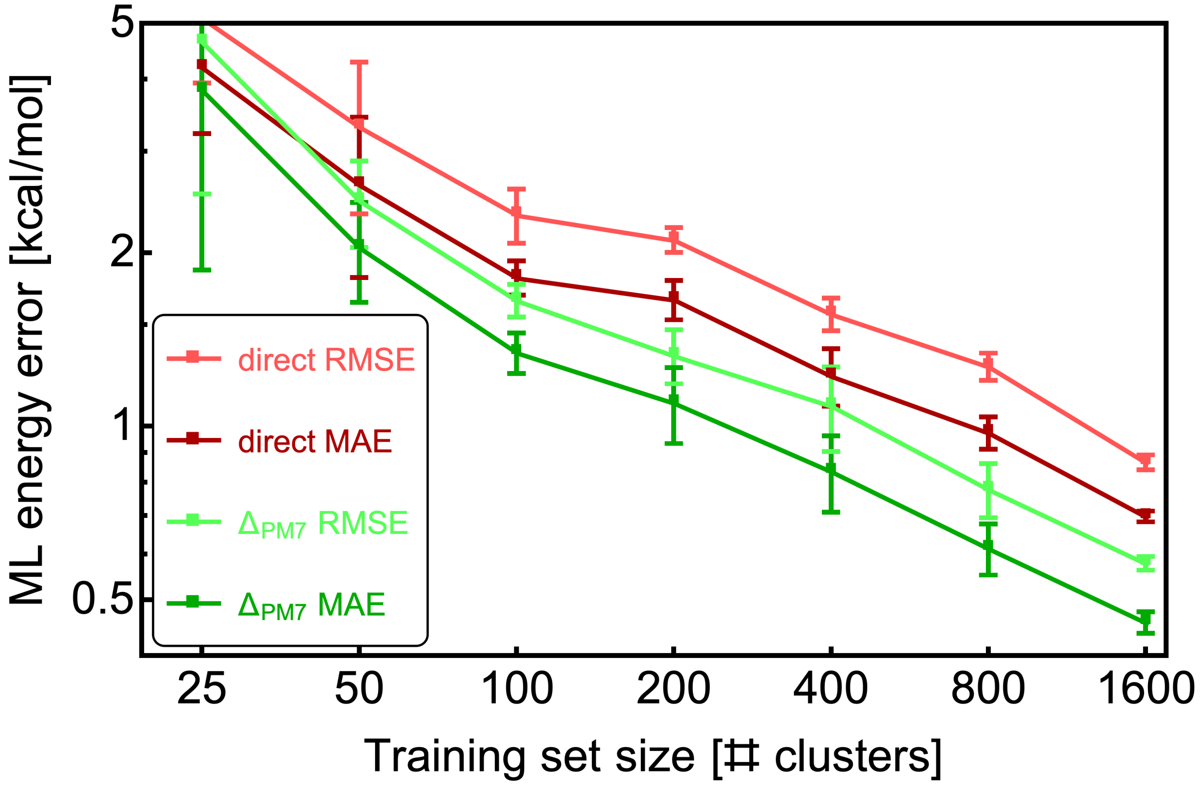


RESULTS


DATABASE
1736 equilibrium structures[DFT]
TEST
TRAIN
$(sa = \text{sulfuric acid})_{2\text{-}4}$
$(sa)_x(w)_y$
$(w = \text{water})_{2\text{-}10}$

Train set = no ~~$(sa)_4(w)_5$~~ ⇒ 1684 strs

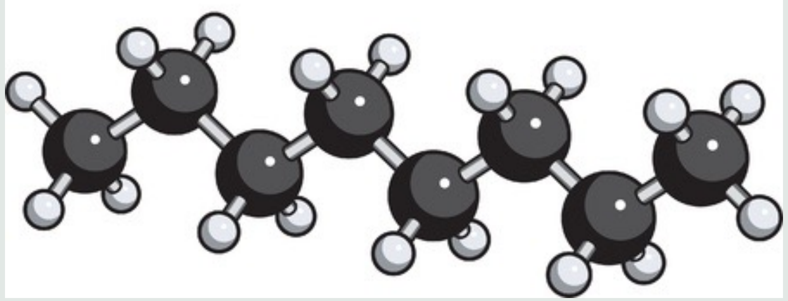
Test set = only $(sa)_4(w)_5$ ⇒ 52 strs

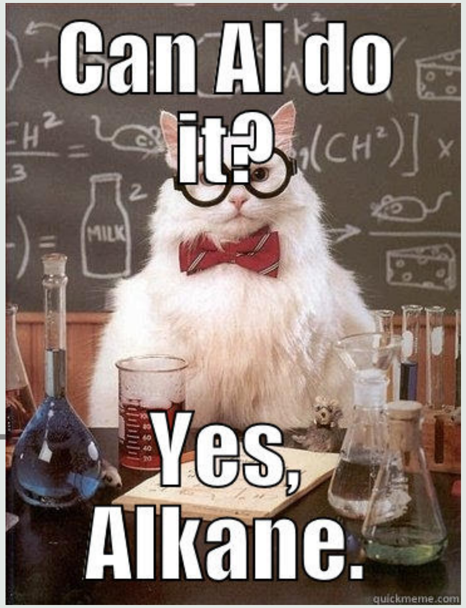


ABOUT CLUSTERS

LECTURE_2/TASK_07/collection*me.pkl

Working with C_nH_{2n+2}





TODO: What alkanes are inside these pickles?

TODO: Is the naming appropriate? Does e.g. following command make sense?

```
JKQC *.pkl -ct -el -formation -unit
```

Let us prepare the test and training database witch artificial atomization energies:

```
JKQC collection8me.pkl     -atomize -out test.pkl
JKQC collection[1-7]me.pkl -atomize -out train.pkl
```

TODO: Examine atoms.pkl. Train and test with QML! (use: -atoms atoms.pkl)

Alkanes are boring

LECTURE_2/TASK_08/database.pkl

QM9 database

Several database can be found online: QM9, GeckoQ, ACDB ···

We picked several $C_7O_2H_{10}$ molecules from QM9 database and extended it with few more conformers of those molecules (using CREST).

TODO: How many entries are in database.pkl?

TODO: Preshuffle and pick e.g. 100 for training and different 100 for testing?

TODO: Train and test with QML.

TODO: Is the result good?

TODO: Visualize the tested molecules?

LECTURE_2/TASK_09/*.pkl

Training on Energies+Forces

TODO: Check that forces are indeed present.(use: -info)

TODO: Select 20 random 4sa5w clusters for testing.

TODO: Select (25,50,100,200) random clusters for training.

TODO: Train NN.

TODO: Visualize the training/validation error. (You can use JKplot)

TODO: Visualize the correlation of predicted and true energies. (You can use JKplot)

TODO: Take one structure and optimize it with ML.

TODO: Force Jakub to write a proper manual.

TODO: Force Jakub to rewrite JKML properly.

LECTURE_2/TASK_09/*.pkl

Smart training

TODO: Partly train the model on 25 structures.

TODO: Test on the full set.

TODO: Partly re-train the model on 25 + 25 worst structures.

TODO: Test on the full set.

...