Linear Programming for Unsupervised Training Set Selection in Molecular Machine Learning

Matthieu Haeberle,[1,2] Puck van Gerwen,[1,3] Ruben Laplaza,[1,3] Jan Weinreich,[1,3] Friedrich Eisenbrand[2] and Clémence Corminboeuf[*,1,3]

[1]Laboratory for Computational Molecular Design, Institute of Chemical Sciences and Engineering, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland
[2]Chair of Discrete Optimisation, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland
[3]National Center for Competence in Research-Catalysis (NCCR-Catalysis), Zurich, Switzerland

E-mail: clemence.corminboeuf@epfl.ch

13 April 2024

Abstract. Computational chemistry increasingly relies on Machine Learning (ML) to estimate molecular properties, but the accuracy of these models is contingent on the availability of high-quality training data. To date, numerous labelled chemical datasets have been made available open-source. Cost-effective ML models can be built by selecting a relevant subset of these labelled datasets, for example by using CUR decomposition, Farthest Point Sampling, and Similarity Machine Learning. However, few works have focused on selecting an optimal subset of *small* molecules to train an ML model to infer properties of *large* molecules. To address this gap, we introduce a Linear Programming (LP) algorithm that efficiently selects small reference environments that are relevant to a larger target molecule. It is shown that the LP-approach outperforms existing training set selection models for the extrapolation task: optimal training sets are selected from QM7 and predictions are made on larger molecules from QM9 or drug-like databases. We expect that this method can be used to alleviate the high computational burden of generating datasets and training ML models for large-molecule property prediction.

1. Introduction

Machine Learning (ML) has become a central tool in the computational chemist's toolbox, where molecular properties can be rapidly estimated using statistical models rather than direct computation.[1–5] Accurate ML models must be trained on a sufficient quantity of relevant training data. To address this data requirement, there has been a surge in the publication of datasets for computational chemistry.[6–19] Despite the growing number of open-source datasets, in many out-of-sample scenarios, the ideal training set is not available. This is because many datasets are out-of-scope or contain largely redundant

examples. A pragmatic solution is *training set selection*: identifying a subset of a larger database, which minimises the number of datapoints while maintaining or improving the accuracy of the model trained on the full database.[20–26] Such a subset is called a *coreset* in the field of deep learning,[27,28] where the emphasis is on reducing the cost of model training. Quantum-chemical datasets are rarely too large for efficiency concerns in ML training. Instead, here the purpose of training set selection is rather to find a training set that has the highest accuracy when the model is evaluated out of sample.

To this end, several training set selection strategies have been developed, falling broadly into supervised[20–23,29] and unsupervised[24,25,30] approaches, where the former makes use of both the representations and labels in the selection process and the latter of the representations only.

Much larger datasets[31–39] can be exploited in the unsupervised setting, but the chosen subset of points must be labelled before performing supervised learning. Unsupervised approaches include (i) CUR matrix approximations[30] and related approaches based on the D-optimality criterion;[40,41] (ii) Farthest Point Sampling (FPS), which selects molecules based on the diversity (*i.e.*, maximum distance) of their representations; and (iii) Similarity Machine Learning (SML)[25] which instead selects the k closest points to the target, *i.e.*, with a minimal distance between their representations. The Atoms-in-MOlecules (AMONs)[26] approach does not select samples from a reference pool, but rather constructs a dedicated training set by performing a substructure search on the target molecule. Being based only on the structure of the target, this also constitutes an unsupervised approach. Nevertheless, it requires the construction of a new database and therefore is distinct from the other approaches discussed. Cersonsky *et al.*[21] introduced a supervised variation of the CUR and FPS using a method inspired by Principal Covariates (PCov) regression.[42] Other supervised approaches include Orthogonal Matching Pursuit (OMP),[22,29] to greedily select environments with the highest inner product with the target property; the Genetic Algorithm (GA)-based approach proposed by Browning *et al*[20] which optimizes the training set *vs.* the population of possible training sets; or the Bayesian-Optimization (BO)-based approach proposed by Heinen *et al.*[23]

Most of the algorithms discussed are designed to select maximally diverse training sets,[21,22,24,43,44] which does not ensure the chosen subsets retain or improve property predictions on a single target molecule. The SML method[25] is the only reported algorithm, to the best of our knowledge, that directly tackles this problem. Therein, a subset of training points is chosen based on the distance of their representations to that of the target molecule. However, SML has not yet been extensively studied for the extrapolation task: training on datasets of smaller molecules and predicting the properties of larger molecules.[22,26,45–48] Since the cost of quantum-chemical computations scales unfavorably with system size, it is critical to devise a training set selection scheme for the extrapolation setting. To this aim, we develop a Linear Programming (LP) algorithm that selects a set of smaller reference environments that are ideally suited

to predict the properties of larger target molecules. We demonstrate superior results compared to the aforementioned baseline methods in property prediction tasks of larger molecules (from QM9, or drug-like molecules) than those present in training (QM7).

2. Methods

2.1. Search algorithm

The aim is to find an *optimal subset* of molecules from an *existing* dataset of *small* molecules, to constitute an optimal training set for an ML model to predict a quantum-chemical property of a particular *large* target molecule. The number of possible training subsets (*i.e.*, the search space) is made up of all possible combinations of molecular fragments of the database, and grows exponentially with database size. The search algorithm is framed as though searching for an optimal decomposition of the target molecule: local environments (fragments) of each atom in the target molecule are matched to local environments in the small molecules. The algorithm is constructed such that it is independent of the choice of chemical representation[49] as well as the ML model used for property prediction. Two main LP algorithms with this functionality are presented below: one for local (atom-centered) representations, and one for global representations (a single vectorial representation of an entire molecule). In the local approach, two variations are introduced, depending on whether atomic environments are represented using matrices or vectors. The general framework is illustrated in Figure 1. The formulations are designed to be as general as possible so that additional constraints or a new definition of objective value can easily be implemented in the existing framework.

Algorithm 1 presents the general outline of our implementation, which finds fragments that approximate the target molecule in the vectorial space given by the representation of choice. The efficient implementation of the search algorithm as an Integer Linear Program (ILP) is described in the next section. Solutions (subset D of size N in Algorithm 1) correspond to an *atom mapping*: an injective map from the set of target atoms to the set of atoms of the chosen fragments in the database. This map must match atoms of the same type, and pays a cost corresponding to the distance of the atoms matched in their representation space. The value of a given atom mapping is the sum of all its matching costs. The aim to minimize this value to find an optimal map. Chemically speaking, this is akin to searching for fragments that could be put together to form the target molecule as exactly as possible, with no redundant or extra atoms.

The problem of optimal mapping is equivalent to finding a maximum matching of minimal weight in a weighted bipartite graph whose vertices are divided into two disjoint sets, the atoms of the target and the atoms of the database fragments. Edges connect vertices of the first set with those of the second whenever their corresponding atom types match. Each edge has a weight equal to the distance of the corresponding

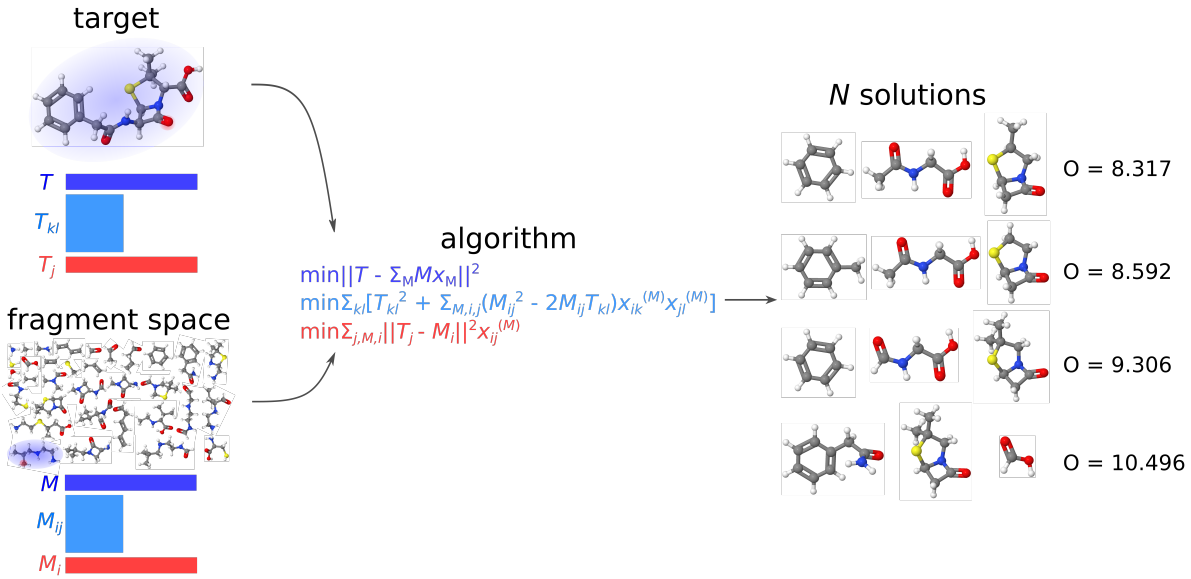


Figure 1: Outline of the search algorithms. Given a target molecule (here, penicillin) and a dataset of molecules, there are three outlined algorithms to search for a set of N combinations of molecules to be used as an optimal training set. The first (in dark blue) is global: *i.e.*, target and training molecules are represented as a whole. The other two algorithms are local: *i.e.*, target and fragment molecules are represented by atom-centered environments. The first (in light blue) represents molecules using matrices, and the second (in red) represents molecules using a vector per atom. Each solution consists of a set of fragment molecules. The objective value O indicates how structurally similar the set of fragments is to the target.

Algorithm 1 Subset selection

Input: database D of fragments, target T, integer N.
Output: selected subset of D of size N.

Initialize the set $S = \emptyset$.
while S has less than N elements **do**
  Find fragments of D that, together, approximate well the target T.
  Add each fragment to S.
  Remove the found combination of fragments from the search space.
end while
Return the first N elements of S.

endpoints in the representation space. Figure 2 represents such a graph in the case where the database has size 2. This is achievable in polynomial time in the number of vertices, that is in our case the total number of atoms in the database.[50] However, the ILP formulation allows us to easily add additional constraints and change the objective value, allowing us to remove specific combinations or penalize undesired ones. The results shown in section 3 demonstrate the effectiveness of this approach.

2.1.1. Atom mapping as an ILP We now describe the subroutine of Algorithm 1 as an ILP for cases in which representations are *local* and summation of the atomic environments results in a global molecular representation. This is notably the case for FCHL19,[51,52] SOAP[53] and SLATM.[54] We are given a target T, a database D of fragments, and a *local representation* of each atom of each molecule. Each atomic environment is represented by a vector in $\mathbb{R}^d$. The *cost* of such a mapping is given by the l^2 norm of the difference of the mapped atoms in the representation space. An optimal solution of our ILP is thus the minimal cost atom mapping between target and fragment molecules.

For each possible atom mapping, we construct a tuple (i, j, M) with $i \in T$, $M \in D$, and $j \in M$ of the same atomic type as i. Since some atom types of the target may not be present in the database, the corresponding atoms are first pruned from the target. Let us denote $i \overset{M}{\mapsto} j$ if atom $i \in T$ is mapped to $j \in M$. We introduce the following decision variable x as,

$$x_{ij}^{(M)} = \begin{cases} 1 \text{ if i} \overset{M}{\mapsto} \text{j}, \\ 0 \text{ otherwise.} \end{cases}$$

This expression gives the interpretation of x and not its definition: we deduce the mapping from the values of the output. The *cost* of each mapping is defined by the L^2 distance between the mapped atoms in the representation space. The ILP is thus formulated as follows.

$$\text{minimize} \qquad \sum_{\substack{i \in T \\ M \in D,\ j \in M}} \|T_i - M_j\|_2^2 x_{ij}^{(M)} \tag{1}$$

$$\text{subject to} \qquad \sum_{M,\ j \in M} x_{ij}^{(M)} = 1 \qquad \forall i \in T, \tag{2}$$

$$\sum_{i \in T} x_{ij}^{(M)} \leq 1 \qquad \forall M \in D,\ \forall j \in M, \tag{3}$$

$$x_{ij}^{(M)} \in \{0, 1\} \qquad \forall i \in T,\ \forall M \in D,\ \forall j \in M. \tag{4}$$

Expression (1) is equal to the sum of squares of the cost associated to each individual mapping. This corresponds directly to the cost of a choice of weighted edges in the bipartite graph illustrated in Figure 2. Equation (2) (resp. inequality (3)) imposes that each atom of the target (resp. database) must be matched exactly once (resp. at most once). Observe that replacing the equal sign in (2) by a "$\leq$" would make the algorithm map singular atoms of the target. Such *closest point search* over each individual atom would not provide solutions representing the target in its entirety.

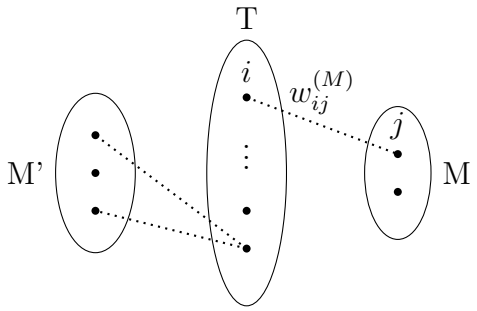


Figure 2: Weighted bipartite graph corresponding to a target T and database $D = \{M, M'\}$. For each fragment $M \in D$ and for each atoms $i \in T$ and $j \in M$ with same atom types, the edge between i and j has cost $w_{ij}^{(M)} = \|T_i - M_j\|_2^2$, where T_i and M_j are the representations of each atom.

2.1.2. Removing found solutions Once a solution of good quality is found, we remove it from the state space of the ILP and proceed. Let $S \subset D$ be a nonempty subset of the database D which we wish to remove. We introduce indicator variables $\{y_M\}_{M \in D}$ that attest whether a fragment is taken. The original ILP formulation with S removed may be rewritten as follows.

$$\text{minimize} \qquad \sum_{\substack{i \in T \\ M \in D,\ j \in M}} \|T_i - M_j\|_2^2 x_{ij}^{(M)}$$

$$\text{subject to} \qquad (2), (3), \text{ and } (4),$$

$$\begin{cases} y_M \leq \sum\limits_{\substack{i \in T, \\ j \in M}} x_{ij}^{(M)} \\ \\ y_M \geq x_{ij}^{(M)} \end{cases} \qquad \forall i \in T,\ \forall j \in M, \tag{5}$$

$$\sum_{M \in S} y_M \leq |S| - 1, \tag{6}$$

$$y_M \in \{0, 1\} \qquad \forall M \in D. \tag{7}$$

Inequalities (5) express the logical disjunction $y_M = \bigvee\limits_{i \in T, j \in M} x_{ij}^{(M)}$. Inequality (6) forbids the set S to be taken as a whole.

Remark importantly that we did not remove each individual fragment (atomic environment) of S from the database, but rather remove the whole combination (entire molecule). This ensures that the solutions are chemically meaningful.

2.1.3. Penalising undesirable solutions Without further constraints or penalty terms, this allows for fragments that, when including the entire molecule associated with the fragment into the pool, introduce many additional atoms compared to the target molecule. This may reduce the efficiency of the chosen subsets. Therefore we introduce two penalty terms: (i) we penalise the introduction of excess atom types that are non-hydrogen and (ii) we penalise the size difference between the target and the fragments. The final ILP is given by the following, for $p \geq 0$ is a constant of choice.

$$\begin{aligned} &\text{minimize} && \sum_{\substack{i \in T \\ M \in D,\ j \in M}} \|T_i - M_j\|_2^2 x_{ij}^{(M)} + p \cdot \left[\sum_M (|M| \cdot y_M) - |T| \right] && (8) \\ &\text{subject to} && (2) - (7) \end{aligned}$$

In section 3, we present results for penalty constants $p = 0$ and 1.

2.1.4. Implementation and parameters The optimization algorithms are implemented using the Gurobi Optimization[55] software, using the python interface `gurobipy`, version 10.0.2. The Gurobi model parameter `PoolSearchMode` is set to 2, meaning that the N best solutions are searched for. A callback function is used in `Model.optimize()` in order to remove the solution found and count the total number of unique molecular fragments already found. The ILP runs until N unique fragments are stored, which makes up the subset to be used as the training set.

When using the QM9 database, hydrogen atoms were pruned as a preprocessing step. While this lowers the precision in the true objective value of the solutions, it significantly speeds up the process and is expected to have a small impact since many hydrogen-to-hydrogen map variables are not defined.

2.2. Molecular representations

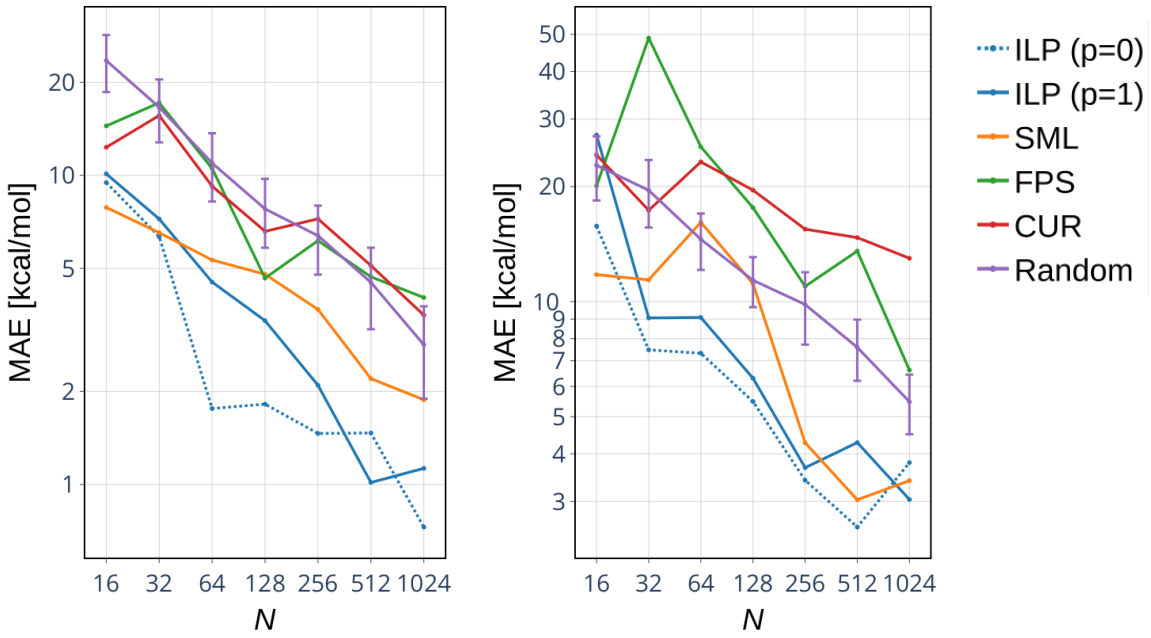
For all applications, we utilized the FCHL19 many-body representation[51,56], which includes both two- and three-body terms to encode the atomic environments of atoms within molecules. With the FCHL19 representation in its local variant, there is one feature vector that describes each atom in the molecule.

2.3. Quantum chemistry

Using the reference geometries, QM7 reference electronic energies[57] were recomputed at the PBE0-D3/def2-SVP level[58–60] with ORCA version 5.0.[61] Analogously, 10 QM9 and 10 drug-like molecules (vide infra) were optimized at the PBE0-D3/def2-SVP level[58–60] using ORCA version 5.0[61], and their electronic energies used for consistency. This ensures that the level of theory is consistent across experiments.

3. Results and Discussion

In this section, we compare the performance of the ILP to existing training set selection approaches (SML,[25] FPS, CUR[30] and random subset selection) on the prediction of atomization energies of differently sized target molecules. As a sanity check, we first test the ILP on 10 QM7 target molecules (removed from the training set). Naturally, these molecules resemble the size and composition of other QM7 molecules. The mean absolute error (MAE) is obtained as an average over the ten target molecules respectively and shown as a function of training set size N (Figure 3a) for the different methods. The same exercise is performed for 10 QM9 molecules with 9 heavy atoms, thus extrapolating slightly beyond the size of the training set molecules, and illustrated in Figure 3b.



(a) Error is averaged on ten out-of-sample QM7 molecules.

(b) Error is averaged on ten QM9 molecules.

Figure 3: Learning curves, mean absolute error (MAE) as a function of training set size. the number of QM7 fragments added N. Penalty term $p = 0$ or $p = 1$. This work, Integer Linear Programming (ILP), selected machine learning[25] (SML), furthest point sampling (FPS), CUR[30] as well as random splits.

While the ILP is not necessarily optimized for the interpolation regime, it nevertheless is outperformed only by SML in the ultra-low data regime ($N \leq 32$). It outperforms all other methods for all dataset sizes and outperforms SML for $32 \leq N \leq 1024$. This suggests that the ILP approach might be usable even for the interpolation task, for "standard" training set selection. The models with and without a penalty constant ($\text{ILP}_{p=0}$ and $\text{ILP}_{p=1}$) both lead to good performance, where $\text{ILP}_{p=1}$ results in a more stable (*i.e.*, almost monotonically decreasing) learning curve but $\text{ILP}_{p=0}$ a steeper learn-

ing curve. In the first extrapolation task (Figure 3b), $\text{ILP}_{p=0}$ leads to best results, again except for SML at ultra-low N ($16 < N < 32$).

figure with chemdraw drugs (can also be embedded in the same figure with learning curves)

As a third example, we predict the atomisation energies of ten significantly larger drug-like molecules containing from 11 to 37 heavy atoms (27±7.7 on average), presented in Figure 4. Except for at the ultra-low $N = 16$, where FPS and $\text{ILP}_{p=1}$ have the same MAE, $\text{ILP}_{p=1}$ outperforms the other training set selection strategies by a large margin. Since the target molecules are larger here, the selection procedure is more challenging and the algorithm benefits from the inclusion of a penalty constant. $\text{ILP}_{p=1}$ results in a monotonically decreasing learning curve until $N = 512$, after which it is likely that most fragments are chemically similar and do not add much chemical functionality to the training set. It is also worth noting that here all selection methods except FPS and ILP are consistently worse than random. We do acknowledge that the absolute errors on the drug-like targets are very high (∼50 kcal/mol).

how to defend this? lol

JAN: One reason for the larger error is that energy is an extensive quantity, which results in a much larger scale of atomization energies for drugs, leading to greater errors. However, the mean absolute percentage errors (MAPEs) are comparable to those predicted for the QM9.

..simple: others methods are even worse and we push extrapolation to its limits! also the error must be compared to the scale of energies.

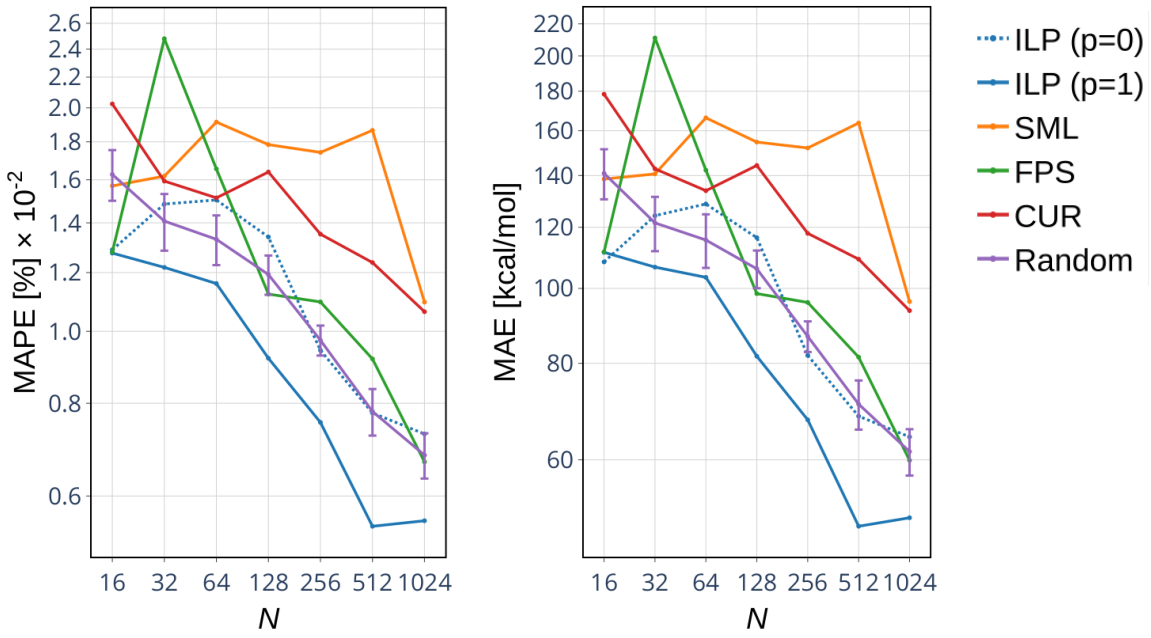


Figure 4: Mean absolute percentage error (MAPE), left, and mean absolute error (MAE), right, averaged over ten drug molecules for various molecular fragment selection schemes.

Finally, we illustrate the time taken to run the ILP in Figure 5. The timing increases non-linearly with the number of training points N. Since we have illustrated (Figures 3-4) that the ILP outperforms other training set selection methods already at low N, and that the prediction MAEs will saturate at higher N, we suggest the user to opt

for a low-intermediate value of N for an ideal trade-off between training set selection time and eventual performance on the target. In case the target molecules need to be labelled, a minimal value of N also considerably reduces the computational budget of eventual quantum chemical calculations.

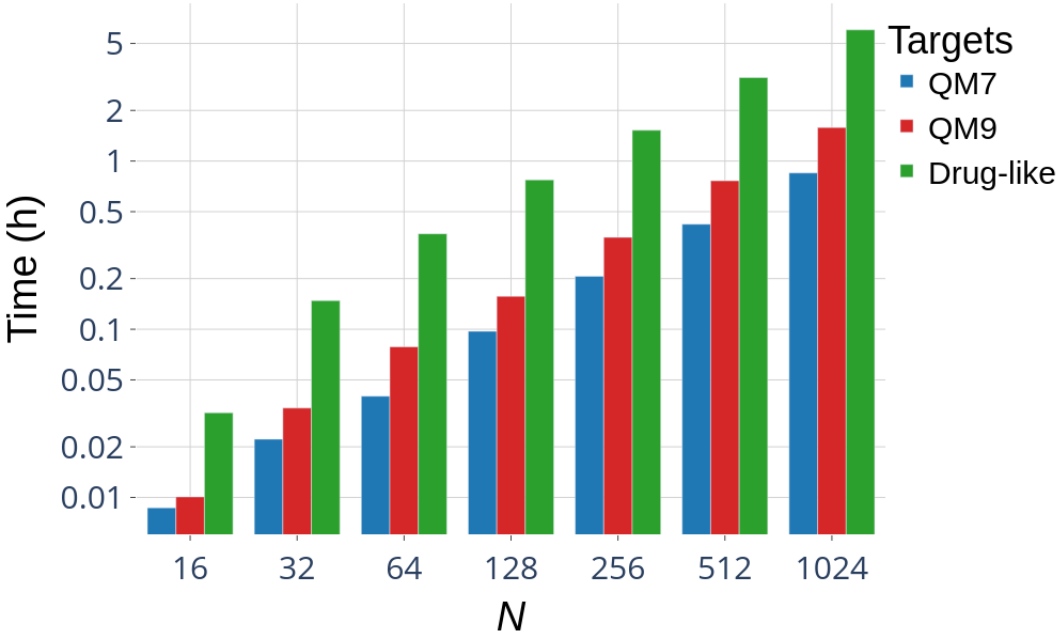


Figure 5: Average time in hours taken to generate N QM7 fragments for each target type. The average is done over 10 out-of-sample targets of each type. The slopes in the double log scales show a near-linear trend of time as function of the number of fragments to generate.

JAN: some words on the algo being written in python and could likely benefit from reimplentation in cpp or faster languages.

4. Conclusion

Developing machine learning models for quantum-chemical applications remains dependent on the availability or otherwise costly construction of labelled datasets. Particularly for large molecules, labelling is expensive and fewer open-source datasets are available. To address this gap, we propose an Integer Linear Programming (ILP) approach to select an optimal subset of small molecules from pre-existing databases to train machine learning models to predict properties of larger molecules. We illustrate that our strategy outperforms existing methods for training set selection and propose this tool as a means to reduce the computational burden of large-molecule property prediction.

JAN: perspective/outlook

- Not sure how many fragments are enough for a given accuracy, take as many as we can afford! should be find because our learning curves decay smoothly

- ...

5. Software Availability

The optimization algorithms are implemented using the Gurobi Optimization[55] software, using the python interface `gurobipy`, version 10.0.2. The algorithms are available at `https://github.com/puckvg/molekuehl`.

Acknowledgements

The authors are grateful to the EPFL for financial support. P.v.G., R.L., J.W. and C.C. acknowledge the National Centre of Competence in Research (NCCR) "Sustainable chemical process through catalysis (Catalysis)" of the Swiss National Science Foundation (SNSF, Grant No. 180544) for financial support. M.H., F.E. and C.C acknowledge the FSB Intrafaculty Seed Funding. F.E. acknowledges support from the Swiss National Science Foundation (SNSF) (Grant 185030).

References

[1] Julia Westermayr et al. "Perspective on integrating machine learning into computational chemistry and materials science". In: *The Journal of Chemical Physics* 154.23 (2021).

[2] Rafael Gómez-Bombarelli and Alán Aspuru-Guzik. "Machine learning and big-data in computational chemistry". In: *Handbook of Materials Modeling: Methods: Theory and Modeling* (2020), pp. 1939–1962.

[3] O Anatole von Lilienfeld and Kieron Burke. "Retrospective on a decade of machine learning for chemical discovery". In: *Nature communications* 11.1 (2020), p. 4895.

[4] Heather J Kulik et al. "Roadmap on machine learning in electronic structure". In: *Electronic Structure* 4.2 (2022), p. 023004.

[5] Philippe Schwaller et al. "Machine intelligence for chemical reaction space". In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* (2022), e1604.

[6] Carles Bo, Feliu Maseras, and Núria López. "The role of computational results databases in accelerating the discovery of catalysts". In: *Nature catalysis* 1.11 (2018), pp. 809–810.

[7] Aditya Nandy, Chenru Duan, and Heather J Kulik. "Audacity of huge: overcoming challenges of data scarcity and data quality for machine learning in computational materials discovery". In: *Current Opinion in Chemical Engineering* 36 (2022), p. 100778.

[8] Claudia Draxl and Matthias Scheffler. "NOMAD: The FAIR concept for big data-driven materials science". In: *Mrs Bulletin* 43.9 (2018), pp. 676–682.

[9] Justin S Smith, Olexandr Isayev, and Adrian E Roitberg. "ANI-1, A data set of 20 million calculated off-equilibrium conformations for organic molecules". In: *Scientific data* 4.1 (2017), pp. 1–8.

[10] Raghunathan Ramakrishnan et al. "Quantum chemistry structures and properties of 134 kilo molecules". In: *Scientific data* 1.1 (2014), pp. 1–7.

[11] Anubhav Jain et al. "Commentary: The Materials Project: A materials genome approach to accelerating materials innovation". In: *APL materials* 1.1 (2013), p. 011002.

[12] Maho Nakata and Tomomi Shimazaki. "PubChemQC project: a large-scale first-principles electronic structure database for data-driven chemistry". In: *Journal of chemical information and modeling* 57.6 (2017), pp. 1300–1308.

[13] Johannes Hoja et al. "QM7-X, a comprehensive dataset of quantum-mechanical properties spanning the chemical space of small organic molecules". In: *Scientific data* 8.1 (2021), p. 43.

[14] Grégoire Montavon et al. "Machine learning of molecular electronic properties in chemical compound space". In: *New Journal of Physics* 15.9 (2013), p. 095003. URL: `http://stacks.iop.org/1367-2630/15/i=9/a=095003`.

[15] Raghunathan Ramakrishnan et al. "Electronic spectra from TDDFT and machine learning in chemical space". In: *The Journal of Chemical Physics* 143.8 (Aug. 2015), p. 084111. ISSN: 0021-9606. DOI: 10.1063/1.4928757. eprint: https://pubs.aip.org/aip/jcp/article-pdf/doi/10.1063/1.4928757/15500464/084111_1_online.pdf. URL: https://doi.org/10.1063/1.4928757.

[16] Rebecca M. Neeser et al. "QMugs 1.1: Quantum mechanical properties of organic compounds commonly encountered in reactivity datasets". In: *Chemical Data Collections* 46 (2023), p. 101040. ISSN: 2405-8300. DOI: https://doi.org/10.1016/j.cdc.2023.101040. URL: https://www.sciencedirect.com/science/article/pii/S2405830023000514.

[17] Xiaohui Qu et al. "The Electrolyte Genome project: A big data approach in battery materials discovery". In: *Computational Materials Science* 103 (2015), pp. 56–67. ISSN: 0927-0256. DOI: https://doi.org/10.1016/j.commatsci.2015.02.050. URL: https://www.sciencedirect.com/science/article/pii/S0927025615001512.

[18] Simone Gallarati et al. "OSCAR: an extensive repository of chemically and functionally diverse organocatalysts". In: *Chemical Science* 13.46 (2022), pp. 13782–13794.

[19] Alexandra Wahab et al. "The COMPAS Project: A Computational Database of Polycyclic Aromatic Systems. Phase 1: cata-Condensed Polybenzenoid Hydrocarbons". In: *Journal of Chemical Information and Modeling* 62.16 (2022), pp. 3704–3713.

[20] Nicholas J Browning et al. "Genetic optimization of training sets for improved machine learning models of molecular properties". In: *The journal of physical chemistry letters* 8.7 (2017), pp. 1351–1359.

[21] Rose K Cersonsky et al. "Improving sample and feature selection with principal covariates regression". In: *Machine Learning: Science and Technology* 2.3 (2021), p. 035038.

[22] Raimon Fabregat et al. "Local Kernel Regression and Neural Network Approaches to the Conformational Landscapes of Oligopeptides". In: *Journal of Chemical Theory and Computation* 18.3 (2022), pp. 1467–1479.

[23] Stefan Heinen et al. *Reducing Training Data Needs with Minimal Multilevel Machine Learning (M3L)*. 2023. arXiv: 2308.11196 [physics.chem-ph].

[24] Giulio Imbalzano et al. "Automatic selection of atomic fingerprints and reference configurations for machine-learning potentials". In: *The Journal of chemical physics* 148.24 (2018).

[25] Dominik Lemm, Guido Falk von Rudorff, and O. Anatole von Lilienfeld. *Improved decision making with similarity based machine learning*. 2023. arXiv: 2205.05633 [physics.chem-ph].

[26] Bing Huang and O Anatole von Lilienfeld. "Quantum machine learning using atom-in-molecule-based fragments selected on the fly". In: *Nature Chemistry* 12.10 (2020), pp. 945–951.

[27] Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. *Coresets for Data-efficient Training of Machine Learning Models*. 2020. arXiv: 1906.01827 [cs.LG].

[28] Zalán Borsos, Mojmir Mutny, and Andreas Krause. "Coresets via Bilevel Optimization for Continual Learning and Streaming". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 14879–14890. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/aa2a77371374094fe9e0bc1de3f94ed9-Paper.pdf.

[29] Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition". In: *Proceedings of 27th Asilomar conference on signals, systems and computers*. IEEE. 1993, pp. 40–44.

[30] Michael W Mahoney and Petros Drineas. "CUR matrix decompositions for improved data analysis". In: *Proceedings of the National Academy of Sciences* 106.3 (2009), pp. 697–702.

[31] Zhengwei Peng. "Very large virtual compound spaces: construction, storage and utility in drug discovery". In: *Drug Discovery Today: Technologies* 10.3 (2013), e387–e394. ISSN: 1740-6749. DOI: https://doi.org/10.1016/j.ddtec.2013.01.004.

[32] Jean-Louis Reymond. "The Chemical Space Project". In: *Accounts of Chemical Research* 48.3 (2015), pp. 722–730. DOI: 10.1021/ar500432k.

[33] Lars Ruddigkeit et al. "Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17". In: *Journal of chemical information and modeling* 52.11 (2012), pp. 2864–2875.

[34] Lorenz C Blum and Jean-Louis Reymond. "970 million druglike small molecules for virtual screening in the chemical universe database GDB-13". In: *Journal of the American Chemical Society* 131.25 (2009), pp. 8732–8733.

[35] Anna Gaulton et al. "The ChEMBL database in 2017". In: *Nucleic Acids Research* 45.D1 (Nov. 2016), pp. D945–D954. DOI: 10.1093/nar/gkw1074.

[36] Teague Sterling and John J. Irwin. "ZINC 15 – Ligand Discovery for Everyone". In: *Journal of Chemical Information and Modeling* 55.11 (2015), pp. 2324–2337. DOI: 10.1021/acs.jcim.5b00559.

[37] Sunghwan Kim et al. "PubChem 2019 update: improved access to chemical data". In: *Nucleic Acids Research* 47.D1 (Oct. 2018), pp. D1102–D1109. DOI: 10.1093/nar/gky1033. URL: https://doi.org/10.1093/nar/gky1033.

[38] Maria Sorokina et al. "COCONUT online: collection of open natural products database". In: *Journal of Cheminformatics* 13.1 (2021), pp. 1–13.

[39] David S Wishart et al. "DrugBank 5.0: a major update to the DrugBank database for 2018". In: *Nucleic Acids Research* 46.D1 (Nov. 2017), pp. D1074–D1082. DOI: 10.1093/nar/gkx1037.

[40] Konstantin Gubaev, Evgeny V Podryabinkin, and Alexander V Shapeev. "Machine learning of molecular properties: Locality and active learning". In: *The Journal of chemical physics* 148.24 (2018).

[41] Evgeny V Podryabinkin and Alexander V Shapeev. "Active learning of linearly parametrized interatomic potentials". In: *Computational Materials Science* 140 (2017), pp. 171–180.

[42] Sijmen De Jong and Henk AL Kiers. "Principal covariates regression: part I. Theory". In: *Chemometrics and Intelligent Laboratory Systems* 14.1-3 (1992), pp. 155–164.

[43] Kevin Rossi et al. "Simulating Solvation and Acidity in Complex Mixtures with First-Principles Accuracy: The Case of CH3SO3H and H2O2 in Phenol". In: *Journal of Chemical Theory and Computation* 16.8 (2020), pp. 5139–5149. DOI: 10.1021/acs.jctc.0c00362.

[44] Felix Musil et al. "Fast and accurate uncertainty estimation in chemical machine learning". In: *Journal of chemical theory and computation* 15.2 (2019), pp. 906–915.

[45] Matthias Rupp, Raghunathan Ramakrishnan, and O Anatole Von Lilienfeld. "Machine learning for quantum mechanical properties of atoms in molecules". In: *The Journal of Physical Chemistry Letters* 6.16 (2015), pp. 3309–3313.

[46] David M Wilkins et al. "Accurate molecular polarizabilities with coupled cluster theory and machine learning". In: *Proceedings of the National Academy of Sciences* 116.9 (2019), pp. 3401–3406.

[47] Hyunwook Jung et al. "Size-Extensive Molecular Machine Learning with Global Representations". In: *ChemSystemsChem* 2.4 (2020), e1900052.

[48] Andrea Grisafi et al. "Transferable machine-learning model of the electron density". In: *ACS central science* 5.1 (2018), pp. 57–64.

[49] Felix Musil et al. "Physics-Inspired Structural Representations for Molecules and Materials". In: *Chemical Reviews* 121.16 (2021), pp. 9759–9815. DOI: 10.1021/acs.chemrev.1c00021.

[50] Alexander Schrijver. "Linear programming methods and the bipartite matching polytope". In: *Combinatorial Optimisation: Polyhedra and Efficiency*. Springer, 2012. Chap. 17.

[51] Felix A. Faber et al. "Alchemical and structural distribution based representation for universal quantum machine learning". In: *J. Chem. Phys.* 148.24 (2018), p. 241717. DOI: 10.1063/1.5020710.

[52] Anders S. Christensen et al. "FCHL Revisited: Faster and More Accurate Quantum Machine Learning". In: *J. Chem. Phys.* 152.4 (2020), p. 044107. DOI: 10.1063/1.5126701.

[53] Albert P. Bartók, Risi Kondor, and Gábor Csányi. "On representing chemical environments". In: *Phys. Rev. B* 87 (18 May 2013), p. 184115. DOI: 10.1103/PhysRevB.87.184115. URL: https://link.aps.org/doi/10.1103/PhysRevB.87.184115.

[54] B Huang and OA von Lilienfeld. "Quantum machine learning using atom-in-molecule-based fragments selected on the fly". In: *Nat Chem* 12.10 (Oct. 2020). Epub 2020 Sep 14, pp. 945–951. DOI: 10.1038/s41557-020-0527-z.

[55] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. 2022. URL: https://www.gurobi.com.

[56] Anders S Christensen et al. "FCHL revisited: Faster and more accurate quantum machine learning". In: *The Journal of chemical physics* 152.4 (2020), p. 044107.

[57] Matthias Rupp et al. "Fast and accurate modeling of molecular atomization energies with machine learning". In: *Physical review letters* 108.5 (2012), p. 058301.

[58] C. Adamo, M. Cossi, and V. Barone. "An accurate density functional method for the study of magnetic properties: the PBE0 model". In: *J. Mol. Struc.-THEOCHEM* 493.1–3 (Dec. 1999), pp. 145–157. ISSN: 0166-1280. DOI: 10.1016/s0166-1280(99)00235-3. URL: http://dx.doi.org/10.1016/S0166-1280(99)00235-3.

[59] Florian Weigend and Reinhart Ahlrichs. "Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy". In: *Phys. Chem. Chem. Phys.* 7.18 (2005), p. 3297. ISSN: 1463-9084. DOI: 10.1039/b508541a. URL: http://dx.doi.org/10.1039/B508541A.

[60] Stefan Grimme et al. "A consistent and accurateab initioparametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu". In: *J. Chem. Phys.* 132.15 (Apr. 2010). ISSN: 1089-7690. DOI: 10.1063/1.3382344. URL: http://dx.doi.org/10.1063/1.3382344.

[61] Frank Neese. "Software update: The ¡scp¿ORCA¡/scp¿ program system—Version 5.0". In: *Wiley Interdiscip. Rev. Comput. Mol. Sci.* 12.5 (Mar. 2022). ISSN: 1759-0884. DOI: 10.1002/wcms.1606. URL: http://dx.doi.org/10.1002/wcms.1606.