Symbolic Learning for Material Discovery

Dan Cunnington
IBM Research Europe
dancunnington@uk.ibm.com

Flaviu Cipcigan
IBM Research Europe
flaviu.cipcigan@ibm.com

Rodrigo Neumann Barros Ferreira
IBM Research Brazil
rneumann@br.ibm.com

Jonathan Booth
Science and Technology Facilities Council
jonathan.booth@stfc.ac.uk

Abstract

Discovering new materials is essential to solve challenges in climate change, sustainability and healthcare. A typical task in materials discovery is to search for a material in a database which maximises the value of a function. That function is often expensive to evaluate, and can rely upon a simulation or an experiment. Here, we introduce SyMDis, a sample efficient optimisation method based on symbolic learning, that discovers near-optimal materials in a large database. SyMDis performs comparably to a state-of-the-art optimiser, whilst learning interpretable rules to aid physical and chemical verification. Furthermore, the rules learned by SyMDis generalise to unseen datasets and return high performing candidates in a zero-shot evaluation, which is difficult to achieve with other approaches.

1 Introduction

Accelerating the discovery of new materials is one of the fundamental challenges facing the scientific community. Many important applications such as carbon capture and battery technology are reliant on new materials in order to realise their potential value and limit global warming in accordance with the 2015 Paris agreement [19]. However, evaluating the large design landscape is a computationally infeasible task, and there is simply not enough time to proceed using traditional methods. Therefore, there is increasing attention from the AI community to accelerate material discovery [12, 13].

Many approaches aim to discover an optimal material within a given database by running in-silico simulations to estimate a desired metric. Whilst the naive brute-force approach is obviously inefficient, optimisation techniques such as Bayesian Optimisation and active learning select high performing candidates whilst minimising the number of evaluations of an expensive in-silico simulation [5, 14]. This is often achieved by balancing an exploration vs. exploitation trade-off of the material search space. However, the underlying machine learning models used are often difficult to interpret, and also can not be easily transferred to new datasets. Therefore, it is unclear whether materials are being selected based on criteria that are physically and chemically sound.

To tackle this problem, we introduce **SyMDis**: Symbolic learning for Material Discovery, which exploits symbolic AI techniques to learn naturally interpretable rules that map material descriptors to the desired performance metric. SyMDis is inspired by active learning, and on each iteration, a small number of samples for in-silico computation are selected from a database based on the learned rules. Then, new rules are learned, and a new batch of materials are selected for evaluation. The goal is to discover a high performing material within a large database, whilst minimising the number of calls to the (expensive) in-silico computation. We evaluate SyMDis for the task of identifying suitable Metal Organic Frameworks (MOFs) to maximise CO_2 uptake for carbon capture. Our experiments show that SyMDis performs comparably to a state-of-the-art Bayesian Optimisation method, returning a

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

MOF with a CO_2 uptake within 92.5% of the maximum in a database of 19.4K MOFs, after only 100 calls to the objective function. This increases to a MOF with 97.5% of the maximum with 250 calls. Crucially, SyMDis learns interpretable rules that express why certain MOFs were chosen in terms of various MOF descriptors. This enables human domain experts to verify the learned rules are physically and chemically sound. Furthermore, our evaluation demonstrates the rules learned by SyMDis can generalise to new unseen datasets in a zero-shot evaluation, obtaining high performing MOFs without requiring *any* calls to the objective function on the new dataset. This would be difficult to achieve with other approaches.

2 Method

Problem statement. Let us assume a material is represented by a set of descriptor value pairs $\boldsymbol{x} = \{\langle d, v\rangle, \ldots\}$, where $d \in \mathcal{D}$ is a descriptor and $v \in \mathcal{V}$ is an associated value. In this paper, we assume each descriptor has a defined value, i.e., $\forall d \in \mathcal{D}$, there exists a $v' \in \mathcal{V}$ s.t. $\langle d, v'\rangle \in \boldsymbol{x}$. An objective function $f : (\mathcal{D} \times \mathcal{V})^{|\mathcal{D}|} \rightarrow \mathbb{R}$ maps a set of descriptor value pairs into a target metric $y \in \mathbb{R}$. Let us also assume a database of *unlabelled* material samples $B = \{\boldsymbol{x}, \ldots\}$. The goal is to find the material in B that maximises y, given f, i.e., $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in B} f(\boldsymbol{x})$, whilst minimising the number of evaluations of f, as f could be computationally expensive (e.g., an in-silico simulation). To tackle this problem, SyMDis uses an iterative approach, inspired by active learning.

SyMDis. In addition to the database of unlabelled materials B, let us also define a database of (initially empty) *labelled* materials, $\hat{B} = \emptyset$. SyMDis begins with a random sample of n materials $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$ chosen from B, which are evaluated using f, to obtain a set of targets $\{f(\boldsymbol{x}_1), ..., f(\boldsymbol{x}_n)\}$. These materials are then removed from B, and alongside their targets, are stored in $\hat{B}$, i.e., $\hat{B} = \hat{B} \cup \{\langle \boldsymbol{x}_1, f(\boldsymbol{x}_1)\rangle, .., \langle \boldsymbol{x}_n, f(\boldsymbol{x}_n)\rangle\}$. SyMDis then constructs a set of training examples for a symbolic learner, which learns a set of logical rules that selects materials from B for the next iteration. SyMDis terminates after a given number of iterations, or when B is empty. The architecture is presented in Figure 1. Let us now describe how the labelled materials in $\hat{B}$ are converted into training examples for the symbolic learner.

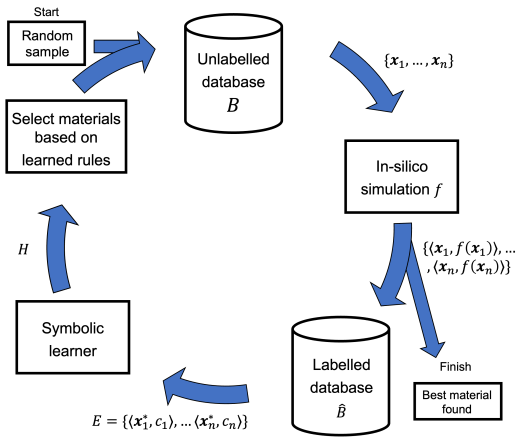


Figure 1: SyMDis architecture

Generating symbolic training examples. On each iteration, a set $\{\langle \boldsymbol{x}_1^*, y_1^*\rangle, ..., \langle \boldsymbol{x}_n^*, y_n^*\rangle\}$ of n samples with the largest target metrics y^* are selected from $\hat{B}$. Note on the first iteration this set of samples is equal to the initial random samples. SyMDis then generates a classification for each sample, by binning the y^* values into a class label $c \in \{\textit{excellent}, \textit{good}, \textit{moderate}, \textit{poor}\}$. The samples with a y^* value in the largest 10% receive the label $c = \textit{excellent}$, the next 20% receive the label $c = \textit{good}$, the next 30% receive the label $c = \textit{moderate}$, and the remaining 40% receive the label $c = \textit{poor}$. Generating class labels in this manner helps to prevent over-fitting to a particular sample, as provided n is large enough, multiple samples receive the same class label. This particular percentage split was chosen to enable the unlabelled database to be filtered based on rules learned from high performing candidates, i.e., candidates within the top 10%, as priority is given to *excellent* candidates during filtering for the next iteration, see *Selecting materials* below. The chosen split performs well in our experiments, and further tuning and investigation is left as future work. The set of examples for the symbolic learner can then be generated as $E = \{\langle \boldsymbol{x}_1^*, c_1\rangle, ..., \langle \boldsymbol{x}_n^*, c_n\rangle\}$.

Symbolic learner. A symbolic learner can then learn a logic program called a hypothesis $H \in \mathcal{H}$ that explains the training examples. In this paper, we assume H is a propositional logic program, although since the symbolic learning component in SyMDis is modular, this program could be of any logical expressivity or in any logical format. Given this modularity, we now define a general notion of a symbolic learner. Firstly, the score of a hypothesis w.r.t. a training example $\langle \boldsymbol{x}^*, c\rangle$ is defined as:

$$SCORE(H, \langle \boldsymbol{x}^*, c \rangle) = \begin{cases} 1 & \text{if } H(\boldsymbol{x}^*) \models c \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $\models$ denotes logical entailment. The goal of the symbolic learner is to learn a hypothesis H s.t.:

$$H^* = \arg\max_{H \in \mathcal{H}} \sum_{\langle \boldsymbol{x}^*, c \rangle \in E} SCORE(H, \langle \boldsymbol{x}^*, c \rangle) \quad (2)$$

which intuitively means to learn a H that maximises the number of training examples where the correct class label is output, given the input descriptors.

Selecting materials Finally, SyMDis selects a new batch of n samples from B using the learned H. This is achieved by evaluating all of the materials in B using H to determine a predicted class label $\hat{c} \in \{\textit{excellent}, \textit{good}, \textit{moderate}, \textit{poor}\}$. Relative to f, this is a cheap computation. SyMDis progressively selects samples in the order of the quality of the predicted label (*excellent* $\rightarrow$ *poor*), until a total of n is reached. A random sample is performed if selecting all samples within a class would exceed the total of n. This forms the new batch of samples $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ for the next iteration.

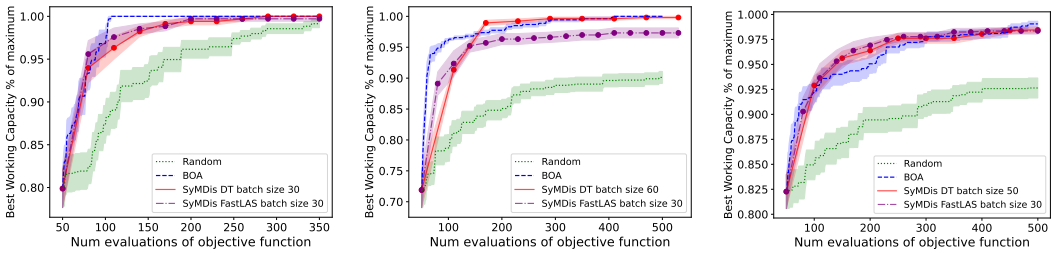
3 Experiments

To evaluate SyMDis, we use three MOF datasets, where the task is to discover the best material for performing carbon capture, whilst minimising the number of in-silico adsorption simulations required. The target metric is to maximise *working capacity*, which is defined as the amount of CO_2 adsorbed during high pressure minus the remaining CO_2 left on the material during desorption (low pressure). Our evaluation aims to address the following questions; (**Q1**) How does SyMDis compare to a state-of-the-art Bayesian Optimisation approach, known for its sample efficiency? (**Q2**) Can SyMDIS take advantage of a logic-based symbolic learner to improve performance and/or interpretability? (**Q3**) Can SyMDis discover physically and chemically sound rules? (**Q4**) Can the learned rules generalise to unseen data, to discover a high performing MOF without requiring *any* calls to the objective function on the new dataset (zero-shot)?

Setup. We utilise three MOF datasets from [16]; *ARABG*, *CoRE2019*, and *BW20K* which contain 387, 9525, and 19,379 MOFs respectively. These datasets are annotated with CO_2 adsorption at pressures 0.15bar and 16bar, with a fixed temperature of 298K. Note that 0.15bar used by [16] is an allusion to the partial pressure of CO_2 in coal-fired flue gas (15% CO_2 / 85% N_2) at ambient pressure. These annotations enable us to perform a comprehensive evaluation without running the simulations, i.e., our objective function is simply a look-up in the database for these features, which can be used to calculate the working capacity. No modifications to the datasets are performed, other than selecting 15 descriptors based on a regression analysis to determine feature importance (see Table 1 in Appendix A). We generate 20 random seeds and select 20 sets of 50 materials at random from each dataset. We plot the best working capacity achieved in terms of a percentage of the maximum, w.r.t. the number of evaluations of our objective function, averaged over the 20 repeats.

For the symbolic learning component, we use a Decision Tree from scikit-learn (denoted **SyMDis DT**), and to address evaluation Q2, FastLAS [11] (denoted **SyMDis FastLAS**), a recent logic-based machine learning approach known for it's ability to generalise and learn expressive logic programs in the form of Answer Set Programming [8]. FastLAS can also learn from noisy data, where each example is given a weight, and the symbolic learner is encouraged to cover examples with higher weight. In our experiments, we assign higher weights to better performing materials to encourage the symbolic learner to learn rules that cover the high performing candidates. We use weights 10, 5, 2, and 1 for the classes excellent, good, mediocre and poor respectively. Future work could investigate additional methods for selecting the weights, and use them to express uncertainty on the descriptor values. We evaluate each variant with varying batch sizes $n \in \{10, 20, ..., 60\}$ and present the best performing result. We also note that FastLAS timed-out after 1 minute with a batch size $n > 30$, so the SyMDis FastLAS variant uses a batch size of $n = 30$ for all experiments.

Baselines. To investigate evaluation Q1, we compare to IBM's state-of-the-art Bayesian Optimisation library (**BOA**) [10]. Following consultation with the authors, and our own experiments, we use Expected Improvement with a Basic Sampler, the Adaptive Epsilon setting, enable x and y normalisation, and set the batch size to 1, as these configurations resulted in the best performance. We



(a) ARABG (b) CoRE2019 (c) BW20K

Figure 2: MOF results. Shaded regions indicate standard error.

```
label(excellent) :- CellV > 5827.87,
      total_SA_gravimetric > 4914.62, Df <= 11.44,
       density > 0.35.

label(good) :- CellV > 5827.87,
      total_SA_gravimetric > 4914.62, Df > 11.44.
```

(a) SyMDis Decision Tree excellent class only

```
label(excellent) :- CellV >= 26335.50.
label(good) :- 13.77 <= Dif <= 13.89.
label(moderate) :- 0.36 <= density <= 0.38.
label(poor) :- 0.39 <= density <= 0.43.
```

(b) SyMDis FastLAS

Figure 3: Example rules learned on BW20K

also evaluate a naive random baseline (**Random**), that randomly selects a material from B at each iteration. For both baselines, we initialise using the same 20 sets of random materials as SyMDis.

Results. The results are presented in Figure 2, and example rules are shown in Figure 3, for one repeat on the BW20K dataset. SyMDis discovers a near optimal MOF in all cases, significantly outperforming random selection. On BW20K, the most challenging dataset, SyMDis discovers a MOF with 92.5% of the maximal working capacity whilst only evaluating the objective function for 100 MOFs out of a total of $\sim$20K samples. This increases to 97.5% with 250 evaluations. To answer evaluation Q1, SyMDis performs comparably to BOA on all datasets, whilst learning naturally interpretable rules (i.e., no post-hoc interpretability is required). For evaluation Q2, FastLAS does not appear to offer an advantage when compared to a Decision Tree in terms of performance, although learns a significantly shorter set of rules, which are easier to interpret (see Figure 5 and Appendix A for analysis). The rules learned by SyMDis enable manual inspection, and indeed correspond to a reasonable chemical explanation. For example, the rules learned on the BW20K dataset for both SyMDis variants show that the unit cell volume descriptor is important. The unit cell within the MOF is the structure that is repeated to make the crystal. It's possible that a larger unit cell is likely to have more space, which could influence adsorption [9]. Also, [6] reports that the gravimetric surface area, pore size, and void fraction are important geometric descriptors for predicting CO_2 uptake. As you can see in Figure 3, the SyMDis Decision Tree variant also uses the gravimetric surface area descriptor. This answers evaluation Q3.

To address Q4, we perform a zero-shot transfer, and apply the rules learned on a source dataset to a target dataset which is completely unseen during the optimisation. We use the learned rules to obtain a predicted class $\hat{c}$ for every material in the target dataset, and take a random sample of 200 materials predicted as *excellent*. We then plot the distribution of working capacities from the sampled MOFs, and compare to 200 random samples. We evaluate all combinations of source/target datasets except where the target is ARABG, as this dataset only contains 387 MOFs. The full results are shown in Appendix B, with an example shown in Figure 4. In most cases, the distribution of working capacity values calculated from the sampled materials is larger than those calculated from a random selection. Surprisingly, even the rules learned on the ARABG dataset can generalise to CoRE2019 and BW20K which have a significantly larger number of MOFs.

Finally, in terms of run-time, SyMDis is very efficient. The average wall-clock time for the Decision Tree variant is 0.11, 2.43, 4.96 seconds to complete the optimisation in full for the ARABG, CoRE2019, and BW20K datasets respectively, and 10.58, 25.25, and 52.57 seconds for the FastLAS variant. Note these run-times include the database lookup as our objective function. In reality, SyMDis adds minimal overhead compared to an expensive objective function.

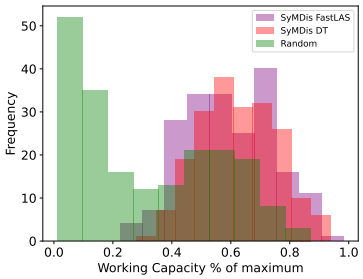
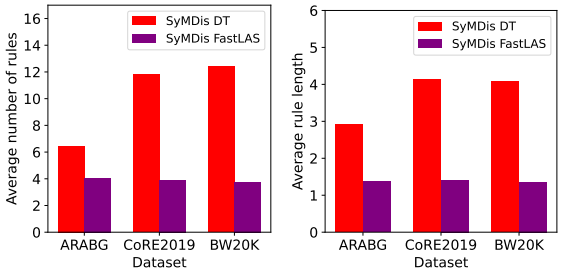


Figure 4: Distribution of MOFs sampled from BW20K using rules learned on ARABG



(a) Avg. number of rules (b) Avg. rule length

Figure 5: Interpretability of the Decision Tree and FastLAS learned rules.

4 Related work

SyMDis is inspired by active learning which interactively queries a third-party to label data [20]. Bayesian Optimisation (BO) [5, 22] is a popular method of active learning, where a surrogate model of an objective function is iteratively constructed given data observations. Whilst BO has achieved state-of-the-art performance in many applications [4, 21], the surrogate model is often a black-box which requires additional components to explain what's been learned. We have demonstrated SyMDis performs comparably to BO on the MOF datasets, whilst providing naturally interpretable rules. A recent approach also learns surrogate MOF adsorption models [18], but relies on a black-box Gaussian Process Regressor. Many approaches have been developed to aid with interpretability, particularly in material discovery [1, 6, 15, 17]. However, these approaches rely on statistical machine learning techniques such as Decision Trees, Random Forests, and Linear Models, which as shown in Figure 5 and Appendix A, can become difficult to interpret when applied to large datasets. In contrast, SyMDis can take advantage of logic-based techniques which offer improved interpretability. As discussed in Section 3, [6] uses a Decision Tree to learn rules for CO_2 uptake, using MOFs from the Northwestern University database [26]. SyMDis learns rules using similar descriptors, and extends this work by integrating the symbolic learner into an active learning loop.

In the context of scientific discovery, symbolic techniques have been applied widely for the purposes of symbolic regression [3, 7, 24, 25], where the goal is to generate a fully interpretable mathematical expression that fits a set of training data points. However, learning a general expression is challenging, and these methods are often sensitive to noise [23]. In contrast, instead of trying to learn an exact mathematical expression, SyMDis learns logical rules that can take advantage of descriptors with different data types (i.e., non-numerical categorical descriptors), and generalise more easily over a set of noisy examples. Furthermore, using a logic-based symbolic learner, SyMDis can easily include existing background knowledge, ensure constraints are satisfied, and learn various logical relations and comparison operators.

5 Conclusion

We have introduced **SyMDis**, a sample-efficient optimisation method that discovers near-optimal materials in a given database, whilst minimising the number of calls to an objective function. We have applied SyMDis to a MOF use-case, and demonstrated SyMDis is able to discover a MOF for CO_2 uptake with 97.5% of the maximum in a database with 19.4K MOFs, requiring only 250 calls to the objective function. Furthermore, SyMDis learns interpretable rules to aid chemical and physical verification, and can generalise via a zero-shot transfer to unseen target datasets.

Future Work. One could extend SyMDis to multi-objective optimisation problems by designing a target objective that incorporates multiple metrics (e.g., *target objective* = *working capacity* + *selectivity* - *cost*. One could also apply SyMDis in other domains, such as solvents or additives. Finally, one could integrate SyMDis with generative models such as GFlowNets [2], by using SyMDis to discover the most optimal candidate output from a generative model. This would combine the benefits of both generation and optimisation approaches to material discovery, as SyMDis could apply more thorough evaluations of novel candidates, before handing off to a lab for further analysis.

Acknowledgments

The authors wish to acknowledge Edward Pyzer-Knapp and Clyde Fare from IBM Research for helping to identify the appropriate BOA configuration for us to use in our experiments. This work was funded through the UKRI Hartree National Centre for Digital Innovation.

References

[1] Alice E. A. Allen and Alexandre Tkatchenko. Machine learning of material properties: Predictive and interpretable multilinear models. *Science Advances*, 8(18):eabm7185, 2022.

[2] Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. *Advances in Neural Information Processing Systems*, 34:27381–27394, 2021.

[3] Bogdan Burlacu, Michael Kommenda, Gabriel Kronberger, Stephan M Winkler, and Michael Affenzeller. Symbolic regression in materials science: Discovering interatomic potentials from data. In *Genetic Programming Theory and Practice XIX*, pages 1–30. Springer, 2023.

[4] Aryan Deshwal, Cory M. Simon, and Janardhan Rao Doppa. Bayesian optimization of nanoporous materials. *Mol. Syst. Des. Eng.*, 6:1066–1086, 2021.

[5] Sanket Diwale, Maximilian K. Eisner, Corinne Carpenter, Weike Sun, Gregory C. Rutledge, and Richard D. Braatz. Bayesian optimization for material discovery processes with noise. *Mol. Syst. Des. Eng.*, 7:622–636, 2022.

[6] Michael Fernandez and Amanda S. Barnard. Geometrical properties can predict co2 and n2 adsorption performance of metal–organic frameworks (mofs) at low pressure. *ACS Combinatorial Science*, 18(5):243–252, 2016. PMID: 27022760.

[7] Jake Fitzsimmons and Pablo Moscato. Symbolic regression modeling of drug responses. In *2018 First International Conference on Artificial Intelligence for Industries (AI4I)*, pages 52–59. IEEE, 2018.

[8] Michael Gelfond and Yulia Kahl. *Knowledge representation, reasoning, and the design of intelligent agents: The answer-set programming approach*. Cambridge University Press, 2014.

[9] Peng Guo, Jiho Shin, Alex G Greenaway, Jung Gi Min, Jie Su, Hyun June Choi, Leifeng Liu, Paul A Cox, Suk Bong Hong, Paul A Wright, et al. A zeolite family with expanding structural complexity and embedded isoreticular structures. *Nature*, 524(7563):74–78, 2015.

[10] Dipti Jasrasaria and Edward O Pyzer-Knapp. Dynamic control of explore/exploit trade-off in bayesian optimization. In *Intelligent Computing: Proceedings of the 2018 Computing Conference, Volume 1*, pages 1–15. Springer, 2019.

[11] Mark Law, Alessandra Russo, Elisa Bertino, Krysia Broda, and Jorge Lobo. Fastlas: Scalable inductive logic programming incorporating domain-specific optimisation criteria. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 2877–2885, 2020.

[12] Jiali Li, Kaizhuo Lim, Haitao Yang, Zekun Ren, Shreyaa Raghavan, Po-Yen Chen, Tonio Buonassisi, and Xiaonan Wang. Ai applications through the whole life cycle of material discovery. *Matter*, 3(2):393–432, 2020.

[13] Yue Liu, Tianlu Zhao, Wangwei Ju, and Siqi Shi. Materials discovery and design using machine learning. *Journal of Materiomics*, 3(3):159–177, 2017. High-throughput Experimental and Modeling Research toward Advanced Batteries.

[14] Turab Lookman, Prasanna V Balachandran, Dezhen Xue, and Ruihao Yuan. Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design. *npj Computational Materials*, 5(1):21, 2019.

[15] Paulius Mikulskis, Morgan R Alexander, and David Alan Winkler. Toward interpretable machine learning models for materials discovery. *Advanced Intelligent Systems*, 1(8):1900045, 2019.

[16] Seyed Mohamad Moosavi, Aditya Nandy, Kevin Maik Jablonka, Daniele Ongari, Jon Paul Janet, Peter G Boyd, Yongjin Lee, Berend Smit, and Heather J Kulik. Understanding the diversity of the metal-organic framework ecosystem. *Nature communications*, 11(1):1–10, 2020.

[17] Eric S Muckley, James E Saal, Bryce Meredig, Christopher S Roper, and John H Martin. Interpretable models for extrapolation in scientific machine learning. *Digital Discovery*, 2023.

[18] Krishnendu Mukherjee, Etinosa Osaro, and Yamil J Colon. Active learning for efficient navigation of multi-component gas adsorption landscapes in a mof. *Digital Discovery*, 2023.

[19] United Nations. Paris agreement 2015. `https://treaties.un.org/pages/ViewDetails.aspx?src=TREATY&mtdsg_no=XXVII-7-d&chapter=27&clang=_en`. Accessed: 30-08-2023.

[20] Burr Settles. Active learning literature survey. *University of Wisconsin-Madison Department of Computer Sciences*, 2009.

[21] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

[22] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

[23] Silviu-Marian Udrescu and Max Tegmark. Ai feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020.

[24] Yiqun Wang, Nicholas Wagner, and James M. Rondinelli. Symbolic regression in materials science. *MRS Communications*, 9(3):793–805, 2019.

[25] Baicheng Weng, Zhilong Song, Rilong Zhu, Qingyu Yan, Qingde Sun, Corey G Grice, Yanfa Yan, and Wan-Jian Yin. Simple descriptor derived from symbolic regression accelerating the discovery of new perovskite catalysts. *Nature communications*, 11(1):3513, 2020.

[26] Christopher E Wilmer, Michael Leaf, Chang Yeon Lee, Omar K Farha, Brad G Hauser, Joseph T Hupp, and Randall Q Snurr. Large-scale screening of hypothetical metal–organic frameworks. *Nature chemistry*, 4(2):83–89, 2012.

A Learned rules

In this section, we analyse the interpretability of the rules learned by SyMDis for both the Decision Tree and FastLAS variants. Before doing so, Table 1 presents an overview of the descriptors used from [16].

Table 1: MOF descriptors used from [16]

Name	Description	Units
ASA	*Accessible surface area to volume ratio*	$m^2\ cm^{-3}$
CellV	*Volume of unit cell of MOF*	$\AA^3$
density	*Density*	$g\ cm^{-3}$
Df	*Diameter of largest free sphere*	$\AA$
Di	*Diameter of largest included sphere*	$\AA$
Dif	*Diameter of largest included sphere along free path*	$\AA$
NASA	*Non-accessible surface area to volume ratio*	$m^2\ cm^{-3}$
POAV	*Ratio of accessible pore volume to mass*	$cm^3\ g^{-1}$
POAVF	*Fraction of total_POV_volumetric that is accessible to probe*	-
PONAV	*Ratio of non-accessible pore volue to mass*	$cm^3\ g^{-1}$
PONAVF	*Fraction of total_POV_volumetric that is non-accessible to probe*	-
total_SA_volumetric	*Surface area to volume ratio regardless of accessibility*	$m^2\ cm^{-3}$
total_SA_gravimetric	*Surface area to mass ratio regardless of accessibility*	$m^2\ g^{-1}$
total_POV_volumetric	*Ratio of pore volume to MOF volume regardless of accessibility*	-
total_POV_gravimetric	*Ratio of pore volume to mass regardless of accessibility*	$cm^3\ g^{-1}$

Figure 3 shows the rules learned for one repeat on the BW20K dataset. During the optimisation for this repeat, the best MOFs discovered were 99.58% and 99.01% of the maximum for the Decision Tree and FastLAS variants respectively, which indicates these models both had strong performance. In terms of interpretability, it is immediately clear that the rules learned by the Decision Tree variant are longer than the rules learned by FastLAS, and are therefore more difficult to interpret. This is why we only show the rules learned for the *excellent* and *good* classes for the SyMDis Decision Tree variant in Figure 3a. To investigate this further, we analyse the interpretability of the rules learned by both SyMDis variants, in terms of the average number of rules learned, and the average length per rule, over the 20 repeats. The intuition is that a shorter set of rules is easier to interpret. Figure 5 presents the results. As you can see, the rules learned by FastLAS are significantly easier to interpret than those learned by the decision tree. Furthermore, the number of rules, and the average length per rule both increase with larger datasets (CoRE2019 and BW20K) for the Decision Tree variant, whereas the size of the FastLAS rules remains constant regardless of dataset size. This indicates that interpreting the rules learned by the Decision Tree may become more difficult when large datasets are used. In practice, one may wish to prioritise interpretability over performance. For example, in Figure 2b, the SyMDis Decision Tree variant outperforms SyMDis FastLAS, but it is more difficult to assess whether the rules learned by the Decision Tree correspond to a reasonable physical and chemical explanation due to their increased length.

In terms of expressivity, in this work the learned rules are propositional and act as a linear classifier. However, one of the benefits of enabling modularity to the symbolic learning component is that higher-order rules could be learned in other tasks using the FastLAS symbolic learner. As FastLAS is based on ASP [8], it can learn highly expressive first-order rules involving negation, choice, constraints, and rules with multiple answer-sets. This would not be possible with the decision tree SyMDis variant. For the purposes of SyMDis, any set of logical rules are supported, provided they can act as a filter on the database for the next iteration. This is feasible, as each database candidate can simply be evaluated against the learned rules for satisfiability, to decide whether or not the candidate is selected for the next iteration.

B Generalisation of learned rules with zero-shot transfer

One of the benefits of learning an interpretable set of rules is the ability to transfer to new target datasets, without requiring *any* calls to the expensive objective function. As the datasets used in this

paper are from different chemical spaces [16], it is interesting to evaluate whether the rules learned on a source dataset can be transferred to a target. This would indicate the learned rules are general, and possibly reflect common underlying chemical or physical phenomena. It is also clearly of practical interest, as a successful zero-shot transfer would significantly reduce the number of evaluations of the objective function. Note that such a transfer is easier to achieve than with a black-box model, where all descriptors present in the source dataset would have to exist in the target dataset. A set of rules on the other hand, can easily be modified in the case of a partial match, where descriptors could be removed from the rules if they don't exist in the target dataset, or certain descriptors could be transformed if the target dataset had any normalisation or other transformation applied. Also, the rules support manual additions by human experts, if any background knowledge or constraints were required in the target domain. This would be difficult to achieve with a black-box model.

We apply the rules learned at the end of each SyMDis optimisation (i.e., the points with the maximum number of calls to the objective function in Figure 2) to unseen target datasets. Specifically, we evaluate all combinations of source/target pairs with the ARABG, CoRE2019, and BW20K datasets, except where the ARABG dataset is the target, since this only contains 387 MOFs. We use the learned rules to obtain a predicted class $\hat{c}$ for every material in the target dataset, and take a random sample of 200 materials predicted as *excellent*. We then plot the distribution of working capacities from the sampled MOFs, and compare to 200 random samples. If the rules have generalised, we expect the 200 samples obtained via the *excellent* prediction from the learned rules to have a larger working capacity than the random samples.

Figures 6 - 9 show the results for all combinations and all 20 repeats. SyMDis provides a clear benefit in; Figure 6 (ARABG to CoRE2019) with 15/20 repeats, Figure 7 (ARABG to BW20K) with all repeats, Figure 8 (CoRE2019 to BW20K) with 18/20 repeats, and in Figure 9 (BW20K to CoRE2019) with 16/20 repeats. This shows the rules learned by SyMDis are able to generalise during a zero-shot transfer and outperform a random selection. In Figures 6 (ARABG to CoRE2019) and 7 (ARABG to BW20K), FastLAS provides a benefit compared to the Decision Tree in 7/20, and 10/20 repeats respectively, whereas in other cases the difference is less significant. This is possibly due to FastLAS learning shorter rules, which are more likely to generalise. The rules learned on BW20K do not generalise as well when applied to CoRE2019 (Figure 9). As BW20K is a *hypothetical* dataset, and CoRE2019 is *experimental* [16], it's likely that some of the rules learned on BW20K may not apply experimentally. Nevertheless, the interpretability of SyMDis enables such analysis and investigation before downstream application, which would be difficult to achieve with a black-box approach. Also, we consider a rule transfer from a hypothetical to an experimental dataset a special case, and the more common use case being a transfer from an experimental to a hypothetical dataset, such as the transfer from CoRE2019 to BW20K in Figure 8. In this case, the rules have improved generalisation compared to Figure 9, and help to select high performing *hypothetical* MOFs that can be further validated experimentally.

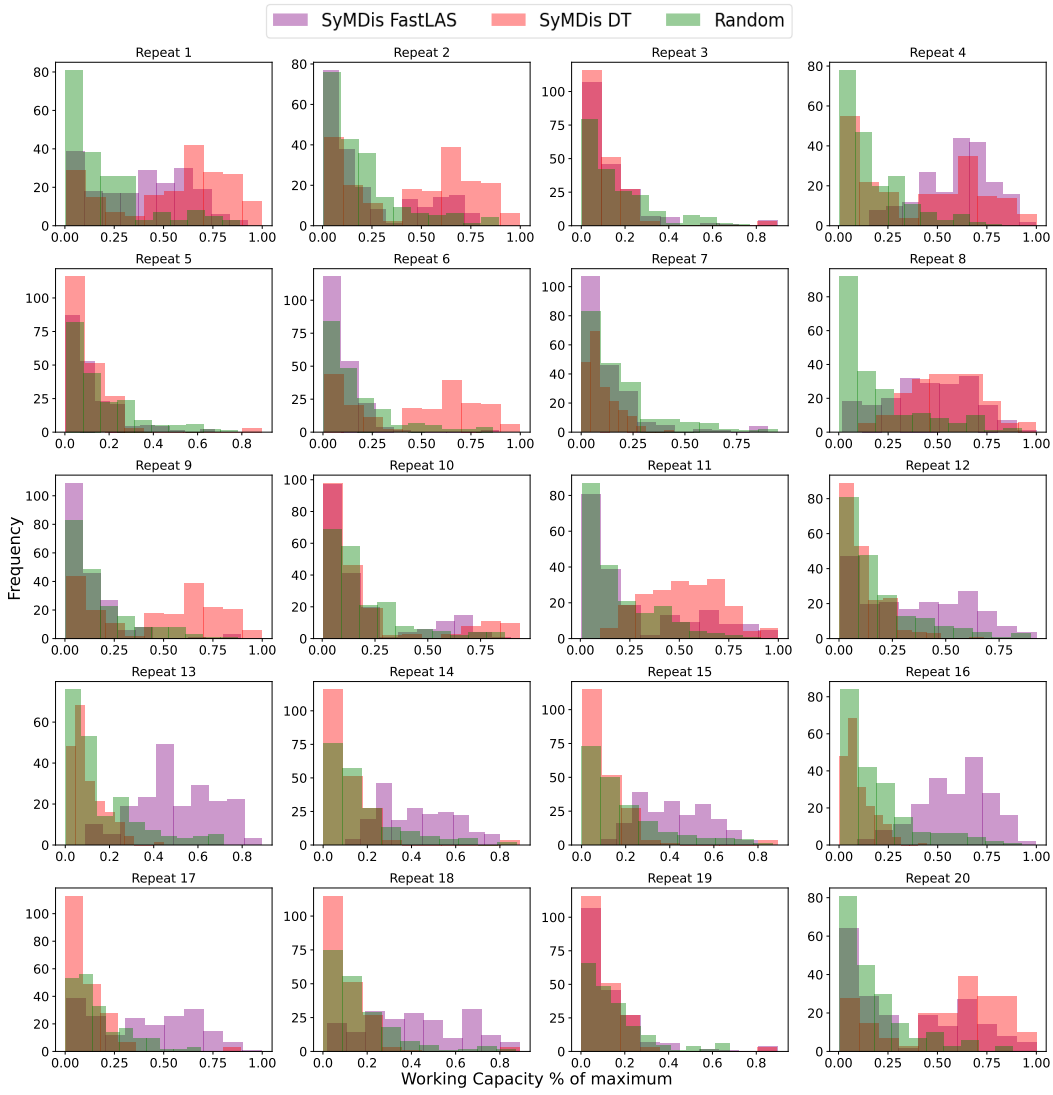


Figure 6: Source = ARABG, target = Core2019

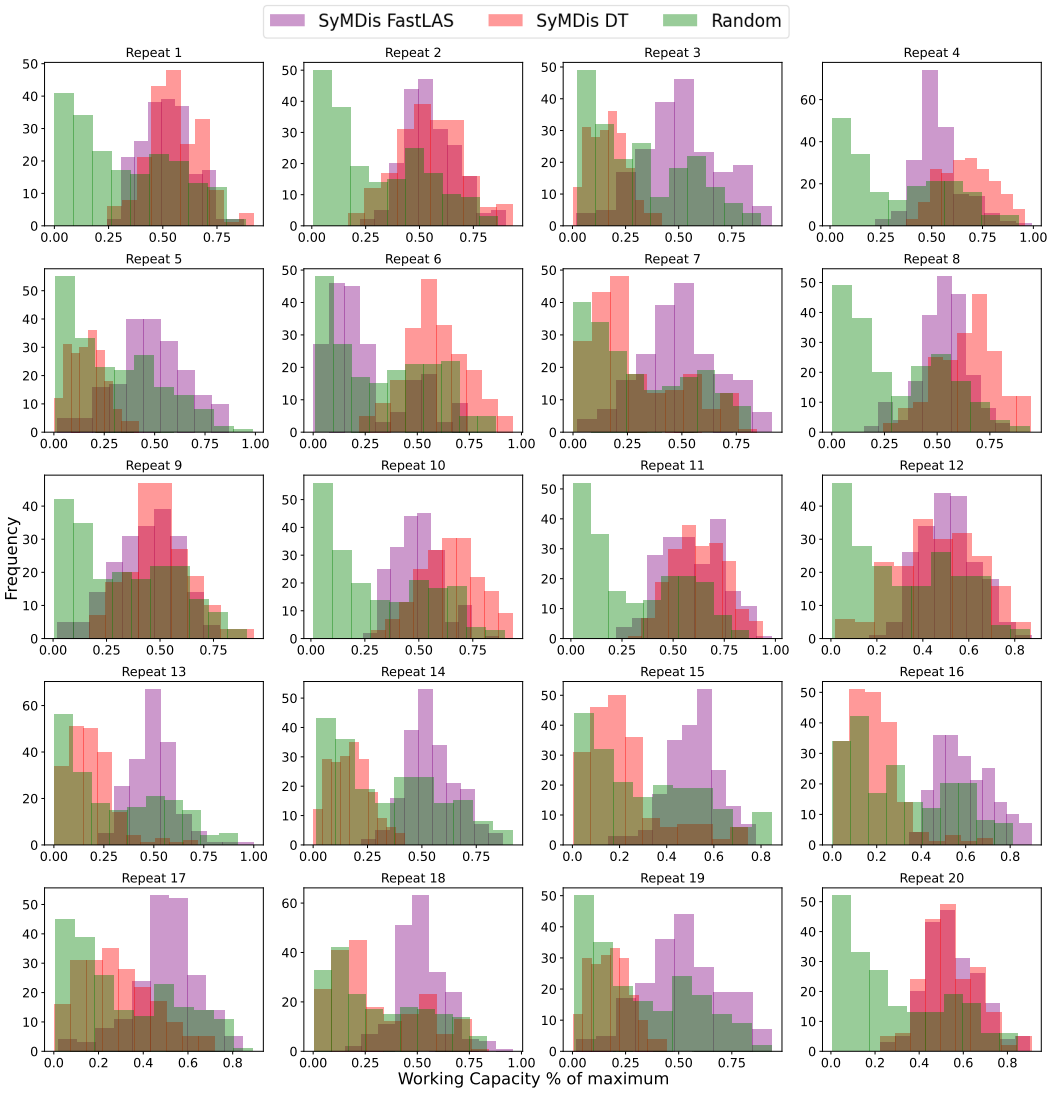


Figure 7: Source = ARABG, target = BW20K

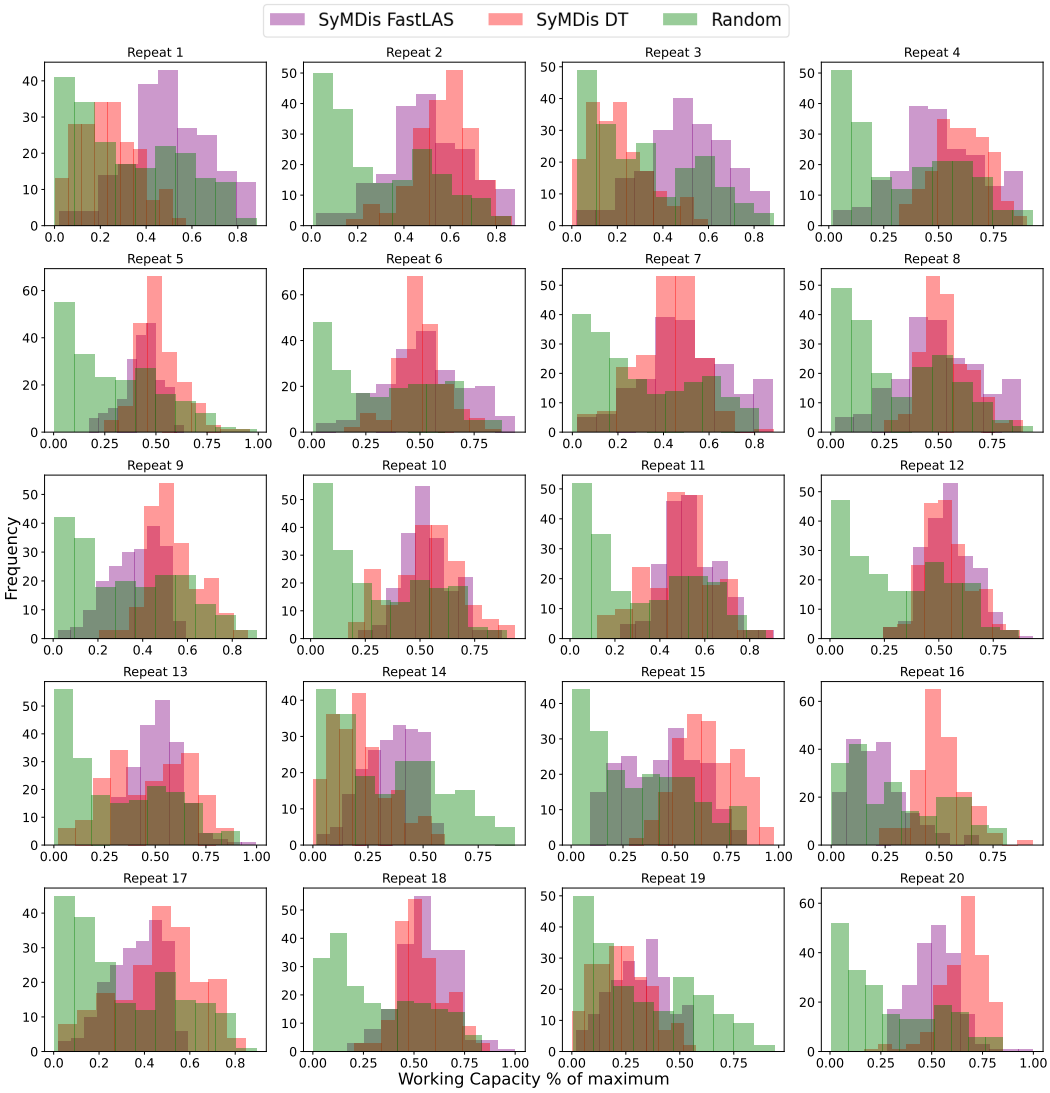


Figure 8: Source = CoRE2019, target = BW20K

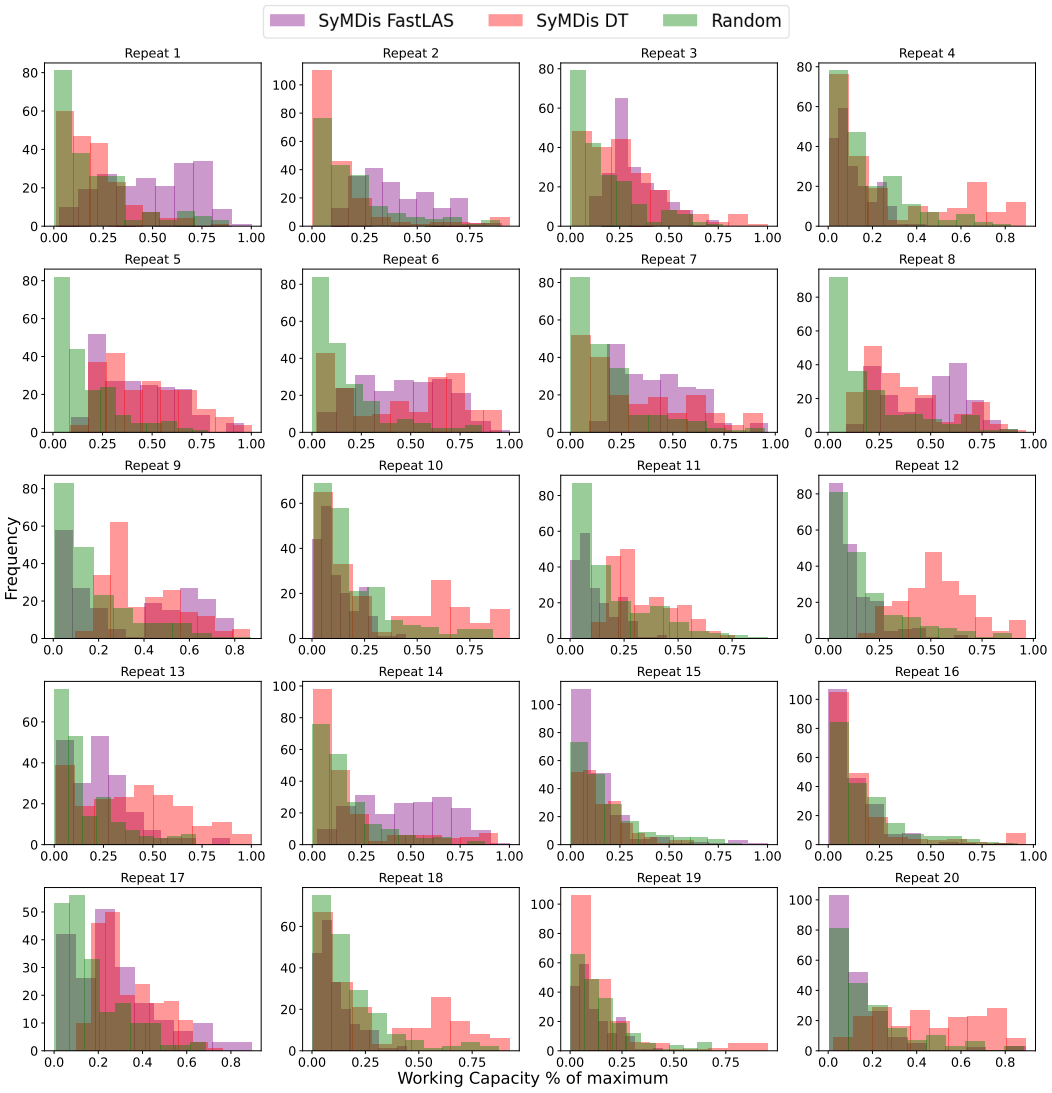


Figure 9: Source = BW20K, target = CoRE2019