Benchmarking knowledge transfer methods in de novo materials discovery

Witold Taisner, Anna Przybyłowska, Dariusz Brzezinski
Institute of Computing Science
Poznan University of Technology
ul. Piotrowo 2, 60-965 Poznan, Poland
{wtaisner,aprzybylowska,dbrzezinski}@cs.put.poznan.pl

Abstract

Generative AI holds immense promise for accelerating materials discovery. However, current AI models, often pre-trained on general chemical datasets, fail to generalize to materials science due to divergent chemical spaces and unique property constraints. In this work, we explore the transferability of knowledge from existing generative AI solutions to the design of porous carbon materials for supercapacitors. We evaluate the ability of open-source models to adhere to stringent material-specific requirements, such as high porosity, electron conductivity, and the potential of cyclization into solid frameworks. Furthermore, we investigate the optimal transfer learning strategies, assessing the trade-offs between retraining, fine-tuning, and reinforcement learning. Preliminary experiments with the REINVENT, MolMIM, and Mol-AIR generative models demonstrate that applying fine-tuning and reinforcement learning increases the generation of valid candidate molecules from at most 5.8% before transfer learning, up to 97.7% afterwards. Critically, our findings reveal that measures commonly used for benchmarking generative models in chemistry, such as validity, novelty, or uniqueness, are not aligned with the true goals of de novo molecule generation for materials science.

1 Introduction

Generative artificial intelligence (AI) has significantly accelerated de novo chemical design for tasks like inverse design [1–3], property prediction [4–6], and synthesis planning [7–9]. However, this success, largely demonstrated in drug discovery [10, 11], has been difficult to replicate in materials science [12]. This is because materials often exhibit specific structural properties, diverse bonding, and symmetry restraints, which are not aptly represented in standard chemical generative AI [13]. Moreover, many new materials are developed using unique chemical structure design principles and offer very limited experimental data for training predictive and generative models [14]. Consequently, many regions of the chemical space related to materials science remain unexplored, and existing methods fail to generalize beyond the subspace on which they were trained [15].

To address such shortcomings, transfer learning is often applied as an effective strategy for the adjustment of solutions from one domain to another [16, 17]. While transfer learning is an established machine learning technique, its application for transferring generative models from other chemical domains to materials science remains largely unexplored [18, 19], with existing works limited to fine-tuning LLMs for question-answering tasks [20].

In this work, we benchmark solutions known from drug design and evaluate the efficacy of transfer learning methods to repurpose them for the discovery of novel electrocatalytic materials. We design and conduct a comprehensive set of experiments tailored towards porous carbon structures (Figure 1A).

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI4Mat: AI for Accelerated Materials Design.

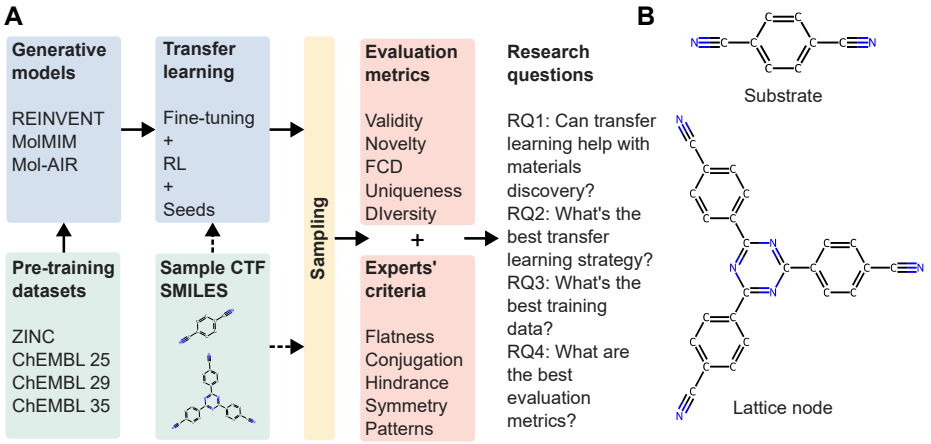

Figure 1: (**A**) Schematic overview of the study. (**B**) An example of a substrate representation (top) and a corresponding lattice node (bottom).

Specifically, we focus on three architectures (REINVENT [21], MolMIM [22], and Mol-AIR [23]) that either allow for pre-training, fine-tuning, or reinforcement learning, and generate molecules that adhere to multiple constraints. We evaluate the generative models in different configurations (e.g., with or without fine-tuning) to assess the trade-off between computational cost and model performance. In particular, we focus on generating candidate substrates that should exhibit several key properties important for supercapacitors, such as high porosity, large surface area, ability to create solid, cyclic structures, and high electron conductivity [24, 25]. Our contribution lies in constituting a practical and challenging testbed for the examined transfer learning methods, and analyzing the requirements and metrics that should be applied to generative AI benchmarks in materials science.

2 Methods

2.1 Covalent Triazine Frameworks

In our experiments, we will focus on Covalent Triazine Frameworks (CTFs), which are porous carbon materials that exhibit desirable properties from the perspective of supercapacitors [24]. CTFs are meshes created from a repeatable small-molecule building block, which we will refer to as *substrate*. However, certain properties of a candidate material, such as symmetry, can only be assessed by analyzing a larger mesh fragment. Given the above, in addition to substrates, we will consider another representation called *lattice node*, which is a part of a mesh, with the central "node" element, which connects the substrates. Lattice nodes are usually a product of a chemical reaction, in which substrates undergo pyrolysis, thus creating a large mesh. The lattice node representation used in our experiments is an expert-extracted meaningful part of the final mesh. An example substrate and lattice node are presented in Figure 1B.

2.2 Generative models

This study evaluated three generative models, which were selected based on the availability of open-source code and weights, use of SMILES/SELFIES representation, and their capability to generate molecules with specified properties.

REINVENT is a recurrent neural network (RNN) [26], more specifically a Gated Recurrent Unit (GRU) that is trained on canonical SMILES [27] from the ChEMBL 25 dataset. Importantly, it utilizes policy-based reinforcement learning (RL) [28] to optimize generated molecules towards desired properties. Notably, REINVENT uses Kullback–Leibler (KL) divergence to keep molecules generated by a reinforcement learning agent close to those generated by the prior [29].

MolMIM is a variational auto-encoder (VAE) [30] coupled with a mutual information machine (MIM) [31] that aims at creating a well-structured latent space of molecules, in which similar

molecules would constitute clusters. It was trained on a filtered subset of 730M instances from the ZINC-15 dataset using SMILES representation.

Mol-AIR is an RL framework for target-aware molecule generation. It uses the ChEMBL 29 dataset with SELFIES [32] representation, and focuses on combining the strengths of various reward strategies. It utilizes an LSTM-based [33] network for molecule generation, coupled with actor-critic layers and a random distillation network [34] to learn better rewards.

3 Experimental setup

3.1 Transfer learning

The generative models benchmarked in this study employ a range of training strategies: all can be pre-trained, MolMIM and REINVENT also support fine-tuning, and REINVENT and Mol-AIR can be optimized using reinforcement learning. The three architectures were evaluated using a grid of possible parametrizations, resulting in 30 settings, with each setting requiring the training of a model. For example, REINVENT with and without fine-tuning will be considered two separate settings. All information regarding data splits and processing is available in Appendix A.1. Moreover, both REINVENT and Mol-AIR were trained using reinforcement learning algorithms specified in respective publications, which requires a definition of a reward function to assess generated molecules. In this study, we utilized a set of expert-defined criteria to quantify whether a molecule is a suitable candidate or not, described in detail in Appendix A.2. Furthermore, if the reward function is too difficult for a reinforcement learning algorithm to comprehend, e.g., due to some criteria being binary, a policy collapse [35] can appear, meaning a model failed to train correctly and usually generates the same few molecules over again. All details regarding the implementation of reinforcement learning, mitigating the policy collapse, and other options, such as seed molecules, are further described in Appendix A.3. The model implementation details are available in Appendix A.4.

3.2 Dataset and task

The datasets used in this paper can be grouped into two categories, namely pre-training and fine-tuning data. Pre-training data consisted of publicly available, large databases of molecules, namely ZINC-15 [36] and ChEMBL [37]. Fine-tuning data was a manually curated set of CTFs, encoded as SMILES strings, from literature search and internal experiments. Consequently, a set of 103 CTFs was compiled by experts (both substrate and lattice node representations) and is available (see Appendix A.4). Given the stringent material requirements and limited data available, the goal is to apply transfer learning techniques such that sampled molecules will be novel, valid, and will exhibit desired chemical and structural properties.

3.3 Metrics

The efficacy of the generative models is evaluated through a set of six metrics: validity, uniqueness, internal diversity, Fréchet ChemNet Distance, novelty, and percentage of molecules passing custom filters.

Validity is expressed as a percent of valid molecules sampled. *Uniqueness* is the percentage of molecules without duplicates among the sampled data. Subsequently, *internal diversity* [38] is a distance metric, which evaluates how dissimilar generated molecules are from each other based on their Tanimoto similarity [39]. The *Fréchet ChemNet Distance* (FCD) [40] measures the distance between distributions of real and generated molecules. Finally, *novelty* measures how well a model discovers new molecules based on a reference dataset. In addition to the above five popular general-purpose metrics, we report the percentage of generated molecules that pass the expert-provided filters, including symmetry, flatness, patterns, conjugation, and steric hindrance (see Appendix A.2).

4 Results and discussion

To provide a comprehensive evaluation of the trained models, a sampling procedure was employed. For each model configuration, the generative process was executed in ten independent repetitions to compute the mean and standard deviation of the six evaluation metrics. Within a single repetition,

REINVENT and Mol-AIR models generated 10,000 molecules per stage, whereas MolMIM sampled 100 molecules for each instance from the training dataset. Table 3 in Appendix A.5 presents a detailed summary of the results.

Out of the six metrics used, the general-purpose measures showed very similar values for all configurations (Table 3). Novelty was always over 99%, while validity and uniqueness were both usually between 90-100%. Furthermore, internal diversity oscillated around 80-90% for the majority of experiments, whereas FCD is not easily interpretable and correlated with uniqueness and novelty (Figure 2) rather than providing additional insights. For the analyzed material structures (CTFs), the most important aspects include structural and electrochemical properties. These properties were encapsulated in the percentage of structures passing filters, therefore, we will focus on this measure while attempting to answer research questions posed in this study (Figure 1A, RQ1–RQ4). We note that in our experimental setup, we did not perform Density Functional Theory (DFT) calculations for the energy of a system, as such information was deemed irrelevant to chemists working on CTFs. We also note that we attempted to evaluate all three models on both substrates and lattice nodes. However, only REINVENT was capable of handling both representations within the allotted training time (either pre-training or RL training) of 72 hours on the NVIDIA A100 GPU used.

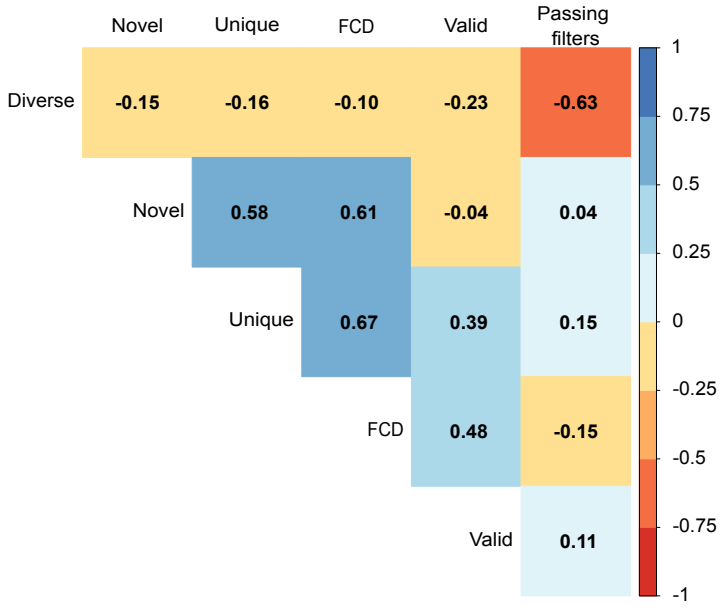

Figure 2: Pearson correlation between values of evaluation metrics.

Pre-training on ChEMBL 35 usually improved the percentage of candidate molecules passing filters, particularly for substrates (Figure 3). This is because ChEMBL 35 contains larger and more diverse molecules than its previous versions and the ZINC dataset. The ZINC dataset was explicitly curated for small molecules, thus, it is unsuitable for larger substrates and lattice nodes. Notably, pre-training itself does not significantly outperform vanilla versions while being the most computationally expensive method, to the point that Mol-AIR could not be pre-trained even on substrates in the allotted time. Furthermore, fine-tuning always improved the results, but at the cost of decreasing uniqueness. Interestingly, combined with pretraining REINVENT on ChEMBL35, fine-tuning (FT) alone was the best strategy for the lattice node representation. As for reinforcement learning (RL), its application always resulted in an increased percentage of molecules passing filters for substrates (Figure 3). On the other hand, providing seed molecules unexpectedly slightly deteriorated the results in all but one case (REINVENT (ChEMBL35) RL+seed), in which the increase was marginal. For lattice nodes,

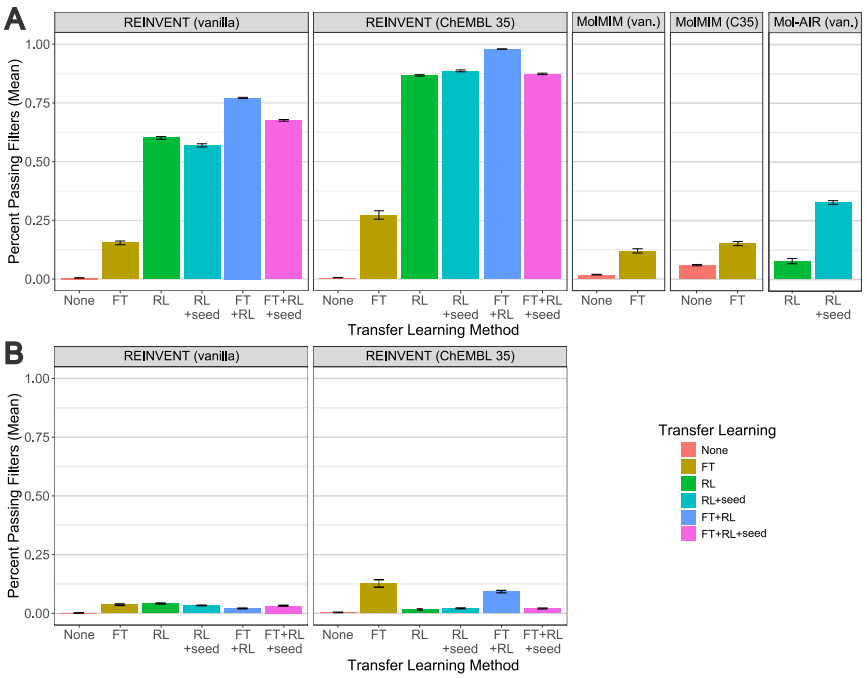

Figure 3: Percentage of molecules passing filters for (A) substrates and (B) lattice nodes. Notably, Mol-AIR, unlike other models, was not designed to sample on a pre-trained prior and requires training an RL agent; thus, there is no Mol-AIR entry for a non-transfer learning approach.

RL failed to outperform FT, and providing seed molecules similarly had almost no effect. Notably, as seen in Figure 2, the passing filters metric did not correlate with other measures except for internal diversity, with which it has a strong negative correlation. This negative correlation stems from the fact that the diversity measure evaluates exploration capabilities, whereas the percentage of passing filter focuses on the exploitation of a particular chemical subspace.

Given that transfer learning significantly improved the results for all models and both molecular representations, we can conclude that it constitutes a viable path for generative materials discovery (RQ1). As for the best transfer learning strategy (RQ2), reinforcement learning seems to be the most potent approach. However, training of an RL agent is highly dependent on the reward calculation and much more expensive than fine-tuning. Nevertheless, in our experiments, this was usually a justified cost. As for the most useful training data (RQ3), we show that fine-tuning on a small dataset is relatively cheap and improves results, whereas pre-training is not a must-have, but is also beneficial. Lastly, we see a gap in metrics used for evaluating generative models for materials science (RQ4). Common metrics for general-purpose molecular data can only indicate elementary problems with the training process, whereas the usefulness of the generated molecules has to be evaluated separately with well-defined material-specific criteria.

In summary, we have shown that most generative measures offer little knowledge about the true capabilities of models, and that even a small dataset can significantly boost the number of promising candidate molecules being generated. Apart from restating the main findings of this study, it is worth listing its limitations. The choice of models focused on popular open-source, open-weights models, and did not consider potentially newer closed models. Moreover, future studies could include more diverse reward definitions, and our ongoing work focuses on novel generative approaches for molecules with complex structural constraints, crucial for many materials.

Acknowledgments and Disclosure of Funding

This research was partly funded by the National Science Centre, Poland, grant number 2023/49/B/ST5/02403.

References

[1] Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, 2018.

[2] Jacqueline M. Cole. A Design-to-Device Pipeline for Data-Driven Materials Discovery. *Accounts of Chemical Research*, 53(3):599–610, 2020.

[3] Jiheng Fang, Ming Xie, Xingqun He, Jiming Zhang, Jieqiong Hu, Yongtai Chen, Youcai Yang, and Qinglin Jin. Machine learning accelerates the materials discovery. *Materials Today Communications*, 33:104900, 2022.

[4] Peter B. Jørgensen, Mikkel N. Schmidt, and Ole Winther. Deep Generative Models for Molecular Science. *Molecular Informatics*, 37(1-2), 2018.

[5] Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: Large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint*, arXiv:2010.09885, 2020.

[6] Sheng Wang, Yuzhi Guo, Yuhong Wang, Hongmao Sun, and Junzhou Huang. SMILES-BERT: Large Scale Unsupervised Pre-Training for Molecular Property Prediction. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 429–436, Niagara Falls NY USA, 2019. ACM.

[7] Shu-Wen Li, Li-Cheng Xu, Cheng Zhang, Shuo-Qing Zhang, and Xin Hong. Reaction performance prediction with an extrapolative and interpretable graph model based on chemical knowledge. *Nature Communications*, 14(1), 2023.

[8] Marwin H. S. Segler, Mike Preuss, and Mark P. Waller. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature*, 555(7698):604–610, 2018.

[9] Pascal Friederich, Artem Fediai, Simon Kaiser, Manuel Konrad, Nicole Jung, and Wolfgang Wenzel. Toward design of novel materials for organic electronics. *Advanced Materials*, 31(26):1808256, 2019.

[10] Kang Zhang, Xin Yang, Yifei Wang, Yunfang Yu, Niu Huang, Gen Li, Xiaokun Li, Joseph C. Wu, and Shengyong Yang. Artificial intelligence in drug development. *Nature Medicine*, 31(1):45–59, 2025.

[11] Yue Liu, Tianlu Zhao, Wangwei Ju, and Siqi Shi. Materials discovery and design using machine learning. *Journal of Materiomics*, 3(3):159–177, 2017.

[12] Jianchang Wu, Luca Torresi, ManMan Hu, Patrick Reiser, Jiyun Zhang, Juan S. Rocha-Ortiz, Luyao Wang, Zhiqiang Xie, Kaicheng Zhang, Byung-wook Park, Anastasia Barabash, Yicheng Zhao, Junsheng Luo, Yunuo Wang, Larry Lüer, Lin-Long Deng, Jens A. Hauch, Dirk M. Guldi, M. Eugenia Pérez-Ojeda, Sang Il Seok, Pascal Friederich, and Christoph J. Brabec. Inverse design workflow discovers hole-transport materials tailored for perovskite solar cells. *Science*, 386(6727):1256–1264, 2024.

[13] Edward O. Pyzer-Knapp, Matteo Manica, Peter Staar, Lucas Morin, Patrick Ruch, Teodoro Laino, John R. Smith, and Alessandro Curioni. Foundation models for materials discovery – current state and future directions. *Nature Computational Materials*, 11(1):1–10, 2025.

[14] Dylan M. Anstine and Olexandr Isayev. Generative Models as an Emerging Paradigm in the Chemical Sciences. *Journal of the American Chemical Society*, 145(16):8736–8750, 2023.

[15] Lane E. Schultz, Yiqi Wang, Ryan Jacobs, and Dane Morgan. A general approach for determining applicability domain of machine learning models. *Nature Computational Materials*, 11(1):95, 2025.

[16] An Chen, Zhilong Wang, Karl Luigi Loza Vidaurre, Yanqiang Han, Simin Ye, Kehao Tao, Shiwei Wang, Jing Gao, and Jinjin Li. Knowledge-reused transfer learning for molecular and materials science. *Journal of Energy Chemistry*, 98:149–168, 2024.

[17] Joseph Oloyede and Joshua Cena. Advancing Machine Learning through Knowledge Transfer: Techniques for Refinement and Domain Adjustment. *SSRN Electronic Journal*, 2024.

[18] Dhruv Menon and Raghavan Ranganathan. A Generative Approach to Materials Discovery, Design, and Optimization. *ACS Omega*, 7(30):25958–25973, 2022.

[19] Xiangru Tang, Howard Dai, Elizabeth Knight, Fang Wu, Yunyang Li, Tianxiao Li, and Mark Gerstein. A survey of generative AI for de novo drug design: new frontiers in molecule and protein generation. *Briefings in Bioinformatics*, 25(4):bbae338, 2024.

[20] Wei Lu, Rachel K. Luu, and Markus J. Buehler. Fine-tuning large language models for domain adaptation: exploration of training strategies, scaling, model merging and synergistic capabilities. *Nature Computational Materials*, 11, 2025.

[21] Hannes H. Loeffler, Jiazhen He, Alessandro Tibo, Jon Paul Janet, Alexey Voronov, Lewis H. Mervin, and Ola Engkvist. Reinvent 4: Modern AI–driven generative molecule design. *Journal of Cheminformatics*, 16(1):20, 2024.

[22] Danny Reidenbach, Micha Livne, Rajesh K. Ilango, Michelle Gill, and Johnny Israeli. Improving small molecule generation using mutual information machine. *arXiv preprint*, arXiv:2208.09016, 2022.

[23] Jinyeong Park, Jaegyoon Ahn, Jonghwan Choi, and Jibum Kim. Mol-air: Molecular reinforcement learning with adaptive intrinsic rewards for goal-directed molecular generation. *J. Chem. Inf. Model.*, 65(5):2283–2296, 2025.

[24] Ahmed G. Saad, Ahmed Emad-Eldeen, Wael Z. Tawfik, and Ahmed G. El-Deen. Data-driven machine learning approach for predicting the capacitance of graphene-based supercapacitor electrodes. *Journal of Energy Storage*, 55:105411, 2022.

[25] Akram Alabadi, Xinjia Yang, Zehua Dong, Zhen Li, and Bien Tan. Nitrogen-doped activated carbons derived from a co-polymer for high supercapacitor performance. *J. Mater. Chem. A*, 2(30):11697–11705, 2014.

[26] Robin M. Schmidt. Recurrent Neural Networks (RNNs): A gentle Introduction and Overview. *arXiv preprint*, arXiv:1912.05911, 2019.

[27] David Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988.

[28] Zhen "Leo" Liu. Policy-Based Reinforcement Learning. In *Artificial Intelligence for Engineers: Basics and Implementations*, pages 357–378. Springer, Cham, 2025.

[29] Raj Ghugare, Santiago Miret, Adriana Hugessen, Mariano Phielipp, and Glen Berseth. Searching for High-Value Molecules Using Reinforcement Learning and Transformers, October 2023. arXiv:2310.02902 [cs].

[30] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[31] Micha Livne, Kevin Swersky, and David J. Fleet. MIM: mutual information machine. *arXiv preprint*, arXiv:1910.03175, 2019.

[32] Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alán Aspuru-Guzik. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Mach. Learn. Sci. Technol.*, 1(4):45024, 2020.

[33] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[34] Yuri Burda, Harrison Edwards, Amos J. Storkey, and Oleg Klimov. Exploration by random network distillation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[35] Shibhansh Dohare, Qingfeng Lan, and A. Rupam Mahmood. Overcoming Policy Collapse in Deep Reinforcement Learning. May 2023.

[36] Teague Sterling and John J. Irwin. ZINC 15 – Ligand Discovery for Everyone. *Journal of Chemical Information and Modeling*, 55(11):2324–2337, 2015.

[37] Barbara Zdrazil, Eloy Felix, Fiona Hunter, Emma J Manners, James Blackshaw, Sybilla Corbett, Marleen de Veij, Harris Ioannidis, David Mendez Lopez, Juan F Mosquera, Maria Paula Magarinos, Nicolas Bosc, Ricardo Arcila, Tevfik Kizilören, Anna Gaulton, A Patrícia Bento, Melissa F Adasme, Peter Monecke, Gregory A Landrum, and Andrew R Leach. The ChEMBL Database in 2023: a drug discovery platform spanning multiple bioactivity data types and time periods. *Nucleic Acids Research*, 52(D1):D1180–D1192, 2024.

[38] Mostapha Benhenda. Chemgan challenge for drug discovery: can AI reproduce natural chemical diversity? *CoRR*, abs/1708.08227, 2017.

[39] Dávid Bajusz, Anita Rácz, and Károly Héberger. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *J. Cheminformatics*, 7:20:1–20:13, 2015.

[40] Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Günter Klambauer. Fréchet chemnet distance: A metric for generative models for molecules in drug discovery. *J. Chem. Inf. Model.*, 58(9):1736–1741, 2018.

[41] Gil Alon, Yuval Ben-Haim, and Inbal Tuvi-Arad. Continuous symmetry and chirality measures: approximate algorithms for large molecular structures. *Journal of Cheminformatics*, 15, 11 2023.

[42] Jason Ansel, Edward Z. Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, C. K. Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Shunting Zhang, Michael Suo, Phil Tillet, Xu Zhao, Eikan Wang, Keren Zhou, Richard Zou, Xiaodong Wang, Ajit Mathews, William Wen, Gregory Chanan, Peng Wu, and Soumith Chintala. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In Rajiv Gupta, Nael B. Abu-Ghazaleh, Madan Musuvathi, and Dan Tsafrir, editors, *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2, ASPLOS 2024, La Jolla, CA, USA, 27 April 2024- 1 May 2024*, pages 929–947. ACM, 2024.

[43] RDKit community. Rdkit: Open-source cheminformatics. `https://www.rdkit.org`, 2025. Version 2025.3.5.

A Appendix

A.1 Data processing

ChEMBL 35 was used for pre-training of models in respective experiments, with 70%/20%/10% train/validation/test splits. The custom dataset was split into train and validation sets with an 80% and 20% ratio, respectively. The split was performed on both substrates and lattice nodes, such that a substrate and corresponding lattice node are both in either the train or validation set.

Furthermore, each of the generative architectures imposes different criteria for molecules it accepts and generates, based on the pre-training data. The most important aspects include the maximum number of elements allowed, as expressed in the number of tokens in the SMILES or SELFIES string, and the absence of certain tokens from the models' vocabulary. Consequently, not all of the molecules from the custom dataset could be used for fine-tuning or as a seed in reinforcement learning with the exact amounts specified in the Table 1. Notably, pre-training on ChEMBL35 always increases the number of custom molecules the model accepts; however, no configuration was capable of processing all the experts' data.

Table 1: Number of custom molecules accepted by models in respective configurations.

Representation	Model	#training	#testing
Substrate	REINVENT (vanilla)	65	19
	REINVENT (ChEMBL 35)	72	20
	MolMIM (vanilla)	72	18
	MolMIM (ChEMBL 35)	80	20
	Mol-AIR (vanilla)	45	12
Lattice node	REINVENT (vanilla)	22	6
	REINVENT (ChEMBL 35)	71	9

A.2 Experts' criteria

Candidate molecules for supercapacitors must conform to multiple constraints to be considered promising. In this study, chemical experts have defined a set of properties that can be applied to either substrate, lattice node, or both representations.

A.2.1 Patterns

Certain patterns present in the substrate representation allow for cyclization with other substrates, which leads to the creation of a mesh in the pyrolysis process. Consequently, a set of important chemical patterns was defined by experts and implemented using RDKit and SMARTS.

In reinforcement learning reward calculation, the presence of any of the defined patterns is evaluated as 1, while absence is scored as 0. What follows is that a molecule passes this criterion only if any specified pattern is present.

A.2.2 Symmetry

Symmetry is an important aspect of a mesh, as it increases the stability of the resulting structure. For this task, we approximate symmetry concerning C2, C3, and C4 groups using Continuous Symmetry Measure (CSM) [41]. CSM returns a measure of distance required for a molecule to adjust to a certain symmetry group, thus, the lower the better.

In reinforcement learning reward calculation, the CSM of the three symmetry groups of interest was calculated for each molecule, and the lowest value of the three was picked. For reward calculation, we map values into the continuous 0–1 range by using a scoring function that assigns a score between 0 and 1. A maximum score of 1.0 was assigned to values falling within an optimal range of CSM (0–5), and for values outside this range, the score was penalized with an exponential decay proportional to the distance from the optimal range boundaries. The optimal range was defined based on the CSM values of the experts' data, and a molecule passes this criterion if its CSM value falls within the aforementioned interval.

A.2.3 Flatness

Flatness is important as flat substrates or meshes can stack onto each other and create a stable structure. We evaluate flatness by fitting a plane onto a sampled molecule's conformation using a linear regression model and calculating the root mean squared error (RMSE) between atoms' coordinates and a plane as a measure of flatness. Similarly to CSM values, we set a threshold at 6.0 based on the data, and then scale values to the 0-1 range using the same procedure. Similarly to CSM, a molecule passes this filter if its flatness is below a 6.0 threshold.

A.2.4 Conjugation

Conjugation estimates whether a substrate or part of a mesh is capable of transferring electricity. We check whether any path exists between any pair of nitrogen atoms, such that no two consecutive single bonds are present, and if they are, whether an atom in the middle has a pair of lone electrons.

Consequently, we assign a score of 1 and allow it to pass through the filter if a molecule is conjugated, and the opposite otherwise.

A.2.5 Steric hindrance

Steric hindrance is a process in which the proximity of atoms in 3D space hinders chemical reactions. We evaluate steric hindrance by calculating the most probable conformation using RDKit and checking whether any two nitrogen atoms are closer than 4.1 Å. If that is the case, the molecule does not pass the filter and receives a score of 0. In the opposite scenario, a score of 1 is assigned, and a molecule is allowed through the filtering process.

A.3 Reinforcement learning setup

A.3.1 Reward function

For each representation, each criterion (as described in Appendix A.2) returns a score between 0 and 1, and these values are then averaged. This allows for using the RL algorithms if they support only single-objective calculations. We differentiate the criteria used between substrate and lattice node representations. Substrates are evaluated on patterns, flatness, conjugation, and steric hindrance. Lattice nodes are evaluated on symmetry, flatness, conjugation, and steric hindrance. Moreover, both representations are also evaluated using similarity to reference (training) molecules.

Importantly, if a criterion requires a given molecule's conformation (e.g., to compute its flatness), and no valid conformation could be obtained using RDKit, the criterion score was set to 0, and such a molecule did not pass the respective filter.

A.3.2 Seed molecules

The evaluation of a reinforcement learning agent's capabilities can be conducted in two distinct variants. The first involves initializing the agent with a dataset of expert instances to facilitate the learning process, which we will call "seed molecules". In contrast, the second paradigm assesses the agent's performance when trained without initialization with any pre-existing data. Consequently, we evaluate the RL capabilities of both REINVENT and Mol-AIR with and without providing them with seed molecules.

A.3.3 Policy collapse

In four of the trained configurations, specifically for REINVENT (ChEMBL 35) RL and FT+RL, on both lattice node and substrate representations, a policy collapse was observed in the initial experiments, which was indicated by uniqueness close to 0 and the percentage of molecules passing filters being exactly 0. Furthermore, investigation of the training process showed that over time, models focused on exploiting a narrow group of the same molecules more and more instead of exploring the chemical space. To mitigate this effect, parameters of REINVENT's reinforcement learning algorithm were adjusted to penalize exploitation and reward exploration more, with exact values described in Appendix A.4.

A.4 Model implementation details

All computations were conducted on a virtual machine with Ubuntu 22.04.5 LTS, AMD Epyc 7713 CPU, 1000 GB of RAM, and NVIDIA A100 GPU with 80 GB of VRAM. If not stated otherwise, Python 3.12.0 was used along with PyTorch 2.5.1 [42], selfies 2.2.0, and RDKit 2025.3.5 [43].

REINVENT's implementation was taken from the official repository under version 4.6. Consequently, the Python environment used was the same as proposed in the repository. Furthermore, MolMIM was evaluated using NVIDIA BioNeMo Framework, version 1.10.1, with the provided environment. Lastly, Mol-AIR's implementation was also based on the official repository, with the Python environment described above.

The code, along with the data and the models, is available at the `https://github.com/wtaisner/battery-rangers`.

All experiments were run with respective parameters, depending on the configuration, as specified in Table 2. For each trained model, sampling was repeated 10 times without a fixed random seed, such that the mean and standard deviation of metrics could be reported. REINVENT and Mol-AIR were tasked with sampling 10,000 molecules, whereas MolMIM sampled 100 molecules for each training instance, as described in Table 1, thus, the number of generated substrates was 7,200 and 8,000 in vanilla and ChEMBL35 setups, respectively.

Table 2: Knowledge transfer methods' parameters. If not specified otherwise, parameters' values (e.g., learning rate) default to those proposed by the authors in their respective implementations. In both pre-training and fine-tuning, models with the lowest validation loss were picked. At each stage of transfer learning, the random seed was fixed for reproducibility. Entry "Reinforcement learning (policy collapse)" was only applied to cases described in Appendix A.3.3.

Model	Knowledge transfer method	Parameters
REINVENT	Pre-training	500 epochs with batch size of 4096.
	Fine-tuning	100 epochs with batch size of 16.
	Reinforcement learning	A minimum of 50 and a maximum of 200 steps. If seed molecules were provided, then 10 samples were randomly picked in each step, with a total of 100 being held in memory.
	Reinforcement learning (policy collapse)	A minimum of 50 and a maximum of 200 steps. Learning rate of 0.00005 and sigma parameter of 256. The diversity filter was changed from IdenticalMurckoScaffold to PenalizeSameSmiles.
MolMIM	Pre-training	10 epochs with a batch size of 256, maximum sequence length increased to 256.
	Fine-tuning	200 steps with batch size determined automatically.
Mol-AIR	Reinforcement learning	250000 training steps with batch size of 64. The count-based intrinsic reward coefficient was set to 0.2.

A.5 Experimental results

Table 3 contains details from all experiments. Notably, results for lattice node representation are only available for REINVENT in two variants, due to the large computational requirements. To reiterate, both REINVENT and Mol-AIR can sample de novo, while MolMIM requires a seed molecule, for which it decodes similar molecules. All metrics except for the percentage of passing filters score very similarly across all configurations and representations, with Mol-AIR having a 100% validity by design through the SELFIES molecule representation.

Table 3: Mean values and standard deviations (in parentheses, in units of the last significant digit of the mean value) of performance metrics calculated for the three analyzed generative models with different transfer learning.

Repr.	Model	Transfer learning	Valid ↑ [%]	Unique ↑ [%]	Diverse ↑ [%]	FCD ↓	Novel ↑ [%]	Passing filters ↑ [%]
Substrate	REINVENT (vanilla)	None	98.3(2)	100.0(0)	87.8(0)	51.0(1)	100.0(0)	0.5(0)
		FT	93.6(2)	25.1(3)	85.2(1)	25.0(3)	99.9(0)	15.4(8)
		RL	97.5(2)	99.8(1)	82.9(1)	46.0(1)	100.0(0)	60.1(6)
		RL+seed	97.8(1)	99.5(1)	83.5(1)	44.3(1)	100.0(0)	56.9(7)
		FT+RL	97.1(2)	90.5(1)	79.2(1)	38.1(1)	100.0(0)	77.1(3)
		FT+RL+seed	96.8(2)	95.0(3)	80.2(1)	35.3(2)	100.0(0)	67.6(4)
	REINVENT (ChEMBL 35)	None	93.2(2)	100.0(0)	87.7(0)	51.2(2)	100.0(0)	0.4(1)
		FT	86.3(5)	3.8(2)	84.6(2)	17.5(3)	99.9(1)	27.1(18)
		RL	99.1(1)	84.0(4)	69.4(1)	44.6(1)	100.0(0)	86.5(3)
		RL+seed	91.6(3)	92.0(3)	78.9(1)	42.9(1)	100.0(0)	88.4(4)
		FT+RL	99.0(1)	82.2(4)	70.7(1)	41.4(1)	100.0(0)	97.7(1)
		FT+RL+seed	93.7(2)	78.2(4)	76.4(1)	37.1(1)	100.0(0)	87.1(3)
	MolMIM† (van.)	None	87.9(5)	98.7(1)	88.2(0)	43.3(2)	100.0(0)	1.7(1)
		FT	69.3(3)	40.3(4)	87.0(1)	26.1(2)	100.0(0)	11.8(9)
	MolMIM† (C35)	None	92.1(4)	93.8(3)	87.2(0)	33.9(2)	100.0(0)	5.8(3)
		FT	96.9(2)	27.2(2)	84.3(1)	21.4(1)	99.9(0)	15.0(9)
	MolAIR (van.)	RL	100.0(0)	0.3(0)	90.1(10)	43.1(6)	100.0(0)	7.6(11)
		RL+seed	100.0(0)	30.9(5)	84.6(1)	36.1(2)	99.9(0)	32.6(8)
Lattice node	REINVENT (vanilla)	None	98.3(2)	100.0(0)	87.8(0)	58.1(1)	100.0(0)	0.2(4)
		FT	96.9(2)	32.2(5)	87.3(1)	35.9(3)	100.0(0)	3.7(33)
		RL	98.2(1)	99.5(1)	84.4(0)	49.0(1)	100.0(0)	2.08(14)
		RL+seed	97.6(1)	99.4(1)	85.5(0)	47.9(1)	100.0(0)	3.37(8)
		FT+RL	98.0(1)	96.6(1)	80.1(1)	41.9(1)	100.0(0)	2.1(14)
		FT+RL+seed	96.3(2)	97.8(2)	82.4(1)	41.9(1)	100.0(0)	3.2(18)
	REINVENT (ChEMBL 35)	None	93.2(2)	100.0(0)	87.7(0)	58.5(2)	100.0(0)	0.2(32)
		FT	29.6(4)	6.7(5)	86.2(3)	23.2(3)	100.0(0)	12.5(16)
		RL	99.0(1)	45.3(4)	77.5(1)	48.7(1)	100.0(0)	1.4(1)
		RL+seed	96.4(3)	89.3(4)	83.8(1)	46.0(1)	100.0(0)	1.9(14)
		FT+RL	98.9(1)	55.7(7)	75.4(1)	50.8(1)	100.0(0)	7.9(5)
		FT+RL+seed	96.3(2)	76.7(3)	75.0(1)	50.0(1)	100.0(0)	1.9(16)

† Due to its design, MolMIM cannot perform de novo sampling and requires a seed molecule to perform a search in latent space. Consequently, for each training instance, MolMIM samples 100 similar molecules.