CHEMICAL STRUCTURE ELUCIDATION FROM TANDEM MASS SPECTROMETRY AND CHEMICAL FORMULA

A novel attempt at a Reinforcement Learning approach to De Novo Chemical Structure Elucidation from Mass Spectrometry and Chemical Formula.

Yu Simu[1] and Shen Bingquan[2]

[1] NUS High School of Math and Science, 20 Clementi Ave 1, Singapore 129957
[2] DSO National Laboratories, 12 Science Park Drive, Singapore 118225

Abstract. This paper presents a novel Reinforcement Learning approach towards De Novo Chemical Structure Elucidation, reducing the generative task to a simple regressive task using a hybrid between A* search and beam search, while additionally constraining the molecule, generated sequentially, on graph connectivity and chemical formula, on top of valency as explored in other works. Graph convolution was also used in generating graph-level embeddings to take into account the permutation-invariant property of graphs, as well as on fragmentation trees as a representation of the mass spectrum. We find that valency and connectivity restrictions have significant contributions in structure generation towards the end of the generation, and that this approach appears to have higher relevance in tasks involving smaller molecules.

Keywords: De Novo Chemical Structure Elucidation, Reinforcement Learning, Search

1 Introduction

Chemical Structure Elucidation (CSE) is a task that is known to be exceedingly difficult to complete by hand, thus, the potential use of computers to aid in this task has been explored[1,2]. However, this task also does not have convenient algorithmic solutions, as MS/MS spectra contain no information on structures, only molecular masses of fragments, while considering all permutations will be unfeasible with large molecules[3] due to too much time being taken. Thus, to fully complete this task computationally, the use of some form of Artificial Intelligence has been explored.

However, de Novo CSE is a generative task with very little training data, with an estimated 60,000 molecules available after combining all large datasets available[4], while other generative tasks are typically trained with >500,000 structures.[5,6,7] Therefore, considering the higher performance of highly algorithmic or statistical approaches as compared to neural approaches with low amounts of training data,[8]the former is more likely successful for this task. Thus, solutions to this problem, for the foreseeable future with similar datasets, will likely require the use of algorithms to aid it, reducing the task to a simpler task, to achieve good results.

Similar molecule generation tasks[5,6] commonly use textual representations of chemicals[9] to generate a molecule with a structure, allowing the generation of molecules using Recurrent Neural Networks autoregressively, as one-shot generation models are extremely limited. Despite their ability to generate molecules of any required size given the scenario due to its sequential nature, molecules are high-dimensional structures which have been unable to accurately be represented textually[10], despite many extensions, and fail to be permutation invariant. Therefore, it would likely be better to have a model that represents molecules as graphs.

Currently trending methods like Graph Diffusion attempt to address the issue of such methods not being permutation invariant[11], as a direct representation of the molecule. These methods perform a process similar to diffusion in images, but on graphs and their adjacency matrices instead[12,13,14]. However, as mentioned in those papers, these methods suffer from scalability difficulties, as their model used in diffusion given graph node features or adjacen-

cy matrices has a fixed size, and despite them using zero padding to allow for the generation of molecules below the maximum size, they still need sufficient model size and training data for molecules of large sizes for their model to work in those cases, and such data is hard to come by and can cause poor model performance in those scenarios.

Current methods also fail to make use of the specific additional constraints this specific task provides, which are molecular formula being able to be inferred from the mass spectrum[15], and the rules of chemical molecules such as valency and connectivity. Most methods, such as those mentioned above, do not enforce the above rules, which do not guarantee chemical validity of generated molecules, while some others that do[16,17] fail to let it be constrained by a molecular formula, especially if considering hydrogen atoms.

Therefore, some approaches perform spectra database search.[18,19,20,21,22] However those have very limited use, especially considering that there are few molecules with their spectra recorded in databases. Other methods attempt to extend to other databases, but they fail to perform the generative task, only searching, being inherently limited.

Therefore, this paper intends to explore a reinforcement learning solution to this that mitigates the aforementioned issues. The reinforcement learning solution would only approximate the value of next states from a state, reducing the task to a regressive score predictor, providing many states for training the predictor. The action space will also be constrained to enforce valency and connectivity rules, as well as a target molecular formula. The molecule will be represented as a graph itself, and graph convolution for graph-level embeddings will be used on it to get the next state, while its generation will be sequential. These are detailed in section 2.

2 Materials and methods

2.1 Enforcing stability of molecules

As mentioned in the introduction, we enforce molecule stability in this system, making use of the knowledge that this is a molecular generation task. The initial state of this reinforcement learning environment contains the target, the mass spectrum, and all the atoms in the molecular formula (excluding Hydrogen), with no bonds between these atoms (except aromatic bond, see end of 2.1), and each having some amount of hydrogens bonded to it that reflects its usual amount of bonds (e.g. C will have 4 Hs, N will have 3, while P will have 5). The actions involve selecting two different atoms from the molecule, and forming a bond between them (single, double, or triple), replacing Hs. The action space is restricted based on not only valency, but also considers graph connectivity. The following is the process for determining if an action is valid, after considering the valency:

- Let the number of connected components be n. Ensure that, at the end of the action, the number of remaining Hs in the entire graph $\leq 2n - 2$. This makes sure that it is theoretically possible to connect the graph.
- Check that the sum of the number of remaining Hs in each component is at least 2 for all but 2 of the components, and at least 1 for the last 2 components (to make sure it is possible to make bonds to connect the components)

The above is determined using a modified Union-Find Data Structure, which also stores the total number of remaining Hs in each component. Though the above rules assume that the molecule has no formal charge, in the context of organic compounds, a very large majority of

compounds large enough to not be trivially identified will fulfil those constraints. In testing, there were zero scenarios of the above resulting in invalid states.

In addition, a slight compromise, made to ensure aromatic bonds are only found in rings, is to make there no actions to create aromatic bonds, and such rings may only exist if specified in the starting state. There are a finite number of structures considered aromatic, and it therefore follows that it could be better to enforce that finite set, rather than allow the model to make mistakes creating aromatic bonds, or learn not to create them in fear of generating invalid molecules.

2.2 Maintaining permutation invariance with scalability

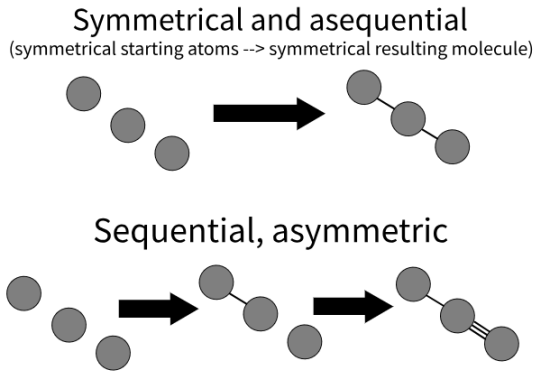

Figure 1. A visualization of the necessity of asymmetric generation of molecules.

As mentioned in the introduction, permutation invariance in graphs is a property that is theoretically helpful, yet graph diffusion methods fail to scale to larger sizes than trained. Instead, this paper proposes that permutation invariance in graphs is only important in extracting its features, while its generation can be sequential. The asymmetries caused by sequential generation are not seen as an issue, but even necessary for effective generation of molecules. For example, if the target molecule has its SMILES representation C#CC, fully symmetrical (and non-sequential) generation would cause both bonds to be the same, as the environment is also fully symmetrical.

2.3 The challenge of the low volume of training data available

Also, to combat the lack of training data for this task, in this reinforcement learning environment, the only model used would be one that predicts the ‘value’ of possible resulting states after taking actions from a state. States that lead to the target state should have higher predicted values, while others will have a low value.

This ‘value’ is calculated by running the partially complete molecule through a Conditional Graph Convolutional Network that considers edge features by running them through a multilayer perceptron to get a matrix to multiply the incoming node features by, taking inspiration from that used in another paper titled “Neural Message Passing for Quantum Chemistry”[23] The node features used were the degree, remaining valence, and a one-hot encoding of the atom type, while the edge features were the bond energy (set to 0 temporarily due to the lack of a prediction mechanism), and a one-hot encoding of the bond type (single, double, triple, aromatic) The conditional features in the aforementioned network will be the output from another network, explained in 2.4.

This causes each individual molecule to provide an amount of states as training data that is at least exponential respective to the amount of atoms in them, due to the extremely large state space in this task. Taking inspiration from anomaly detection[24], where the training data is largely non-anomalous due to the prohibitive cost of having anomalous training data, we take anomalies as states that are not in the training data, and vice versa, allowing for effective learning despite only one type of training data (correct states) being available. Then, we can

simply train the model on states which can be represented by combinations of the bonds being removed. The loss used will be the same as that used in[24].

2.4 The challenge of mass spectrum noise

Another challenge with this task, not mentioned in the introduction, is that slight errors in mass spectra, peak shifts, are common, and training a network to be resistant to such errors requires many test cases. Also, at each peak, there may be multiple possible molecular fragments that match the mass spectrum.

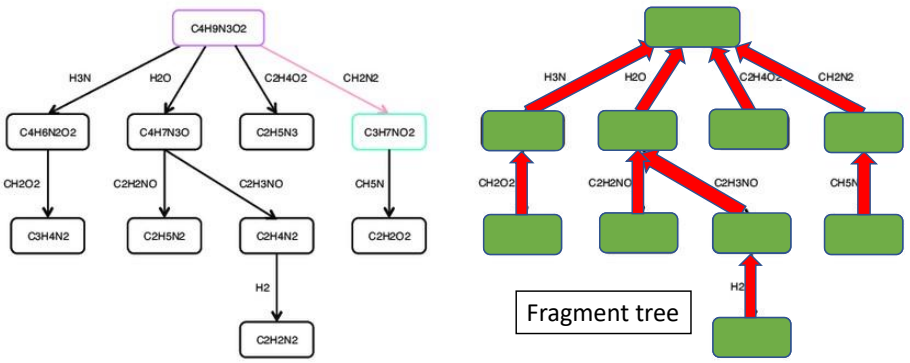

Fig. 2. An illustration of graph convolution's effect on the root node of a fragmentation tree. The figure on the left is taken from[27]

Fragmentation Trees[25] have been developed and used in the SIRIUS software[26], as a method of viewing the fragmentation process. This gives the molecular formula of each fragment, which has a lot of potential as a representation of the mass spectrum, as it is created from a mass spectrum itself. However, to the authors' knowledge, this, thus far, has only been used in the same way as it had been used in CSI:FingerID and IOKR[26,27], where they compared fragmentation trees instead of mass spectra when reconstructing a molecular fingerprint. This is likely because fragmentation trees do not have a fixed size, thus cannot be used in a model easily.

In this paper, we intend to make more use of this, running the mass spectrum through a graph convolution network, with directed edges pointing towards the root, and taking the final node features of the root node as an output of this model, an embedding meant to represent the tree. The fragments are represented with 7-length vectors, the first 6 values being the number of each atom (CHNOPS) in the fragment, and the last being the molecular mass of the fragment.

2.5 Overview of method

Firstly, we pretrain a Fragmentation Tree Graph Convolutional Network (FTree GCN), ensuring that its embedding contains relevant information by having a predictor make it predict the original mass spectrum back. We test this by getting the mean squared error of the value for relative intensity at each m/z ratio bucket range. We also visualize the mass spectra in some of the cases, for comparison purposes.

Then, we train a Conditional Graph Convolutional Network to take in a partially built molecule, and the embedding of the mass spectrum given by the FTree GCN as a conditioning feature, to predict low scores for correct states and high scores for wrong states. This is done by choosing combinations of the bonds of the target molecule to remove, acting as intermediate states. This will use anomaly detection concepts seen in[24], as mentioned in section 2.3.

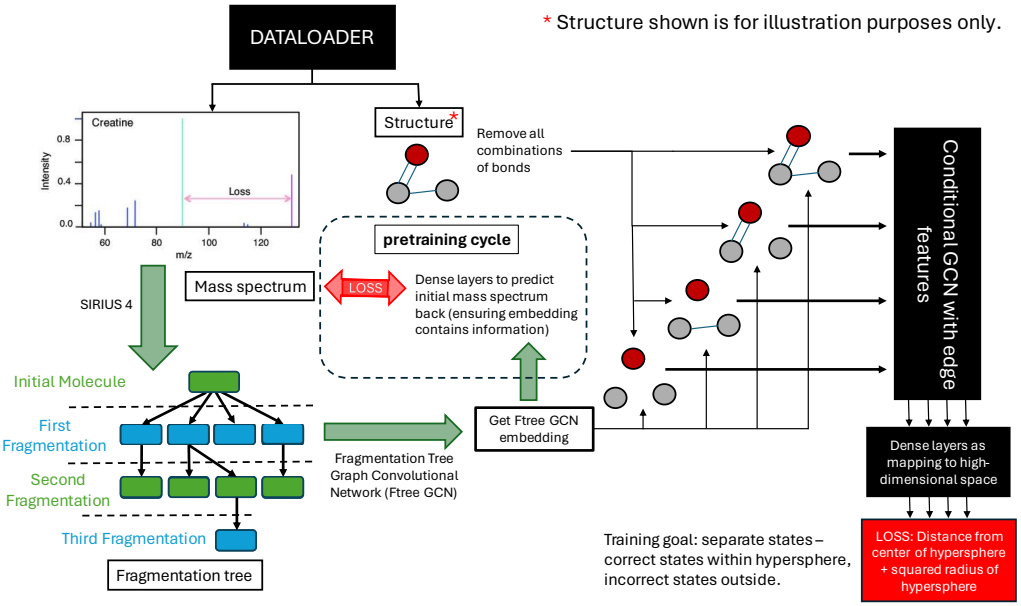

Fig. 3. A visualization of the training process of our models. The seemingly peculiar training targets are due to adapting anomaly detection methods[24] for this purpose, detailed in section 2.3.

This will be tested in two ways. Firstly, sample states, both correct and wrong, will be taken, and their scores given by the model will be recorded, then plotted on a histogram, showing the separability of the two by the model. Then, a variation of the A* search algorithm will be used to try searching for the correct target state given an initial state (with aromatic bonds already present). The heuristic used in this algorithm will be:

$$(1 - score_{anomaly}) * (H_t/2\,) \; - \; H_{left}$$

Where $score_{anomaly}$ is the anomaly score by the model, H_t is the total number of hydrogens in the initial state subtracted by the target state, and H_{left} is the number of hydrogens in the current state subtracted by the target state.

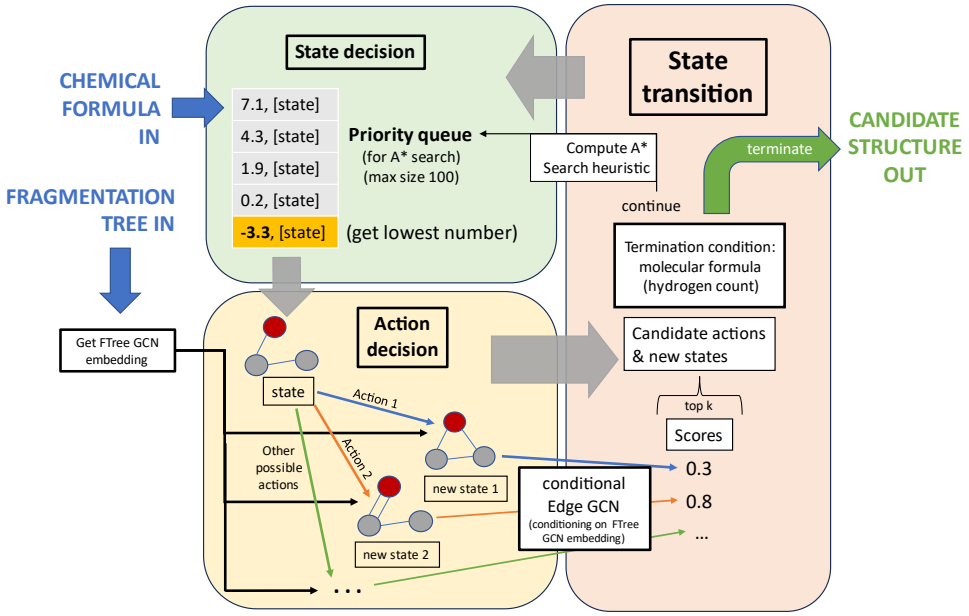

Fig. 4. A visualization of the generative A* search performed by our model, with the heuristic mentioned in section 2.5. The number in the priority queue is the negative of the heuristic, as Python's priority queue takes the smallest value.

So, a state being less anomalous and closer to the goal (less Hs left) will be better for the heuristic. To save memory, this will also use the beam width concept in beam search, always only storing the actions leading to the potential new states with the best 100 heuristic scores. Below is a visualization of the A* search in action.

3 Results & Discussion

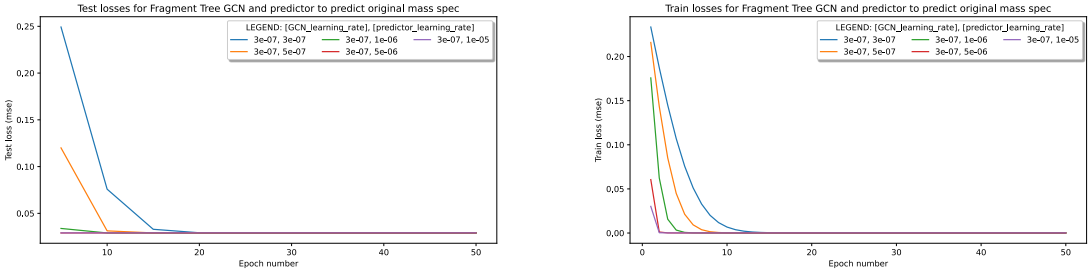

Fig. 5. Fragment Tree Graph Convolutional Network and predictor performance in predicting the original mass spectrum, given a grid search of hyperparameters for GCN learning rate and predictor learning rate.

We used the CANOPUS dataset in our tests, constructing fragmentation trees using the SIRIUS 4 software[15]. For reproducibility, all our code, as well as figures, can be found in this GitHub Repository: https://github.com/Simul-Eqn/CSE-final

When pretraining the FTree GCN, we used a learning rate of 3e-07 for the FTree GCN, and searched multiple learning rates for the predictor, with results plotted in figure 5, using Mean Squared Error loss. Since there is a lot of noise in these mass spectra, some amount of loss is to be expected. Figures related to this training step, and example mass spectrum predictions (using model at epoch 20 with predictor learning rate 1e-06) can be found under RL_attempt/figures/mass_spec_comparison in the aforementioned GitHub Repository.

We took the model at epoch 20 with predictor learning rate 1e-06 as an example model and trained the conditional graph convolutional network on molecules with at most 12 non-Hydrogen atoms, and that were either non-aromatic or with a benzene ring, then allowed all molecules with at most 12 non-Hydrogen atoms, then further relaxed it to at most 15 non-H atoms. We tested them on their accuracy in giving anomaly scores, where the best histograms are shown in figure 6. The mean and standard deviation at each epoch are tabled in annex 1.

Then, we used the ASTAR search algorithm with the heuristic mentioned in section 2.5 to attempt the main Chemical Structure Elucidation task. Figure 7 shows the success rates of our models compared to taking random actions given our restrictions, starting from an initial state of only aromatic bonds, taking up to 2 different actions from each state, and figure 8 is the equivalent with up to 3 different actions per state instead. Figure 9 shows the success rates if instead starting some amount of steps (depth) from the end, when taking up to 2 different actions per state, and figure 10 shows the same but with up to 3 actions per state instead.

Bar charts and graphs of the action accuracy rates are available in the annex 2. The accuracy of the actions was evaluated during the search, where each action taken would be compared to the list of correct actions.

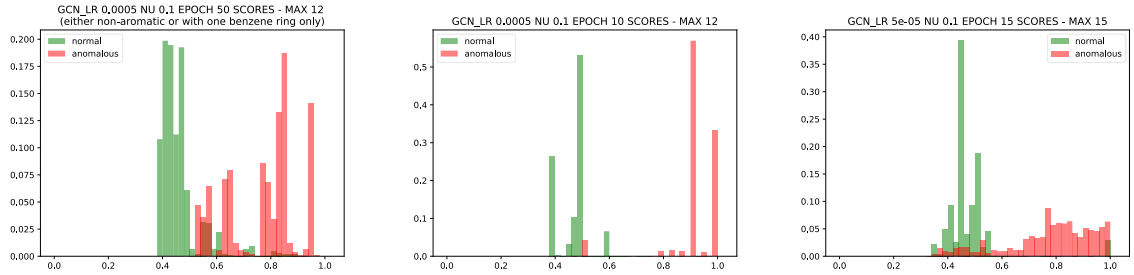

Fig. 6. Anomaly scores for test states with stated constraints. Green refers to normal states, considered non-anomalous, while red is considered anomalous. Corresponding figures at different training epochs are available in the GitHub repository, under RL_attempt/figures/non_anomalous_[case]_scores_visualization, where [case] is the task type, such as “max_12”

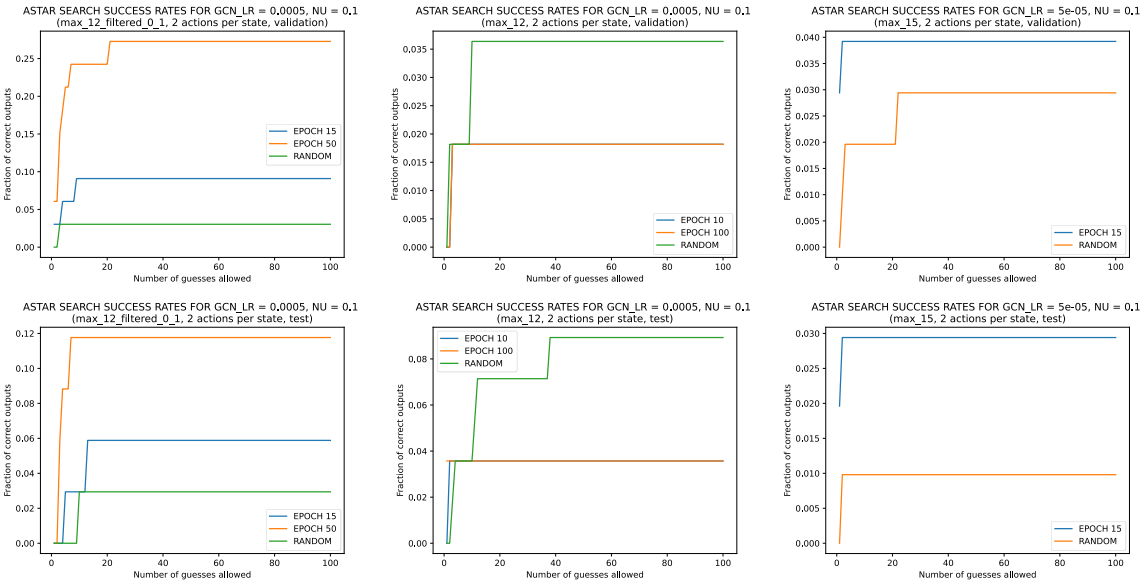

Fig. 7. A* search success rates in different scenarios, starting from initial state with only aromatic bonds, taking up to 2 actions per state.

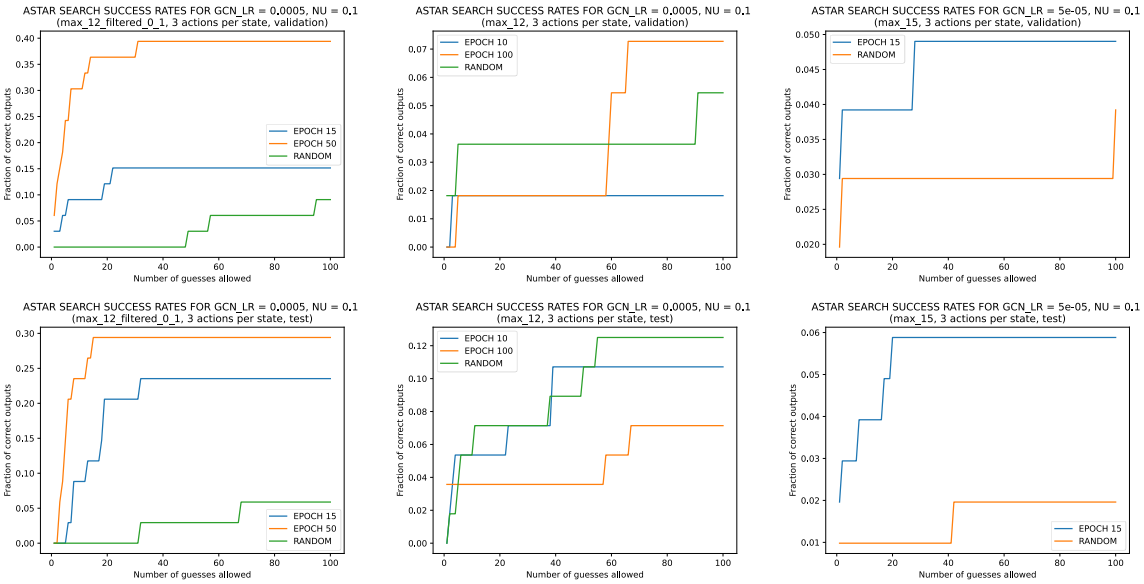

Fig. 8. A* search success rates in different scenarios, starting from initial state with only aromatic bonds, taking up to 3 actions per state.

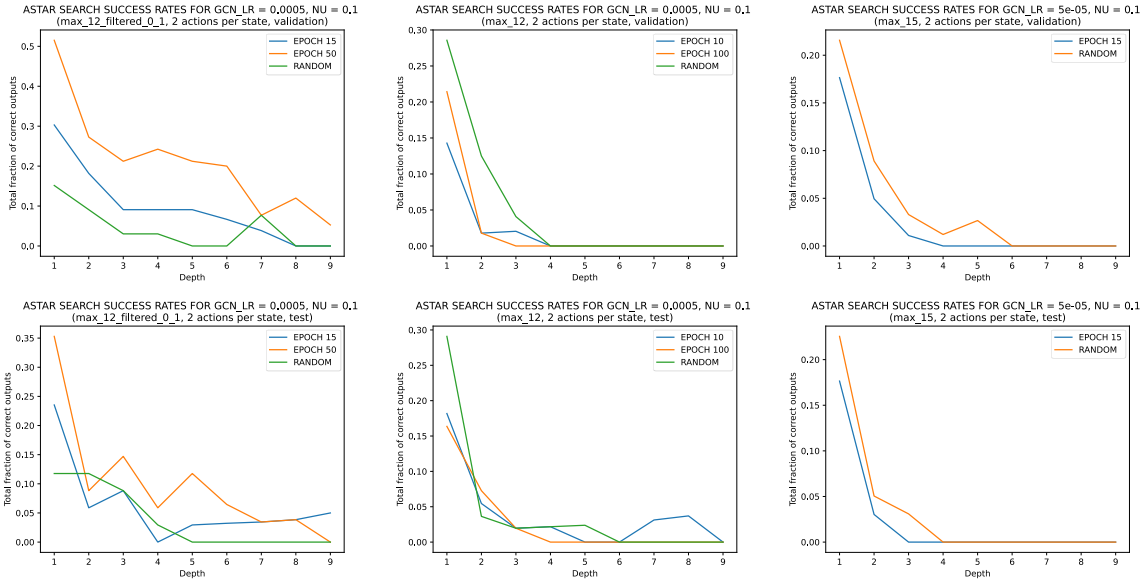

Fig. 9. A* search success rates in different scenarios, starting [depth] bonds away from the target state, taking up to 2 actions per state.

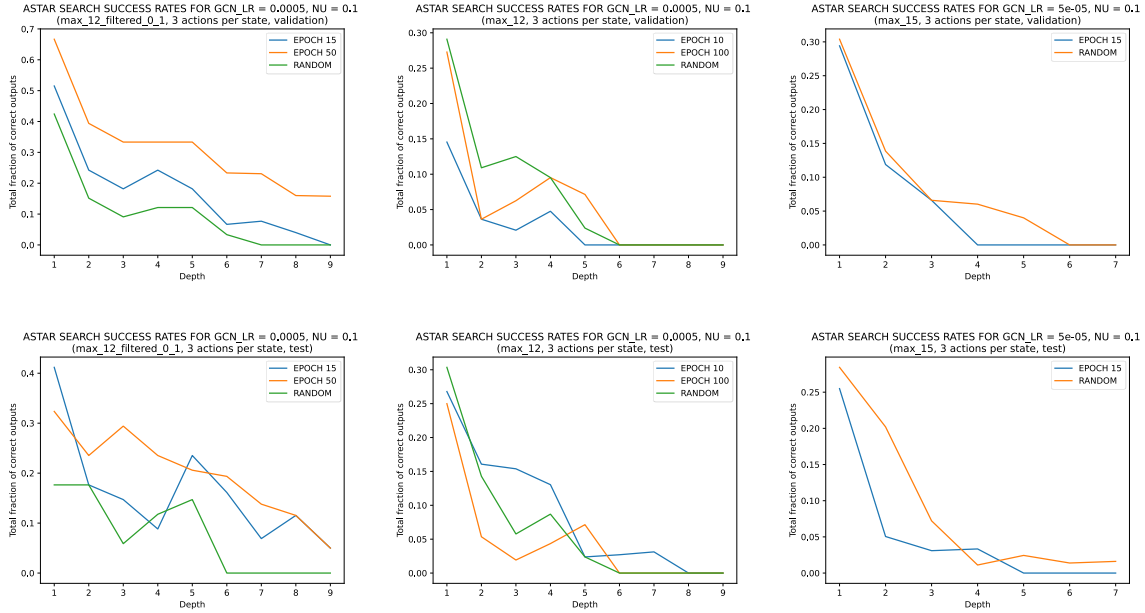

Fig. 10. A* search success rates in different scenarios, starting [depth] bonds away from the target state, taking up to 3 actions per state.

Observing figures 9 and 10, we realise that random actions seem to have comparable performance to our model, when running at low to intermediate depths, in the cases with at maximum 12 or 15 heavy atoms, despite there being a significant separation between normal and anomalous states observed in figure 6. In addition, our model's performance reaches near 0 around depth 5, yet, from figures 7 and 8, our model has a higher performance at maximum depth, especially when considering the case with at maximum 15 heavy atoms. To investigate this phenomenon, we graphed the frequency of single bonds as actions taken during the search starting [depth] bonds away from the target in figures 11 and 12. The frequency of single bonds as actions taken for full searches is in annex 3.

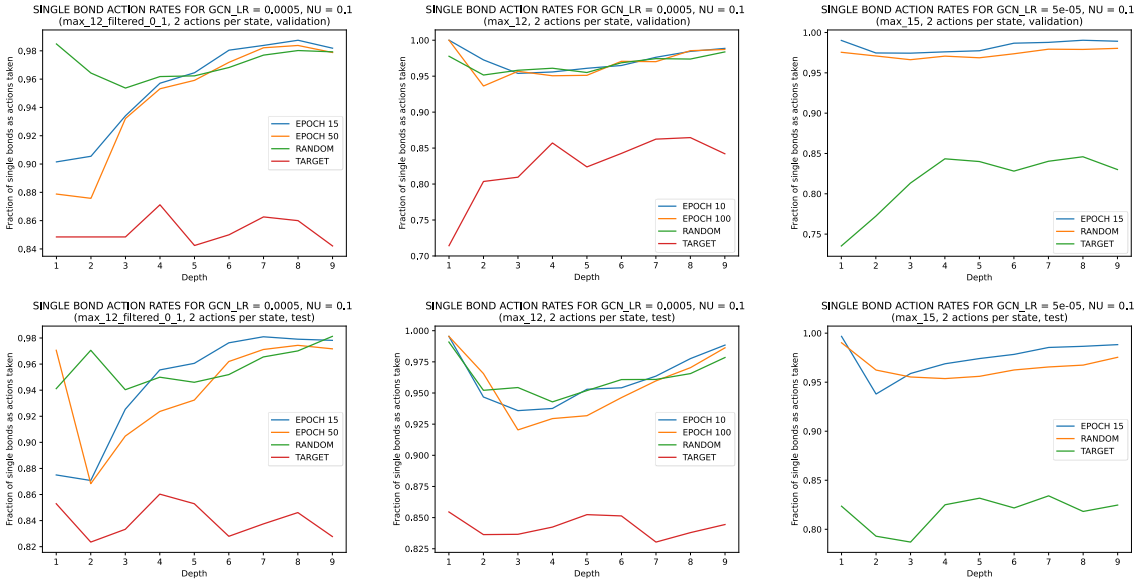

Fig. 11. Frequency of single bonds as actions taken during search, starting [depth] bonds away from the target state, taking up to 2 actions per state.

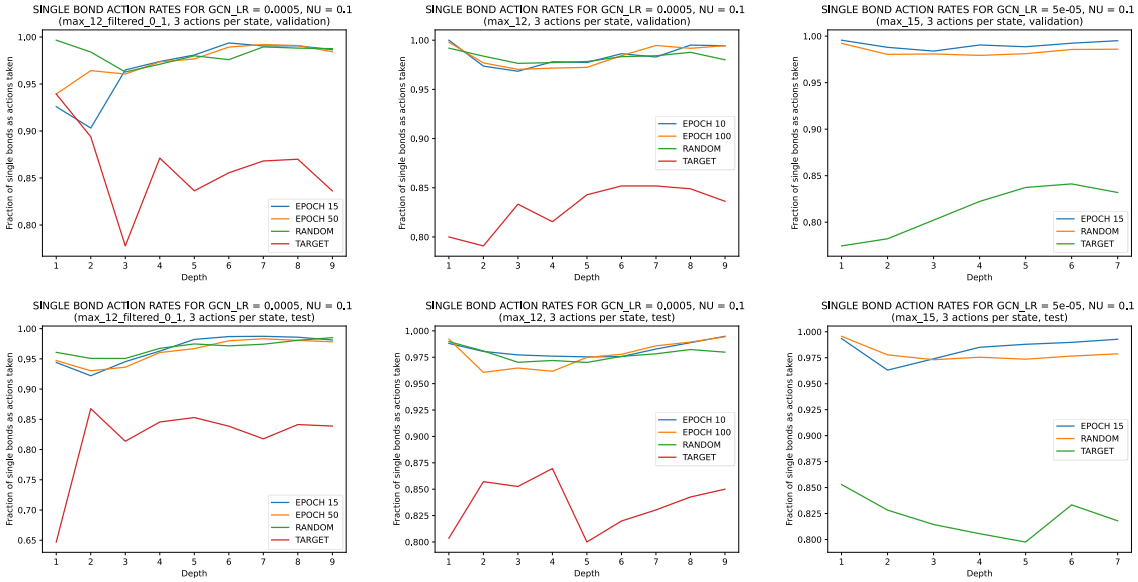

Fig. 12. Frequency of single bonds as actions taken during search, starting [depth] bonds away from the target state, taking up to 3 actions per state.

As can be very clearly seen in figures 11 and 12, as well as in annex 3, our models appear to favour states with more single bonds. Thus, a theorized reason for the model performing only marginally better, or even worse, than random at depths from around 5, is that favouring states with single bonds, or equivalently, taking more actions that result in the creation of single bonds, contributes greatly toward fulfilling the connectivity restriction, and does not cause the state to approach the limitation of the valency restriction as much, due to the nature of single bonds, allowing more possible actions than if random actions were taken instead, increasing the likelihood of choosing an incorrect action.

Despite this decrease in performance from depths of around 5, in figures 7 and 8, it can be seen that, at maximum depth, the model still gets some correct, with better performance than

it did around depth 5. This suggests that the model may somehow have the lowest accuracy with intermediate depths, while with low or high depths, the model has a higher accuracy.

This phenomenon could be due to two overlapping factors causing the training to only be effective with extreme values for depths: The first is the low amount of training data on states at low depths, as fewer bonds are removed so fewer permutations are available, which causes performance with lower depths to increase insignificantly during training, and only have performance at higher depths increase significantly. The second is the low amount of restrictions on the actions that can be taken at low depths as compared to high depths, not significantly affecting performance with higher depths, improving performance with lower depths. Due to these, perhaps the performance will be lower with an intermediate depth, while higher with depths closer to 1 or the maximum.

If that is taken to be correct, then the increase in performance due to the restrictions, considered in the second factor, can be clearly seen in figures 9 and 10, where the models perform greatly at a depth of 1 despite there being few training cases close to the target state.

4 Future work

Considering the low amount of training data on low depths in this training method, causing the model to learn from training data at high depths more, work is to be done in weighing these training cases to resolve this issue, or perhaps using focal loss.

In addition, regarding the potential issue where the model is biased towards taking actions that create single bonds mentioned in the discussion, even despite heavy training[1], work is to be done in considering this bias in the training data and verifying this phenomenon. Since this approach enforces model validity, having the model favour non-single bond creation will not be an issue as single bonds will eventually have to be created, and this is one of the potential solutions to be developed in the future.

More work should also be done in considering larger molecules, as the model trained on larger molecules appears to have performed significantly worse than those trained on smaller molecules. Investigation on the effectiveness of taking random actions in these scenarios is also to be done.

The method suggested in this paper also fails in that it does not allow the creation of aromatic bonds. A future direction may be considering incorporating the creation of aromatic structures in the action space, or otherwise allowing aromatic bonds to be generated by the model rather than be incorporated into the initial state.

Finally, work could be done in incorporating a GNN Explainer–inspired[28] utility for this to assist in chemical structure elucidation even when generation fails, and potentially be used in generating targeted loss for further training to improve performance.

5 Acknowledgements

YS thanks Lim Jing for advice in pretraining the fragmentation tree graph convolutional network.

[1] It is to be noted that, somehow, the bias towards single bonds is not as strong at low depths as compared to high depths, despite the aforementioned problems of having little training data at low depths. This question awaits further inquiry, of whether it was indeed a product of some generalization by the model, or somehow connected to the valency or connectivity restrictions, or for other reasons.

6 References

1. Buchanan BG, Smith DH, White WC, Gritter RJ, Feigenbaum EA, Lederberg J, Djerassi C: Applications of Artificial Intelligence for Chemical Inference .22. Automatic Rule Formation in Mass-Spectrometry by Means of Meta-Dendral Program. J Am Chem Soc 1976, 98(20):6168–6178.
2. Elyashberg, M.; Argyropoulos, D. Computer Assisted Structure Elucidation (CASE): Current and Future Perspectives. Magn. Reson. Chem. 2021, 59, 669–690.
3. Kerber, A., Laue, R., Meringer, M. & Rücker, C. *Molecules in silico: potential versus known organic compounds*. MATCH Commun. Math. Co. 54, 301–312 (2005).
4. Stravs, M.A., Dührkop, K., Böcker, S. et al. *MSNovelist: de novo structure generation from mass spectra.* Nat Methods 19, 865–870 (2022). https://doi.org/10.1038/s41592-022-01486-3
5. Gómez-Bombarelli, R. et al. Automatic chemical design using a data-driven continuous representation of molecules. ACS Cent. Sci. 4, 268–276 (2018).
6. Segler, M. H. S., Kogej, T., Tyrchan, C. & Waller, M. P. *Generating focused molecule libraries for drug discovery with recurrent neural networks.* ACS Cent. Sci. 4, 120–131 (2018).
7. Elton, D. C., Boukouvalas, Z., Fuge, M. D. & Chung, P. W. *Deep learning for molecular design—a review of the state of the art.* Mol. Syst. Des. Eng. 4, 828–849 (2019).
8. Philipp Koehn, Rebecca Knowles. *Six Challenges for Neural Machine Translation*. First Workshop on Neural Machine Translation, 2017. arXiv:1706.03872 [cs.CL]
9. Weininger, D. *SMILES, a chemical language and information system.* 1. Introduction to methodology and encoding rules. J. Chem. Inf. Comput. Sci. 28, 31–36 (1988).
10. Ucak, U.V., Ashyrmamatov, I. & Lee, J. *Reconstruction of lossless molecular representations from fingerprints.* J Cheminform 15, 26 (2023). https://doi.org/10.1186/s13321-023-00693-0
11. Zhang, M., Qamar, M., Kang, T., Jung, Y., Zhang, C., Bae, S., Zhang, C. A Survey on Graph Diffusion Models: Generative AI in Science for Molecule, Protein and Material. arXiv preprint: https://doi.org/10.48550/arXiv.2304.01565
12. Jo, J., Lee, S., Hwang, S. J.. Score-based Generative Modeling of Graphs via the System of Stochastic Differential Equations. ICML 2022. arXiv:2202.02514 [cs.LG]
13. Hoogeboom, E., Satorras, V. S., Vignac, C., Welling, M.. *Equivariant Diffusion for Molecule Generation in 3D.* ICML 2022. arXiv:2203.17003 [cs.LG]
14. Niu, C., Song, Y., Song, J., Zhao, S., Grover, A., Ermon, S.. *Permutation Invariant Graph Generation via Score-Based Generative Modeling.* AISTATS 2020. arXiv:2003.00638 [cs.LG]
15. Dührkop, K. et al. SIRIUS 4: a rapid tool for turning tandem mass spectra into metabolite structure information. Nat. Methods 16, 299–302 (2019).
16. Ahn, S., Chen, B., Wang, T., Song, L. *Spanning Tree-Based Graph Generation for Molecules*. ICLR 2022. https://openreview.net/forum?id=w60btE_8T2m
17. Liu, Q., Allamanis, M., Brockschmidt, M., Gaunt, A. L.. Constrained Graph Variational Autoencoders for Molecule Design. 2018. NeurIPS 2018. arXiv:1805.09076 [cs.LG]
18. Stein S.E. (2012) Mass spectral reference libraries: an ever-expanding resource for chemical identification. Anal Chem 84(17):7274–7282
19. Horai H, Arita M, Kanaya S, Nihei Y, Ikeda T, Suwa K et al (2010) MassBank: a public repository for sharing mass spectral data for life sciences. J Mass Spectrom 45(7):703–714
20. Wishart D.S., Knox C., Guo A.C., Eisner R., Young N., Gautam B et al (2009) *HMDB: a knowledgebase for the human metabolome*. Nucleic Acids Res 37:D603–D610
21. Tautenhahn R, Cho K, Uritboonthai W, Zhu Z, Patti GJ, Siuzdak G (2012) An accelerated workflow for untargeted metabolomics using the METLIN database. Nat Biotechnol 30(9):826–828
22. Dührkop, K., Shen, H., Meusel, M., Rousu, J. & Böcker, S. *Searching molecular structure databases with tandem mass spectra using CSI:FingerID*. Proc. Natl Acad. Sci. USA 112, 12580–12585 (2015).
23. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., Dahl G. E.. (2017) *Neural Message Passing for Quantum Chemistry*. ICML 70, 1263–1272.
24. Wang, X., Jin, B., Du, Y., Cui, P., Yang, Y.. (2021) *One-Class Graph Neural Networks for Anomaly Detection in Attributed Networks*. Neural Computing & Applications, volume 33, pages 12073–12085
25. Böcker S, Rasche F (2008) Towards de novo identification of metabolites by analyzing tandem mass spectra. Bioinformatics 24:I49–I55
26. Dührkop, K., Shen, H., Meusel, M., Rousu, J. & Böcker, S. (2015) *Searching molecular structure databases with tandem mass spectra using CSI:FingerID*. Proc. Natl Acad. Sci. USA 112, 12580–12585.
27. Brouard, C., Shen, H., Dührkop, K., d'Alché-Buc, F., Böcker, A., Rousu, J. (2016) Fast metabolite identification with Input Output Kernel Regression. Bioinformatics 32(12):i28-i36
28. Ying,R., Bourgeois, D., You, J., Zitnik, M., Leskovec, J. (2019) *GNNExplainer: Generating Explanations for Graph Neural Networks.* NeurIPS vol 32, pg 9244-9255.

7 Annex 1. Tables of exact values for anomalous scores

Scenario	Max 12 heavy atoms, only non-aromatic or with one benzene ring, epoch 50	Max 12 heavy atoms, epoch 10	Max 15 heavy atoms, epoch 15
Mean (normal)	0.4595	0.4678	0.4781
Standard deviation (normal)	0.07603	0.05167	0.1003
Mean (anomalous)	0.0.7695	0.9167	0.7792
Standard deviation (anomalous)	0.1277	0.09797	0.1613

Table 1. Mean and standard deviation of anomalous scores for normal and anomalous states, at different epochs in the mentioned scenarios. Note that, for the max 15 heavy atoms scenario, we only tested it on 20 molecules to save time, as each molecule could have up to 16384 cases at that size.

8 Annex 2. Correct action rates graphs

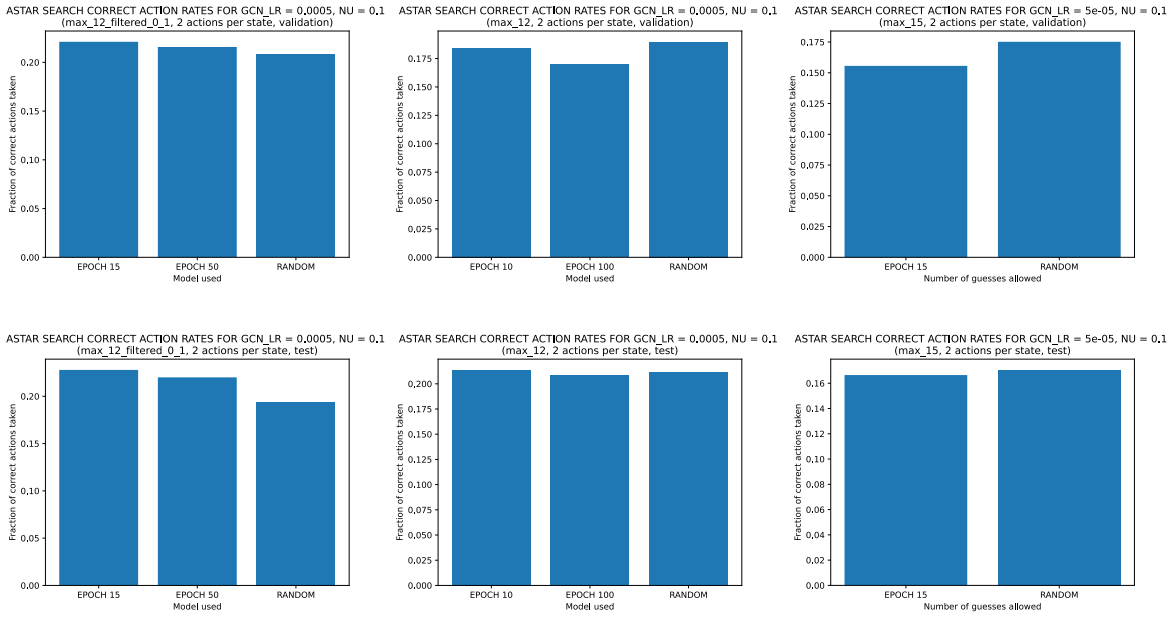

Fig. 13. A* search correct action rates in the above scenarios, starting from an initial state with no aromatic bonds, taking up to 2 actions per state.

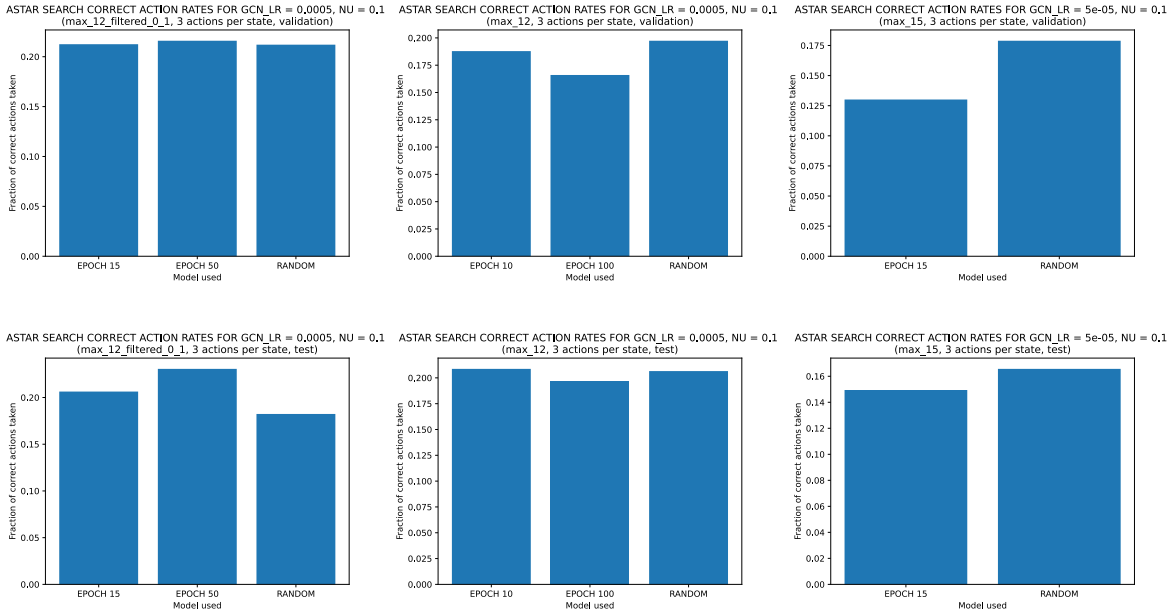

Fig. 14. A* search correct action rates in the above scenarios, starting from an initial state with no aromatic bonds, taking up to 3 actions per state.

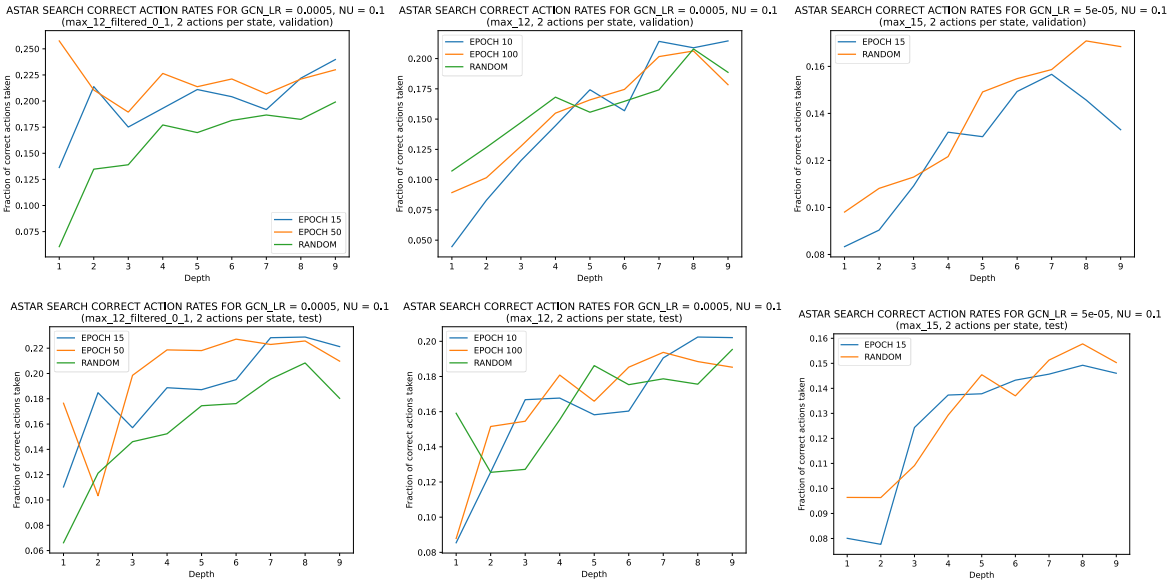

Fig. 15. A* search correct action rates in the above scenarios, starting [depth] bonds away from the target state, taking up to 2 actions per state.

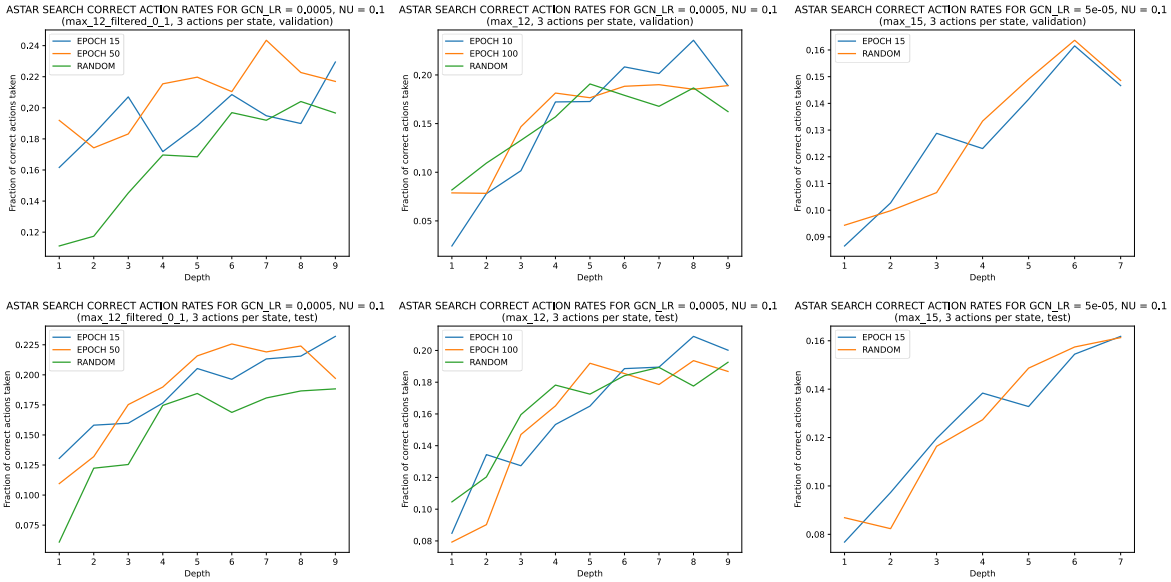

Fig. 16. A* search correct action rates in the above scenarios, starting [depth] bonds away from the target state, taking up to 3 actions per state.

9 Annex 3. Frequency of actions taken as single bonds

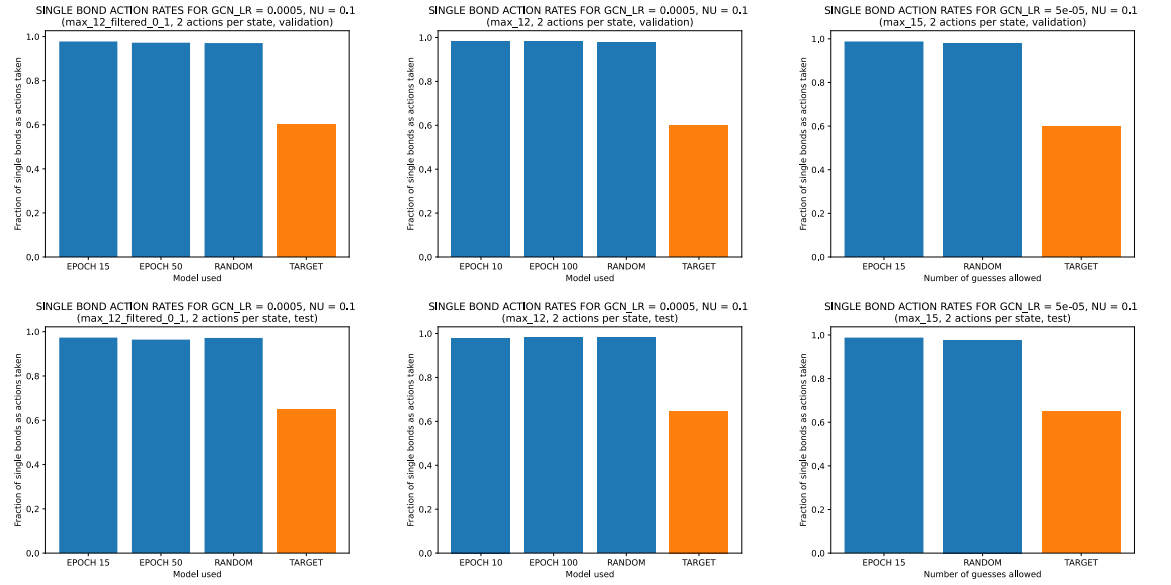

Fig. 17. Frequency of single bonds as actions taken during search, starting from an initial state with no aromatic bonds, taking up to 2 actions per state.

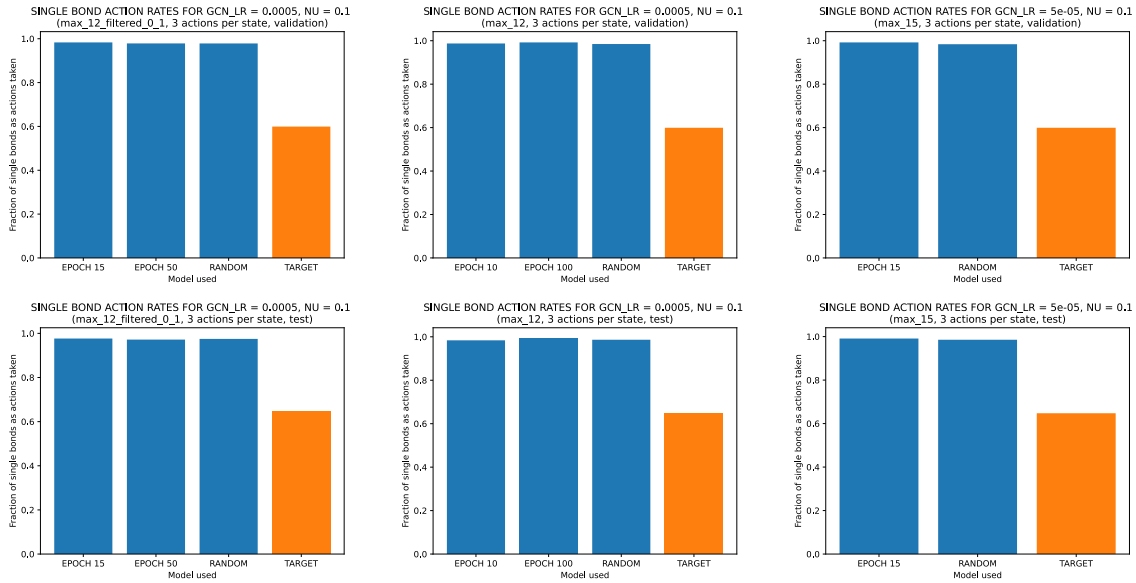

Fig. 18. Frequency of single bonds as actions taken during search, starting from an initial state with no aromatic bonds, taking up to 3 actions per state.