Reproducibility Study of ”Amortized Tree Generation for Bottom-up Synthesis Planning and Synthesizable Molecular Design”

Gabriel Fleischer, Elisa Michalski, Victor Sabanza Gil

Reproducibility Summary

Scope of Reproducibility

In this work, we reproduce some of the results obtained by SynNet[1], a ML model for *de novo* molecular design. We propose three main claims: reproducing table 1 and figure 4 from the original publication, testing the model on a different target molecules set, and training the model from scratch to obtain the same results as the default model. By stating these three objectives, we try to test and confirm the validity of the original results and the proposed model by checking different functionalities of the system. Our first claim tries to evaluate the robustness of the original model. The second claim tries to evaluate the model's ability to generalize to different chemical spaces. The third claim tries to confirm the validity of the original code to train the model from scratch and its robustness to obtain the same results.

Methodology

We used the original code to run the models and train networks from scratch. We also partially used this code to analyze the results. We wrapped the code in notebooks to reproduce results easier and wrote code to download data, run models and analyze results. Overall, we roughly spent 130 GPU hours training the models and 20 hours in preprocessing and inference tasks (computational time is specified in detail in *Computational requirements* section).

Results

Our results support the first and third claim, and partially the second. We reproduced recovery rates to within 5% of reported values for the reachable set and 2% for the unreachable set, confirming our first claim. We also got a difference of roughly 40% between both sets, similar to the one in the original work. The third claim is also supported by our results, as we obtained recovery rates differing from the default models by less than 5% in all cases by using our trained models. Our second claim is partially supported by our results, as we obtained a lower recovery rate value for the ZINC set compared to the reachable molecules set (18-23% vs 47-51%), but not as low as the obtained for ChEMBL. Overall, our results support the validity and reproducibility of the results described in the original paper.

What was easy

- The codebase was very well documented, we had no issue understanding each component's purpose and how to use it.
- Code modularity allowed running different parts of the project independently, speeding up the reproducibility task.

What was difficult

- The new version of PyTDC (3.8.0) has breaking issues, preventing any use of the Oracle feature. We had to downgrade it to version 3.7.0.[2]
- We were unable to use the computation in parallel feature as it was not following the library's guidelines[2] : The codebase uses global variables to pass fixed arguments to the parallelized function. This work on UNIX, but on Windows, every time a new thread is spawned, a new interpreter is created. Therefore, any global variable is reset, and it cannot be accessed by the thread. For the reason explained above, using global variables to pass arguments is a bad practice, and we had to modify the codebase and pass the arguments directly to address the issue.
- Table 1 is generated using the test set of the model training, which is not provided in the original repo. As we re-trained the model ourselves, we simply used the test set from this training.
- Computation time is high for standard hardware (e.g.: personal laptop). This limited the reproducibility scope and impose a time constraint on the project, forcing to focus on some specific tasks.

Communication with original authors

We contacted one of the current developers of the model. We briefly mentioned some of the problems we have found, like the version issue using PyTDC. We also intend to send this report with our reproducibility results to the original authors.

Abstract

De novo molecular design involves automatic generation of chemical structures and may be used to accelerate drug discovery. Although many generative models have been used to create new molecules with optimal properties, the proposed structures are usually synthetically infeasible. Coley et al. have recently proposed a model to overcome this limitation and generate synthetizable molecules. In this project, we reproduce and evaluate some of the main findings obtained by their model. Our results show that it is possible to reproduce the original results for synthesis planning, obtaining recovery rates of 47.2% and 6.0% for reachable and unreachable molecules respectively. We also add a different unreachable molecules set, getting recovery rates between 18 and 23%. In addition, we train the models from scratch, obtaining recovery rates that differ less than 5% from the original results. This work confirms the validity and robustness of SynNet, proposing it as a baseline for future work in this field.

I. Introduction

De novo molecular design involves creating new chemical structures through an automated process, with the goal of achieving some desired molecular characteristics. This approach can dramatically accelerate computer-assisted drug discovery. Recently, machine learning techniques have been applied to propose new molecules with desired properties [3]. Although these models are able to generate new chemical compounds with optimal properties, there is no guarantee that the proposed molecules can be synthesized in the lab. Therefore, synthesizability should be taken into account when designing new ML methods for *de novo* molecular design. There have been attempts to overcome this limitation by combining models for molecular optimization and models for synthesis planning in two steps [4]. However, no method has simultaneously addressed these two tasks. A method combining synthesis and property optimization tasks together would help to improve the progress of computer-assisted molecular design and the discovery of new high-value chemicals, facilitating its synthesis and further commercialization.

Coley et al. have recently proposed a new ML model for de novo molecular generation called SynNet [1]. This approach represents chemical synthesis as trees, where molecules are nodes and branches are chemical reactions linking one or more molecules that react together. It is possible to define the tree generation task as a Markov Decision Process (MDP), where each state is independent of the previous states (the same way a new step in a chemical synthesis does not depend on past actions). The model takes a target molecule as an input and it is able to build a synthetic tree to reconstruct this target from commercially available building blocks. It can also apply a genetic algorithm to optimize a molecular structure for a given oracle function, providing a synthesizable molecule that has a high score for the selected oracle. This represents a significant step towards *de novo* synthesizable molecular design and should be evaluated to understand its current limitations and potential improvements. In this work, we investigate SynNet and reproduce some of the results from the original publication. Our results agree with the original values and support the use of this model for synthesizable molecular generation.

II. Scope of reproducibility

The addressed problem is optimizing a molecule for a specific property while guaranteeing its synthesizability, or proposing a synthetic route to make it. The model takes a vector embedding representing the molecule as input and generates a synthetic tree stepwise by combining the results given by 4 Multi-Layer Perceptrons (MLPs). The final result is a synthetic tree whose root is the target molecule or a similar one, or an optimized molecule in the case of synthetic design. In the optimized synthesis planning, the molecule can be successfully reconstructed or an analog can be obtained.

We propose three main objectives in this study:

- Reproducing table 1 and figure 4 results from the original publication by running the default trained models provided by the authors for synthesis planning.
- Testing the model using a different unreachable molecules dataset as targets for synthesis, expecting a similar recovery rate to the one obtained for the ChEMBL dataset (4.5%) in the original publication.
- Training the networks from scratch following the instructions in the original repository, and using these results to run the model and obtain the same recovery rates as the ones obtained by running the default models.

III. Methodology

For this project, we mostly used the code provided in the original SynNet repository. The code is well documented and modular, allowing a non-expert user with minimal knowledge of *de novo* molecular design to implement all the necessary steps efficiently. Original code was used to run the models, generate and featurize the synthetic trees, and train the model from scratch. We have written code to download and process all the necessary datasets, run the reproducibility experiments, and analyze the results. Our reproducibility results can be generated by running the notebooks that are included in the project repository.

A. Model descriptions

The system described in the paper is based on 4 MLPs and is used to generate synthetic trees constrained by a synthetic target or optimization score. The model takes a vector concatenating an embedding of the target molecule and an embedding of the current synthetic tree step. Each MLP is used to predict one feature for one step of the synthetic tree generation of the target molecule. These networks are called Action (act), Reactant 1 (rt1), Reaction (rxn), and Reactant 2 (rt2). Action (act) network samples from 4 possible actions (add, expand, merge or end) that define the next tree step. Reactant 1 (rt1) network samples from the list of possible building blocks and the intermediate root molecules, selecting a molecule to apply in the next reaction. Reaction network takes the rt1 molecule output as input and samples from all the available reaction templates (if the input molecule can react). Finally, Reactant 2 network samples from building blocks list if the reaction selected from rxn network is bimolecular. The synthetic step is added to the tree by merging the outputs generated in each MLP. In the case of molecular optimization, a genetic algorithm is applied to optimize the target molecule embedding and condition tree generation towards high-performing structures.

B. Datasets

We used the same reaction templates and building blocks that were applied in the original paper (we requested access to the building blocks file from the chemical company Enamine, as these data were not open access). Data used to train the models have been generated by running the corresponding steps described on INSTRUCTIONS.MD in the original repository. The resulting files contain 269322, 89774, and 89775 synthetic trees for train, validation and test sets, respectively. We analyze data distribution in the three sets, confirming their similarity and a predominance of depth=1 trees and root molecules with a molar weight between 250 and 500. The analysis is included in data.ipynb.

For model testing, we employed three different sets of 10000 molecules each randomly taken from the test set, ChEMBL and ZINC respectively. Molecules from the test set correspond to root molecules of synthetic trees, and are named as *reachable* because they come from the same data distribution that was used to train the model (thus, the model would be able to reach these targets). ChEMBL dataset [5] was used in the original paper as a source of *unreachable molecules* (molecules that the model would not easily recover). This database contains biologically active molecules with drug-like properties that have been manually curated. We also included a different set of unreachable molecules extracted from ZINC dataset to test the model. ZINC is a free database of commercially-available compounds for virtual screening [6] .

C. Hyperparameters

Default hyperparameters were used to train the networks. All MLPs consist of 5 fully connected layers with 1000, 1200, 3000, and 3000 hidden neurons respectively. All hidden layers use batch normalization before applying ReLU activation. Adam optimizer was used for training with a learning rate of 1e-4 and mini-batch size of 64.

D. Experimental setup and code

To reproduce the original results, we have implemented the code in the form of Jupyter Notebooks, which can be found in the project repository. In run.ipynb, we run the necessary preprocessing steps to get the original models, molecular embeddings and molecules sets and then we run the models to generate synthetic trees for the target molecules. The analysis.ipynb notebook takes as input the files with the proposed synthetic trees from the run.ipynb and computes and plots the relevant statistics.

E. Computational requirements

The project was run on a node with AMD Ryzen 9 5900X 12-Core 2.2 GHz processors and 62 GB RAM. All models were trained on an NVIDIA GeForce RTX 3090 graphic card with 25 GB VRAM. The total GPU time spent to train each MLP model was: 30h (act), 22h (rt1), 40h (rxn) and 40h (rt2). Data preprocessing (building blocks precomputing, tree generation, filtering and featurization) took 3h in the workstation. Synthesis inference took 150 minutes for each set of 10000 molecules (with a total time of 15h for running the 6 sets of molecules). Optimization inference took 1h.

IV. Results

A. Results reproducing original paper

Reproduced results are displayed in table I (corresponding to table 1 on the original paper). We used 10000 randomly sampled molecules for each testing set (reachable and unreachable) to run the synthesis planning. Although the original publication used a bigger sample size (69548 reachable and 20000 unreachable molecules, respectively), we used a smaller one to reduce the computing time by assuming a similar distribution and a sufficiently large sample size. We compare the recovery rate (percentage of target molecules that are correctly reconstructed) obtained by our reproduced model to the original recovery rate. In the case of the reachable set, the rate of recovered molecules is 47.20% (51.0% in the original paper). In the case of the unreachable set, we obtain a value of 5.97 % (4.5% in the original paper). The reachable average similarity is 0.581 (0.759 in the original paper), and the unreachable similarity is 0.479 (0.423 in the original paper).

Table I. Results of reproduced synthetic tree construction for “reachable” and “unreachable” target molecules

Dataset	N	Recovery Rate	Average Similarity	KL Divergence	FC Distance
Test set	10000	47.20 %	0.581	0.993	0.153
ChEMBL	10000	5.97 %	0.479	0.961	1.897

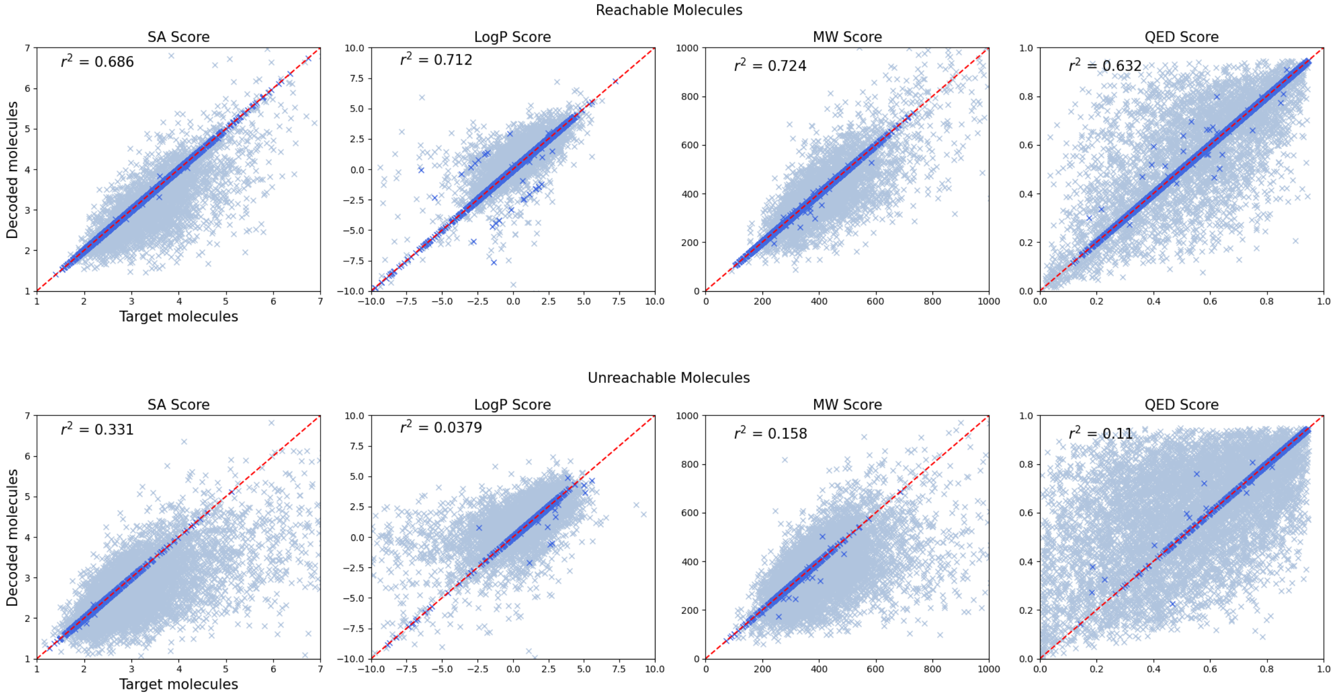

Fig. 1: Correlation between properties of target and product molecules of reproduced results.

Figure 1 shows the correlation between some properties of the targets and products for reachable and unreachable sets (corresponding to figure 4 in the original paper). Reachable correlation values are between 0.63-0.72 (0.60-0.70 in the original paper), while unreachable values are between 0.11-0.33 (0.03-0.28 in the original paper).

We have also run a simple optimization inference for GSK score using random molecules from ZINC as seed, but with a lower number of generations than in the default script due to a time constraint (thus expecting a lower score). The top molecule has a GSK score of 0.78, and the proposed structures are simpler than the ones showed in figure 5 on the original paper for other optimization methods.

B. *Results beyond original paper*

Table II displays results for synthesis planning using default and trained models and three different target molecules sets (reachable, ChEMBL and ZINC). Synthesis planning using default model with ZINC dataset shows a recovery rate of 18.98%. Results using our trained model show recovery rates of 47.2, 6.05, and 23.7 for unreachable, ChEMBL and ZINC sets respectively. We also provide average similarities and KL divergences and FC distances. Average similarity is lower than 0.6 in all cases, being higher for the ZINC set than for the reachable set.

Table II. Results of synthetic tree construction for “reachable” and “unreachable” target molecules using default and trained models

Dataset	N	Recovery Rate	Average Similarity	KL Divergence	FC Distance
Test set (default)	10001	42.41 %	0.549	0.991	0.182
Test set (trained)	10000	47.20 %	0.581	0.993	0.153
ChEMBL (default)	10000	6.05 %	0.466	0.963	1.943
ChEMBL (trained)	10000	5.97 %	0.479	0.961	1.897
Zinc (default)	10000	18.98 %	0.571	0.763	1.178
Zinc (trained)	10000	23.73 %	0.579	0.742	0.921

C. Discussion

Table I and figure 1 results are in agreement with the original paper. First, recovery rates for the reachable set of molecules differ by less than 5% and have values around 50%, suggesting that the model can successfully propose synthetic routes for half of the molecules in the test set. Similarity values also confirm this trend. The recovery rates for ChEMBL (unreachable molecules) differ by less than 2% and have values around 5%, proving that the model is not able to generally reconstruct molecules from this data distribution. The model was trained on a set of reaction templates, and test molecules are a combination of those reactions, while ChEMBL molecules may require different reactions for synthesis. This fact makes ChEMBL set harder for the model and originates a large difference in recovery rates between the two molecule sets. The similarity of original and reproduced values also indicates that a sample size of 10000 is sufficiently large to reproduce the results, and confirms the consistency of our first claim. Besides, the optimization algorithm is able to generate molecules with high oracle function scores that are also structurally simpler and easier to synthesize than previous examples, as it was suggested by the authors.

The additional ZINC set used to evaluate the original model has a recovery rate between the reachable and unreachable sets (18 and 23% for default and trained models, respectively). This may be due to the fact that molecules from this dataset are more general than the ChEMBL dataset and therefore may be reachable by the chemical reactions included in the training reaction templates. This partially supports our second claim, proving that the model is not able to perform as well as in the reachable molecules set for different molecule distributions. It also supports the outlook provided by the authors, where they suggested that a more comprehensive reaction template set may expand model reachable chemical space and improve recovery rates.

Finally, all results obtained by our trained models show recovery rates similar to the ones obtained with the default models trained by the authors. Table II displays recovery rates for the three molecule sets in both default and trained models, showing that all differ by less than 5%. This suggests that the training process has been correctly done and that it is possible to train the models from scratch and obtain the same model as the original publication using the base code in the original repository. It also confirms our third claim and supports the reproducibility and robustness of the original code.

V. Conclusion

We have studied and reproduced the results of SynNet, a model proposed by Coley et al. for synthesizable *de novo* molecular design. We have proposed 3 claims to test model reproducibility: reproducing table 1 and figure 4 from the original paper, testing synthesis planning on molecules extracted from ZINC database, and training the model from scratch to check if the computed results are still the same. Our results confirm first and third claims, and partially the second claim. We have shown that recovery rates are consistently higher for reachable molecules set than for ChEMBL set (around 50% vs 5% respectively) using default and trained models. Besides, we have obtained recovery rates around 20% using both models, confirming that the system performs worse with molecules coming from a different data distribution than the one generated with reaction templates included in the model. This project supports the validity and robustness of SynNet, and proporses this model as a baseline for future improvements and new ML systems for this task.

References

[1] Wenhao Gao, Rocío Mercado, and Connor W. Coley. Amortized tree generation for bottom-up synthesis planning and synthesizable molecular design, 2021. arXiv:2012.11522.

[2] Janis Born. Oracle import in 0.3.8. https://github.com/mims-harvard/TDC/issues/180, 2022. [Online; accessed 19-December-2022].

[3] Joshua Meyers, Benedek Fabian, and Nathan Brown. De novo molecular design and generative models. *Drug Discovery Today*, 26(11):2707–2715, 2021.

[4] John Bradshaw, Brooks Paige, Matt J. Kusner, Marwin H. S. Segler, and José Miguel Hernández-Lobato. Barking up the right tree: an approach to search over molecule synthesis DAGs, December 2020. arXiv:2012.11522.

[5] Mark Davies, Michał Nowotka, George Papadatos, Nathan Dedman, Anna Gaulton, Francis Atkinson, Louisa Bellis, and John P. Overington. ChEMBL web services: streamlining access to drug discovery data and utilities. *Nucleic Acids Research*, 43(W1):W612–W620, 04 2015.

[6] John J. Irwin, Teague Sterling, Michael M. Mysinger, Erin S. Bolstad, and Ryan G. Coleman. Zinc: A free tool to discover chemistry for biology. *Journal of Chemical Information and Modeling*, 52(7):1757–1768, 2012. PMID: 22587354.