Dynamically Scheduled Curriculum Learning for Molecule Generation

Rayan Taghizadeh, Kaushik Vemparala

1 Extended Abstract

Curriculum (staged) learning has been an important advancement in Deep Reinforcement Learning over the last few years. A modified training schedule where the agent starts with easier environments (i.e. learning how to move limbs) before moving on to more difficult ones (i.e. walking) has shown to be effective in tasks where the final goal is too difficult to directly optimize. Training in this type of environment would yield sparse rewards for the agent and unsubstantial updates to the policy.

While these methods have proven to be effective, most curriculum learning tasks still rely on a source of expert intuition to sort the schedule from which the agent learns. The field of automated curriculum learning has emerged in order to remove this dependence and form an optimized staging of goals.

In this paper we explore a naive, "greedy" implementation of curriculum learning applied to train an RL agent towards generating pharmacologically active molecules. After every stage of training, we sampled molecules from the trained agent and defined distributions of their individual scoring breakdowns. We then chose the next reward function to train over based on the scoring function that yielded the highest expected value, essentially optimizing the agent over something that it was already good at. By doing this, the agent could generate molecules that incrementally improve in function based on the scores from the previous round.

From a combination of domain knowledge, ChatGPT, and implementation ability, we defined 4 key rewards to train our agent with: Molecular Weight (how heavy the generated molecule is), H-bond info (number of hydrogen-bond donors and hydrogen-bond acceptors), QED (how drug-like a molecule's properties are), and SAScore (how easy the molecule is to synthesize). We then separately trained agents on three different schedules: our algorithm's proposed schedule, an "expert" intuition schedule (GPT-4 opinion), and a randomly generated schedule.

We observed better performance on the "drug-likeness" metric compared to the other two runs, indicating the potential for more complex dynamic scheduling methods, such as teacher algorithms, to be very effective in the context of drug design/generation. In addition, our approach achieves much more stable scores during training than other approaches.

2 Introduction

A domain that has been bottlenecked by extensive, time-consuming processes is drug discovery. One of the main challenges in this field is generating novel molecules that satisfy a set of strict constraints to elicit therapeutic effect–for example, a proposed molecule weighing above 500 Daltons is unlikely to have a strong pharmacokinetic profile due to difficult absorption through gut walls. While drug discovery has recently seen a boom in the incorporation of deep learning to find new drug candidates, many concepts have not been fully explored.

In deep learning problems, there are multiple ways to represent a molecule. A common method is to treat them as graphs, with each atom being a node, and each bond being an edge of different weights. Another common way is through a 2d representation of its atoms in a structure known as "SMILES" (Simplified Molecular Input Line Entry System) Strings. These sentence-like representations are easy for an LLM to learn, and serve as the molecular representation throughout the paper.

The the main functions used to score a generated SMILES string were part of the RDKit library. These scores were normalized so that they shared the same scale, and during each training run, they were fully weighted to that objective was the only focus of the agent.

3 Related Works

We derived the inspiration for this project through the recently published REINVENT paper: doi.10.26434/chemrxiv-2023-xt65x.

4 Training Background

We treat this problem through a policy-based reinforcement learning approach consisting of a prior and an agent. These both begin as LLMs pretrained on a large corpus of SMILES such that it understands the 2d grammar of molecules and can be abstracted out of this experiment. During training, the agent generates tokens, and the probability of the token sequence is given by an augmented likelihood equation,

$$\log P_{\text{aug}}(T) = \log P_{\text{prior}}(T) + \sigma S(T)$$

This equation was proposed by the authors of the REINVENT paper, which includes a regularization term for the generalized prior.

We incorporate this likelihood into the loss function used to train the agent, which the authors of the paper refer to as "Difference between Augmented and Posterior" (DAP), which takes the form:

$$L(T) = \left(\log P_{\text{aug}}(T) - \log P_{\text{agent}}(T)\right)^2$$

We used $\sigma = 128$ as the original paper did. We also retained a diversity filter (to generate samples across all protein manifolds) and an experience replay buffer to incorporate a loss from the highest-scoring molecules.

We had a bank of scoring functions compiled from the original repository and the RDKit Chemistry library. Based on personal domain experience and Chat GPT-4 opinion, we augmented some of the scoring functions and distilled them to a set of five important ones: Molecular Weight, H-bond info, QED, and SAScore.

5 Implementation

In the regular implementation of curriculum learning, we defined the expert's sequence of rewards before training, and executed them in order. In our random trial, we randomly selected a stage to perform next upon completion of the previous stage. In our proposed solution, before every round of training, we sampled 100 molecules (in the form of SMILES strings) from the updated agent and generated distributions of scores from scoring functions we did not choose yet. We then used the scoring metric with the highest mean,

reasoning that we should let the model optimize over something it is already relatively good at. The runs are seen below:

6 Experiments

We tested three different approaches to scheduling the curriculum:

- VanillaCL: This is the basic approach in which the curriculum is set beforehand. (White lines in plots)
- RandomCL: In this approach, after each stage a random stage is selected to be completed next. (Red lines in plots)
- CurricuChemL: In this the proposed approach in this paper. (Green lines in plots)

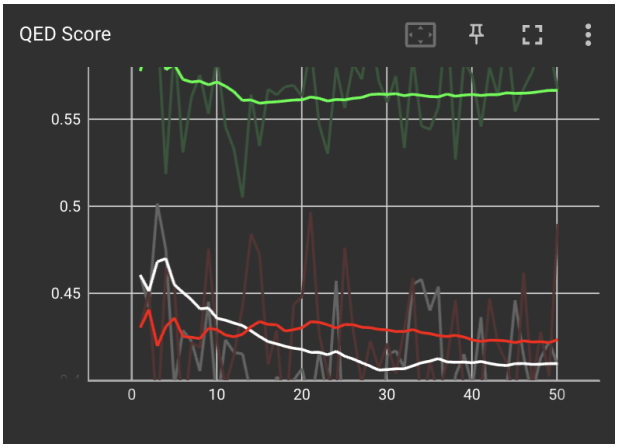

Figure 1: QED Scores.

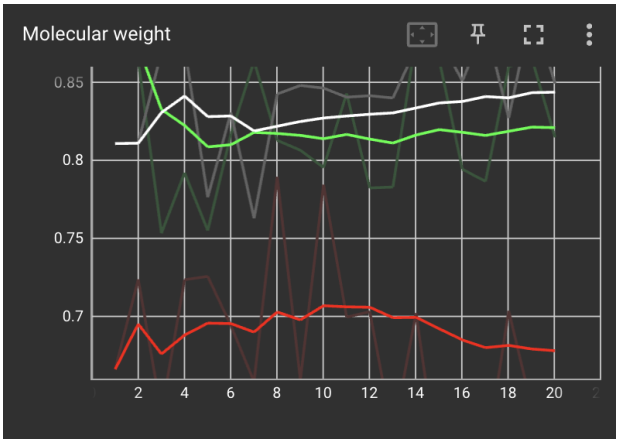

Figure 2: Molecular Weights.

Figure 3: HBond Donors.

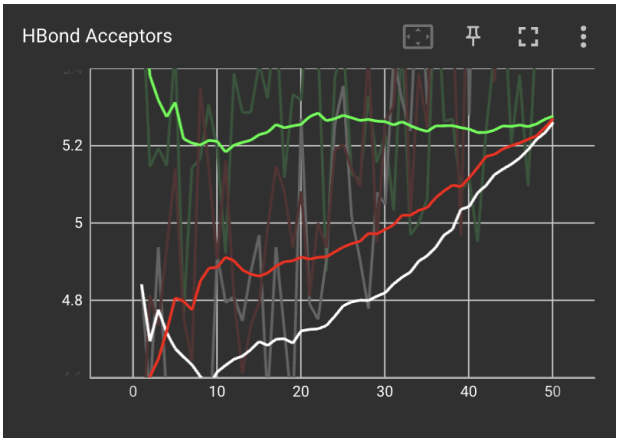

Figure 4: HBond Acceptors.

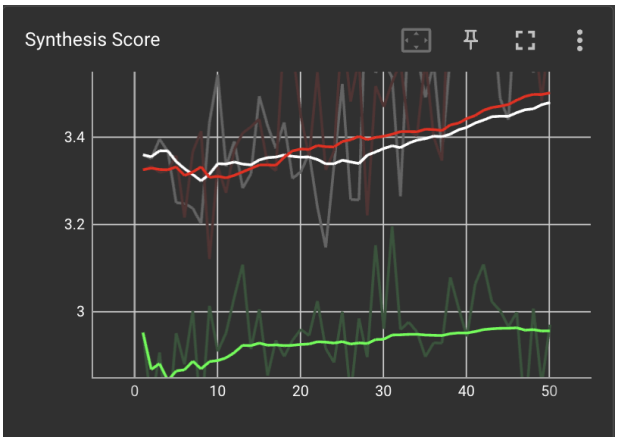

Figure 5: Synthesis Scores.

6.1 Discussion and Path Forward

As can be seen in the plots, the results are quite different between the scoring functions. However, one thing is common among all scoring methods: CurricuChemL is the most stable of the three approaches. In addition, in the majority of the scoring methods, CurricuChemL outperforms (or is neck to neck with) the two other approaches.

As for the path forward, CurricuChemL is currently only implemented for the Reinvent model, so one natural extension is to provide functionality to train the other models (Mol2Mol, LibInvent, etc.). In addition, instead of using the mean to pick the next stage, other metrics can be looked into, and even a combination of metrics. Finally, we could experiment with possibly running the training for longer, and with different parameters.

References

[1] Loeffler H, He J, Tibo A, Janet JP, Voronov A, Mervin L, et al. REINVENT4: Modern AI–Driven Generative Molecule Design. *ChemRxiv*. Cambridge: Cambridge Open Engage; 2023; This content is a preprint and has not been peer-reviewed.

[2] arXiv:2003.04664.

[3] arXiv:2303.12726.