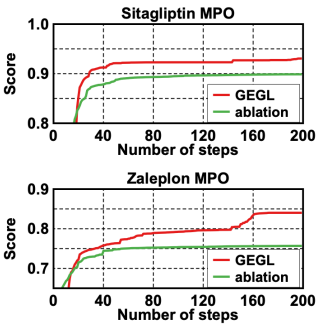
We sincerely thank all reviewers for their valuable efforts and insightful comments. As the reviewers have pointed out, we believe that our Genetic Expert-Guided Learning (GEGL) framework provides a substantial contribution to the field with a novel or timely idea (R1, R2), clear writing (All), and extensive evaluations (All). In the following, we provide our responses to the comments.

Response to R1

Unclear contribution of the apprentice policy. We thank R1 for the helpful comment. *The apprentice policy contributes to GEGL by encoding knowledge over many molecules seen throughout the training.* This is in contrast to the genetic expert policy which only use molecules in the priority queue $\mathcal{Q}_{\texttt{ex}}$ to generate molecules. Especially, the genetic expert policy alone cannot outperform GEGL since it is likely to meet a poor local optimum when important "seed" molecules are discarded from the priority queue $\mathcal{Q}_{\texttt{ex}}$.



Following R1's insightful suggestion, we compared GEGL with an additional "ablation" algorithm in the right figure. The algorithm is similar to GEGL without the apprentice policy (in Section 4.3), except the expert policy using molecules from $\mathcal{Q}_{\texttt{ex}}$ (instead of $\mathcal{Q}$). We will incorporate this in the final draft, for further clarifying the contribution of the apprentice policy in GEGL.

Clarification on details. We thank R1 for the opportunity to make the following clarifications. First, we indeed used a different set of low-scoring molecules under different `PenalizedLogP` metrics. Next, we observe that GEGL is not biased towards generating small molecules; our second-best molecule for optimizing `PenalizedLogP` is a chain of 81 sulfur atoms with `PenalizedLogP` value of 31.790.

Response to R2 and R3

Generated molecules being unrealistic. We thank R2 and R3 for mentioning an important point. We agree with R2's comment: the current literature fails to search for a molecule that is high-scoring and realistic simultaneously.[1] However, we are believe *GEGL can generate high-scoring and realistic molecules under proper regularization, as supported by Table 2(b).* In the experiment for Table 2(b), we apply a post-hoc filter [Brown et al., 2019] for rejecting unrealistic molecules as suggested by Gao and Coley [2020], and show that GEGL significantly outperforms the baselines for finding high-scoring molecules even after rejecting many unrealistic molecules. A similar approach can be used for settings where the oracle score function is unknown (as described by R3), e.g., one may use a DNN that estimates the true score, while also accounting for the uncertainty of its estimation and realistic-ness of the molecule for regularization.

Irrespective of the "unrealistic molecule" issue, *the impressive capability of GEGL for finding deficiencies in the scoring functions can be useful* in its own way. To be specific, it is valuable to have methods that can quickly find the limitations and pitfalls of optimization tasks. Such methods allow us to gain intuition on the problem and to develop better and rational candidates for the optimal solutions. For example, practitioners have reported many cases for finding bugs of hardware or simulation while running evolutionary algorithms. We also refer to more detailed discussion on this point by Lehman et al. [2020].

Simple method that lacks novelty. We do believe that our work is novel; GEGL is the first to offer a new paradigm of combining deep reinforcement learning with domain-specific exploration. Since such a paradigm is not known in the current literature, it may inspire researchers to develop similar algorithms in other domains. Furthermore, we believe the simplicity of GEGL is its strength rather than a weakness. Namely, we believe GEGL to be robust, easy to implement, reproducible, and extendable to broader applications.

Response to R1, R2, and R3 (for editorial comments)

We plan to fully incorporate the incredibly helpful comments in the final draft, with the following highlights:
(R1) We will change Table 2 using the standard `PenalizedLogP` metric, as reported in the supplementary material.
(R1, R2) We will report more of the generated molecules in our final draft and the codebase.
(R3) We will clarify how DA-GA and GB-GA are different from GEGL; DA-GA only uses a DNN to augment its score function and GB-GA use the same genetic operator as GEGL without using a DNN.

References

N. Brown et al. Guacamol: benchmarking models for de novo molecular design. *JCIM*, 2019.
W. Gao and C. W. Coley. The synthesizability of molecules proposed by generative models. *JCIM*, 2020.

[1] As we discuss in Section 4.1, this problem arises for methods regardless of the choice on using a DNN or a genetic algorithm.