The following is our response to all major comments.

Baselines and benchmarks (Reviewers #2 and #3): We appreciate the reviewers' input on additional baselines and benchmarks. The reason that the method MSO in "Efficient multi-objective molecular optimization in a continuous latent space" achieved a higher penalized logP with unlimited property evaluations than ours (26.1 vs 15.18) is due to different experimental settings. The best achievable penalized logP score in this experiment is highly correlated with the maximum number of atoms allowed, which, in our method, is determined by the maximum number of steps (L_{max}). We set $L_{max} = 51$ in the experiment to be consistent with the training data (Appendix C, as commonly done in the literature), while the top molecules presented in the above paper involved many more atoms than 50. With a larger L_{max}, the best penalized logP score can be significantly increased. For instance, with $L_{max} = 110$, our top penalized logP score increases to 32.38 and by limiting MSO to molecules of at most 50 atoms, its top penalized logP score decreases to 14.19. We will include a fair comparison with the new baseline in the final version. We have started running the experiments on GuacaMol as suggested. But, due to the size of the data and number of objectives, we estimate that the full experiments will take several weeks to complete and hope to report the results in the final version.

Insight into the performance improvements (Reviewer #2): Compared with a molecular hypergraph grammar (MHG), our proposed method has a higher coverage rate (in Appendix C, MHG failed to cover 16 of the 5000 molecules while our method only failed to cover two). Thus, our inferred grammar can represent more molecular structures and explore the chemical space more effectively. Compared with atom-by-atom methods such as GCPN, our method is much more sample-efficient since the validity of the generated structure is regularized by the grammar and the deep-learning model can focus on property optimization. Moreover, we think that the ability to update bond features in the GCN also helps to improve the performance. These insights as well as a suitable ablation study will be added in the final version.

Validity and evaluations of generated molecules (Reviewers #3 and #4): We apologize for overlooking the validity of the molecules presented in Figures A.2 and A.3 containing a valence-5 carbon. This molecule is only intended for clarifying the inference and derivation steps of our proposed grammar and is **not** among the molecules generated by our method. We will fix these two figures in the final version. All generated molecules in the appendix have been double-checked by both RDkit and human experts. Since almost all previous publications used RDkit to check the validity of generated molecules, we think it is a reasonable measure. On the other hand, we agree that some unstable structures might be missed by RDkit, which is a well-known problem in molecular optimization with ML methods.

The diversity of the generated molecules in optimizing QED and penalized logP with unlimited property evaluations are both 0.8, and we will add these in the final version. Following ORGAN and GCPN, the diversity within generated molecules is calculated as the average pairwise Tanimoto distance between the Morgan fingerprints of the molecules. Same as previous work, the radius is 4, the number of bits is 2048, and the range of the obtained diversities is consistent with the values presented in the paper of GCPN. Thus, we believe our results are reliable.

Limited number of property evaluations (Reviewers #1, #3 and #4): We highlight that our method can optimize molecules with a limited number of property evaluations because in many real-world biological and biomedical applications, the required property evaluations can be very expensive while efficient and accurate proxy models are unavailable. For instance, in the development of novel drugs, wet-lab experiments or time-consuming computations are usually needed to evaluate the properties of a molecule, and thus a model's ability to perform optimization with a limited number of property evaluations is crucial and can significantly reduce the cost. But, we appreciate the Reviewer 4's suggestion of plotting the results of this experiment and plan to add a plot in the final version.

Antibacterial experiment (Reviewers #2 and #4): This experiment is meant to illustrate an application of our method to a real-world problem. A trained classifier is used as a surrogate for the unknown property evaluation function as wet-lab experiments were not available in our lab. Since the evaluation method (inhibitor scores) of generated molecules is independent from the pseudo evaluation function, we think that the molecules generated by our method with high inhibitor scores can still be used as candidate antibacterial molecules.

Relationship with the original NCE grammars (Reviewer #2): We have made significant modifications to the original NCE grammars especially concerning the embedding function and LHS of a production. It is easy to see that the valency validity cannot be guaranteed by the original NCE grammars. In our proposed grammars, we constrain the form of an LHS to a subgraph consisting of only *a node and its neighbors* to simplify the production rules and our embedding function defines the labeled connections between these neighbors and the nodes in the RHS. As the bonds connected to an atom remain constant throughout molecule generation process and the production rules are chosen based on known molecules, the valency validity can be guaranteed.

Model training details and running time (Reviewer #4): Since the classical PPO is used in our method, off-policy training is not supported. On the ZINC dataset, the pre-training of our method took less than 24 hours and each optimization task took less than 12 hours. More details of running time comparison as well as hyper-parameter optimization will be included in the final version.