Towards Optimized Use of LLMs in Drug Discovery

Anonymous Author(s)
Affiliation
Address
email

Abstract

Large language models (LLMs) have recently emerged as a promising tool for small-molecule generation in drug discovery. One notable recent work in this field is MOLLEO, which combines an evolutionary algorithm with an LLM that acts as the operator for making crossovers and mutations on the ligand population. MOLLEO demonstrates strong results on optimizing molecular docking scores, but several aspects of their model are not well suited to real-world drug discovery. In this work, we make a set of novel optimizations that greatly improve the efficacy of LLMs in small-molecule drug discovery. First, we show that MOLLEO's use of molecular docking as the fitness function results in ligands unlikely to show experimental binding using molecular dynamics simulations. We find that replacing docking with the recently released biomolecular foundation model Boltz-2 greatly improves the predicted binding affinity from molecular dynamics. Second, we incorporate knowledge of existing ligands, which is present in most practical drug discovery scenarios, using ligands from BindingDB instead of ZINC250k as the starting population for the genetic algorithm. Third, we fine-tune a version of Llama to better modify existing ligands towards higher activity, and find that its use in MOLLEO significantly improves the quality of generated ligands over the base Llama model. We demonstrate our results on the receptor tyrosine kinase c-MET, a crucial protein that drives the growth of various human cancers.

1 Introduction

Large Language Models (LLMs) have gained recent interest for their ability to make significant optimizations and advancements in scientific areas. This is perhaps most notable in the recent AlphaEvolve [1], an evolutionary algorithm which used LLMs to progressively improve the quality of a generated algorithm. It successfully developed state-of-the-art algorithms for multiple problems in mathematics and computer science.

However, studies on applying LLMs to the field of small-molecule generation for drug discovery have been relatively limited. Most previous work in machine learning for small-molecule drug design have focused on VAEs [2, 3, 17], diffusion models [14, 9, 26], reinforcement learning [12, 5, 16], and other generative frameworks [27]. These methods are often guided by a cheap oracle like AutoDock [22] which reports the binding affinity of generated compounds to a particular protein target; however, it is known to be inaccurate in reflecting actual experimental activity [7]. Thus, most current frameworks struggle with generating compounds that are both strong binders and realistic candidates for experimental activity.

Recently, LLMs have begun to garner interest in the field as a generative framework, showing promise in generating strong, drug-like ligands. LLMs hold the distinct advantage of being implicitly aware of how chemistry is typically done (e.g. common reactions, lead optimization techniques, etc.), giving them great potential in problems related to chemical discovery [24]. We explore this potential by introducing a set of novel optimizations to current methods that significantly improve the effectiveness

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

of LLMs in protein-ligand optimization. We build on the notable previous work MOLLEO [23] (MIT License), an evolutionary algorithm that incorporates LLMs as an operator in its optimization cycle. First, we replace the AutoDock [22] oracle in MOLLEO with the new biomolecular foundation model Boltz-2 [19] (MIT License). We demonstrate that this relatively cheap oracle significantly improves the quality of generated ligands, as measured by the gold-standard Absolute Binding Free Energy (ABFE) [4], over AutoDock docking. To our knowledge, this is the first work to demonstrate Boltz-2 as a superior oracle for generative frameworks over the currently standard molecular docking. Second, we change the starting population in MOLLEO to sample from the large protein-ligand database BindingDB [15], sharply focusing the algorithm toward the exploitation of existing strong binders. Finally, we again utilize BindingDB to create a specially-generated dataset, aimed toward guiding models to generate higher quality molecules based on provided context. We use this dataset to fine-tune a small LLM and significantly improve the quality of its generations within the MOLLEO framework.

To summarize, our contributions are as follows:

- We demonstrate the effectiveness of Boltz-2 [19] as a cheap oracle within a molcular generation framework, showing clear advantages over AutoDock [22].
- We improve the quality of generations throughout the MOLLEO [23] genetic algorithm by optimizing its starting population.
- We introduce a novel post-training framework that produces datasets aimed toward improving the quality of LLM-generated compounds, and demonstrate its effectiveness by fine-tuning a small LLM and significantly improving its molecule generations.

2 Methodology

2.1 Boltz-2 as a fitness evaluator

Boltz-2 [19] is a new biomolecular foundation model that utilizes a transformer-based, SE(3) equivariant architecture to carry out 3D structure prediction, and subsequent binding affinity estimation on the predicted structure. This framework approaches the accuracy of much more expensive gold-standard free energy methods like Absolute Binding Free Energy (ABFE), at around 1/1000 the cost. In this work, we replace the docking-based fitness metric used in MOLLEO with the much more accurate affinity predictions from Boltz-2, which adds only minimal computational cost.

2.2 Optimizing starting population of MOLLEO

MOLLEO [23] is an evolutionary algorithm (EA) for small molecule generation that builds upon the Graph-GA algorithm [11]. It utilizes LLMs to make structural modifications (crossovers and mutations) to a starting population of ligands, gradually improving the fitness of the ligand population. It has demonstrated strong results for protein-ligand optimization on 3 protein targets. The original algorithm uses a random sample of ZINC 250k [21] compounds as the initial population, which are not designed for any particular target.

Our optimization to MOLLEO involves employing the large protein-ligand database BindingDB [15] instead of ZINC 250k to give the MOLLEO algorithm a significantly stronger starting point. With BindingDB, we are able to selectively pick strong known binders to the particular target that we are interested in (c-MET, in our case), comprising an initial population that promises much greater experimental activity. This focuses the algorithm more on the exploitation of existing strong binders (which are often known during drug discovery projects), instead of exploration based off non target-specific initial molecular structures.

To form this starting pool, we first separate the set of BindingDB ligands corresponding to c-MET into clusters using the Butina algorithm, which creates clusters based on the pairwise Tanimoto similarity of all ligands to each other. We use a distance threshold of 0.4. This ensures that ligands are structurally diverse across different clusters, because very similar ligands are all grouped within the same clusters. This is desirable because we want the algorithm to have the potential to create entirely novel molecules through crossovers between diverse ligands. From there, we take the ligand with the best binding affinity from each cluster. After sorting this list of ligands, we provide the top n ligands in binding affinity as the starting population for MOLLEO.

2.3 Fine-tuning with BindingDB

To form a synthetic dataset for supervised fine-tuning (SFT) using BindingDB, we begin in a very similar way to the clustering method described above. We form n distinct clusters from the ligand pool for a protein target, using Butina clustering with distance threshold 0.4. Then within each cluster, we first sort the ligands by affinity, then form a series of "ligand chains". This is done by first picking a weak affinity ligand, then repeatedly selecting a ligand with binding affinity stronger than the current by some threshold (we used 0.5 kcal/mol). The result is that for each cluster, we end up with several chains of ligands that are ordered with increasing binding affinity. All ligands within a chain are guaranteed to be relatively similar in structure due to the clustering, and the affinity threshold between ligands in the chain accounts for experimental variance in the BindingDB results. The point of doing this is to form a dataset where an LLM learns to make decisions that change a weak-binding ligand into a guaranteed strong-binding one as it moves down the chain during training. The changes are usually minimal due to the structural similarity, so each chain represents a somewhat realistic series of modifications that a chemist might make.

We form the dataset itself using a strong LLM; we employ GPT 4.1 nano [18] for cost efficiency. The exact details of how we utilize the ligand chains to form an SFT dataset can be found in Appendix B.1. This dataset is used in a classic supervised fine-tuning run. We progress until we observe the validation loss reach a plateau. We apply this training framework to the relatively small Llama-3.1-8B-Instruct model [6]. Details about the training process are provided in Appendix B.2

3 Results

Boltz-2 Table 1 compares the Absolute Binding Free Energy (ABFE) scores [4] of ligands generated using Boltz-2 [19] and AutoDock docking [22] as the fitness evaluator for MOLLEO. Our setup for ABFE calculations is provided in Appendix A.2. We also compare against MF-LAL [3], a VAE-based generative method that specifically focuses on achieving strong ABFE results with a multi-fidelity optimization approach.

Table 1: ABFE Results for Autodock, Boltz-2, and MF-LAL

Method	Count	Mean $\pm$ SD	1st	2nd	3rd
MF-LAL	10	-4.3 $\pm$ 3.7	-8.7	-8.5	-8.3
MOLLEO (AutoDock)	10	-3.6 $\pm$ 5.0	-12.8	-8.7	-8.0
MOLLEO (Boltz-2)	10	**-7.3** $\pm$ 5.5	**-14.0**	**-12.7**	**-11.8**

For this, we take the top 10 best molecules generated from each run according to the respective oracle. We can see that MOLLEO using AutoDock does not beat the MF-LAL baseline, but simply incorporating Boltz-2 as the oracle improves the results drastically. MOLLEO with Boltz-2 results in compounds with much better ABFE scores than with AutoDock, having a difference in mean ABFE score of -3.7 ($p = 0.085$ from independent Student's t-test; rigorous significance is not **yet** reached due to small sample size and computational/time constraints). Further analysis of the correlation between Boltz-2, docking, and ABFE can be found in Appendix A.1, showing that Boltz-2, but not docking, is strongly correlated with ABFE scores. Our main takeaway from these results is that Boltz-2 is much better than AutoDock as an oracle for producing compounds with high ABFE scores.

Results of MOLLEO optimization Table 2 shows the results of the starting-population optimization to the MOLLEO algorithm described above, as well as the results of the small fine-tuned Llama model. Every MOLLEO run terminates at 1000 oracle calls, with an initial population (and population size) of 120 and offspring size of 70. We report all results with Boltz-2 calculated binding affinities instead of ABFE due to computational constraints, but rely on the demonstrated correlation between Boltz-2 and ABFE (Appendix A.1) to support the validity of relative differences between methods.

For the calculated mean, we first remove all compounds generated by the default crossover/mutation operators in MOLLEO, which the algorithm falls back on if the LLM happens to generate an invalid molecule. We do this to remove a source of variance between runs so that we can sharply focus on the differences between performance of the LLMs themselves. We then Butina cluster the resulting pool

Table 2: Boltz-2 Scores for Various MOLLEO Configurations

LLM	Starting Dataset	Mean $\pm$ SD	# < Threshold	% Valid Generations
GPT-4.1-mini	ZINC 250K	-11.1 $\pm$ 0.2	7	66.0
GPT-4.1-mini	BindingDB	**-12.0** $\pm$ 0.2	**56**	60.5
BioT5	BindingDB	**-12.0** $\pm$ 0.3	40	**100.0**
Llama (untuned)	BindingDB	-11.2 $\pm$ 0.3	9	9.8
Llama (tuned)	BindingDB	-11.6 $\pm$ 0.2	25	34.3

of generated molecules, then take the best 10 scores that belong to distinct clusters. This way, we more effectively assess the quality of 10 structurally unique generations. Thus, the mean is comprised of the top 10 diverse, LLM-generated compounds for each configuration.

The results show a substantial increase in top Boltz-2 scores (filtered for only-LLM generations) when we change the starting pool to use ligands from BindingDB instead of from ZINC 250k ($p < 0.0001$). Similarly, it shows a significant increase in resulting scores between our fine-tuned small Llama model and the untuned version ($p = 0.0003$). We also measure the number of diverse top generated compounds that exceed an activity threshold of -11 kcal/mol; we see that incorporating BindingDB increases this metric significantly, as does the fine-tuning process for the Llama model. We additionally compare the percentage of valid LLM responses, and find that the fine-tuning process significantly reduces the number of invalid LLM responses. BioT5 [20] is a chemistry-trained LLM with the T5 architecture, used in the original MOLLEO paper. It scores 100% in this metric because it utilizes SELFIES [13], which cannot translate to invalid molecules. We observe it to perform very similarly to GPT-4.1-mini in this setup.

4 Discussion and Conclusion

In this work, we make several improvements to the MOLLEO framework for LLM-based small molecule drug design. We show that Boltz-2 is a better fitness function than docking, producing compounds more likely to show real-world binding. This result is notable given previous concerns that Boltz-2 performs poorly out-of-distribution, and suggests that Boltz-2, instead of docking, should be used as an oracle for other molecular generative models. We also modify the starting population of MOLLEO, resulting in significantly stronger generated structures and molecules throughout the algorithm. Finally, we present a fine-tuning framework that employs BindingDB to create a novel synthetic dataset, which improves the molecular generation abilities of a small Llama 3 model. While we don't yet exceed the state-of-the-art (GPT-4) metrics with our very small fine-tuned model, we demonstrate strong relative improvements, and hypothesize that the same fine-tuning method can be applied to larger models and yield a similar relative increase in performance.

Limitations While our BindingDB approach to MOLLEO demonstrably improves the performance on the c-MET target, we recognize that this is a very well-studied target. For less studied protein targets, this method may be entirely inapplicable if there are not enough **diverse** known binders to comprise a starting population. This also somewhat applies to our fine-tuning framework; however, we demonstrate in Appendix C that we can utilize other ligands from BindingDB still achieve comparable performance. We also do not consider import molecule properties like synthesizeability in this work. However, since MOLLEO supports multi-objective optimization, we plan to explore other desirable properties alongside binding affinity in future work. Additionally, our results for MOLLEO optimizations are reported from one run per configuration; due to the high computational cost of running 1000 Boltz-2 predictions per run (typically 30 hours on a NVIDIA H200), we are unable to report more runs at this time. Statistical significance tests report strong results, but we acknowledge that there may still be high variance between separate optimization runs using the same configuration. We aim to concretely report results from a higher quantity of runs in future work.

Impact Statement We recognize that improved molecular optimization frameworks may be utilized to generate chemically dangerous compounds. However, since our work does not consider complicated properties and requirements for generation and synthesis of harmful compounds, our contribution is not imminently problematic in this direction.

References

[1] Stephen F. Altschul et al. "Basic local alignment search tool". In: *Journal of Molecular Biology* 215.3 (Oct. 1990), pp. 403–410. ISSN: 0022-2836. DOI: 10.1016/s0022-2836(05)80360-2. URL: http://dx.doi.org/10.1016/S0022-2836(05)80360-2.

[2] Peter Eckmann et al. *LIMO: Latent Inceptionism for Targeted Molecule Generation*. 2022. arXiv: 2206.09010 [cs.LG]. URL: https://arxiv.org/abs/2206.09010.

[3] Peter Eckmann et al. *MF-LAL: Drug Compound Generation Using Multi-Fidelity Latent Space Active Learning*. 2025. arXiv: 2410.11226 [cs.LG]. URL: https://arxiv.org/abs/2410.11226.

[4] Mudong Feng, Germano Heinzelmann, and Michael K. Gilson. "Absolute binding free energy calculations improve enrichment of actives in virtual compound screening". In: *Scientific Reports* 12.1 (Aug. 2022). ISSN: 2045-2322. DOI: 10.1038/s41598-022-17480-w. URL: http://dx.doi.org/10.1038/s41598-022-17480-w.

[5] Tianfan Fu et al. *Reinforced Genetic Algorithm for Structure-based Drug Design*. 2022. arXiv: 2211.16508 [q-bio.QM]. URL: https://arxiv.org/abs/2211.16508.

[6] Aaron Grattafiori et al. *The Llama 3 Herd of Models*. 2024. arXiv: 2407.21783 [cs.AI]. URL: https://arxiv.org/abs/2407.21783.

[7] Koichi Handa et al. "On the difficulty of validating molecular generative models realistically: a case study on public and proprietary data". In: *Journal of Cheminformatics* 15.1 (Nov. 2023). ISSN: 1758-2946. DOI: 10.1186/s13321-023-00781-1. URL: http://dx.doi.org/10.1186/s13321-023-00781-1.

[8] Germano Heinzelmann and Michael K. Gilson. "Automation of absolute protein-ligand binding free energy calculations for docking refinement and compound evaluation". In: *Scientific Reports* 11.1 (Jan. 2021). ISSN: 2045-2322. DOI: 10.1038/s41598-020-80769-1. URL: http://dx.doi.org/10.1038/s41598-020-80769-1.

[9] Emiel Hoogeboom et al. *Equivariant Diffusion for Molecule Generation in 3D*. 2022. arXiv: 2203.17003 [cs.LG]. URL: https://arxiv.org/abs/2203.17003.

[10] Edward J. Hu et al. *LoRA: Low-Rank Adaptation of Large Language Models*. 2021. arXiv: 2106.09685 [cs.CL]. URL: https://arxiv.org/abs/2106.09685.

[11] Jan H. Jensen. "A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space". In: *Chemical Science* 10.12 (2019), pp. 3567–3572. ISSN: 2041-6539. DOI: 10.1039/c8sc05372c. URL: http://dx.doi.org/10.1039/C8SC05372C.

[12] Woosung Jeon and Dongsup Kim. "Autonomous molecule generation using reinforcement learning and docking to develop potential novel inhibitors". In: *Scientific Reports* 10.1 (2020), p. 22104. DOI: 10.1038/s41598-020-78537-2. URL: https://doi.org/10.1038/s41598-020-78537-2.

[13] Mario Krenn et al. "Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation". In: *Machine Learning: Science and Technology* 1.4 (Oct. 2020), p. 045024. ISSN: 2632-2153. DOI: 10.1088/2632-2153/aba947. URL: http://dx.doi.org/10.1088/2632-2153/aba947.

[14] Seul Lee, Jaehyeong Jo, and Sung Ju Hwang. *Exploring Chemical Space with Score-based Out-of-distribution Generation*. 2023. arXiv: 2206.07632 [q-bio.BM]. URL: https://arxiv.org/abs/2206.07632.

[15] T. Liu et al. "BindingDB: a web-accessible database of experimentally determined protein-ligand binding affinities". In: *Nucleic Acids Research* 35.Database (Jan. 2007), pp. D198–D201. ISSN: 1362-4962. DOI: 10.1093/nar/gkl999. URL: http://dx.doi.org/10.1093/nar/gkl999.

[16] Eyal Mazuz et al. "Molecule generation using transformers and policy gradient reinforcement learning". In: *Scientific Reports* 13.1 (May 2023). ISSN: 2045-2322. DOI: 10.1038/s41598-023-35648-w. URL: http://dx.doi.org/10.1038/s41598-023-35648-w.

[17] Juhwan Noh et al. "Path-Aware and Structure-Preserving Generation of Synthetically Accessible Molecules". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 17–23 Jul 2022, pp. 16952–16968. URL: https://proceedings.mlr.press/v162/noh22a.html.

[18] OpenAI et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL]. URL: https://arxiv.org/abs/2303.08774.

[19] Saro Passaro et al. "Boltz-2: Towards Accurate and Efficient Binding Affinity Prediction". In: (June 2025). DOI: 10.1101/2025.06.14.659707. URL: http://dx.doi.org/10.1101/2025.06.14.659707.

[20] Qizhi Pei et al. *BioT5: Enriching Cross-modal Integration in Biology with Chemical Knowledge and Natural Language Associations*. 2024. arXiv: 2310.07276 [cs.CL]. URL: https://arxiv.org/abs/2310.07276.

[21] Teague Sterling and John J. Irwin. "ZINC 15 – Ligand Discovery for Everyone". In: *Journal of Chemical Information and Modeling* 55.11 (Nov. 2015), pp. 2324–2337. ISSN: 1549-960X. DOI: 10.1021/acs.jcim.5b00559. URL: http://dx.doi.org/10.1021/acs.jcim.5b00559.

[22] Oleg Trott and Arthur J. Olson. "AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading". In: *Journal of Computational Chemistry* 31.2 (June 2009), pp. 455–461. ISSN: 1096-987X. DOI: 10.1002/jcc.21334. URL: http://dx.doi.org/10.1002/jcc.21334.

[23] Haorui Wang et al. *Efficient Evolutionary Search Over Chemical Space with Large Language Models*. 2025. arXiv: 2406.16976 [cs.NE]. URL: https://arxiv.org/abs/2406.16976.

[24] Andrew D. White. "The future of chemistry is language". In: *Nature Reviews Chemistry* 7.7 (May 2023), pp. 457–458. ISSN: 2397-3358. DOI: 10.1038/s41570-023-00502-0. URL: http://dx.doi.org/10.1038/s41570-023-00502-0.

[25] Yaowei Zheng et al. *LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models*. 2024. arXiv: 2403.13372 [cs.CL]. URL: https://arxiv.org/abs/2403.13372.

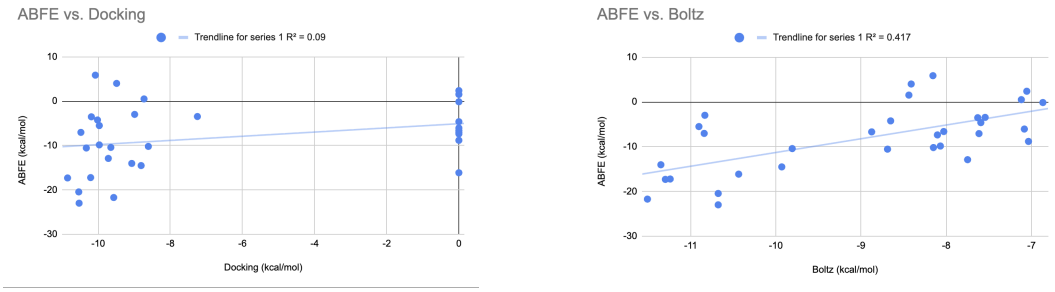
[26] Xiangxin Zhou et al. *DecompOpt: Controllable and Decomposed Diffusion Models for Structure-based Molecular Optimization*. 2024. arXiv: 2403.13829 [q-bio.BM]. URL: https://arxiv.org/abs/2403.13829.

[27] Yiheng Zhu et al. *Sample-efficient Multi-objective Molecular Optimization with GFlowNets*. 2023. arXiv: 2302.04040 [cs.LG]. URL: https://arxiv.org/abs/2302.04040.

A Boltz-2 and ABFE

A.1 Correlation Analysis

Here, we show additional analysis of the correlation between Boltz-2 scores and ABFE scores. We take 32 compounds for c-MET, 16 of which are known binders, and 16 of which are presumed inactives. We calculate the ABFE, Boltz-2, and AutoDock docking binding affinities for all 32 compounds.



(a) AutoDock vs ABFE Scores

(b) Boltz-2 vs ABFE Scores

Figure 1: Comparison of Correlation between AutoDock & ABFE and Boltz-2 & ABFE

From Figure 1, we see that ABFE and AutoDock docking show $r^2 = 0.09$ among the active compounds, while ABFE and Boltz-2 show $r^2 = 0.42$. As an oracle nearly 1000x less computationally expensive than ABFE, Boltz-2 shows exceptional correlation with ABFE, especially in comparison to docking. Furthermore, we calculate the ROC-AUC score for Boltz-2 and docking, to see how well they can separate binders from non-binders. Boltz-2 scores 0.95 for this metric, while docking scores 0.84.

A.2 ABFE Setup

For our ABFE calculations, we utilize the following Binding Affinity Tool `BAT.py` [8] (MIT License) repository: `https://github.com/GHeinzelmann/BAT.py`. We simulate using OpenMM and the standard SDR method. For calculations of molecules generated by docking as the oracle, we use the ligand pose generated by AutoDock as the starting pose for the calculation. For calculations of molecules generated by Boltz-2 as the oracle, we use the Boltz-2 predicted ligand pose as the starting pose. We separate the source of the poses to avoid potential bias toward one particular oracle in the ABFE calculation.

Our simulation steps parameters are as follows:
eq_steps1 = 500000 (Number of steps for equilibration gradual release)
eq_steps2 = 15000000 (Number of steps for equilibration after release)

m_steps1 = 500000 (Number of steps per window for component m (equilibrium))
m_steps2 = 1000000 (Number of steps per window for component m (production))

n_steps1 = 500000 (Number of steps per window for component n (equilibrium))
n_steps2 = 1000000 (Number of steps per window for component n (production))

e_steps1 = 250000 (Number of steps per window for component e (equilibrium))
e_steps2 = 500000 (Number of steps per window for component e (production))

v_steps1 = 500000 (Number of steps per window for component v (equilibrium))
v_steps2 = 1000000 (Number of steps per window for component v (production))

On 4 NVIDIA H200 GPUs, one ABFE calculation typically takes us around 16 hours to complete.

B Additional LLM Fine-tuning Information

B.1 LLM Prompts For Dataset Formation

This section provides the exact prompts used to create the supervised fine-tuning dataset used in this work.

We first recap the full process of obtaining the dataset through Figure 2.

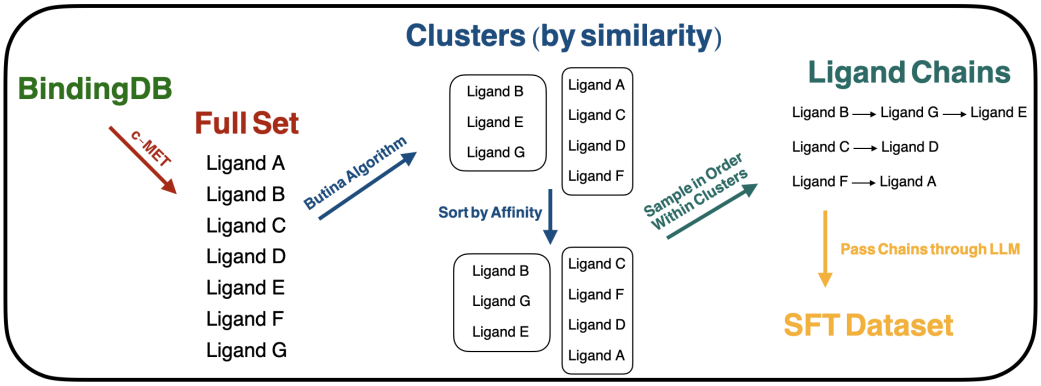


Figure 2: Full Process for Preparing Supervised Fine-Tuning Dataset

Consider one of the ligand chains formed by the clustering-sorting process. For each ligand/position in the chain, we first ask the LLM to generate a summary based on all the past (weaker affinity) ligands in the chain. This summary is used in the input for SFT, simulating the information the LLM might receive for an optimization step.

Prompt to generate the summary of past ligand modifications used in the input for SFT

You are a chemistry-aware assistant that is collaborating with me on generating a ligand for a protein with high binding affinity. Below is a chronological history of past ligands you've generated. Provide a summary of changes and modifications you've made so far in regards to the ligand structure and how it impacts the binding affinity; the goal is to give context about past iterations to another agent. Be sure to explictly output the SMILES of every past ligand. Do not provide any suggestions for future generations at this time. Keep your response relatively short.
SMILES: Affinity
SMILES: Affinity
SMILES: Affinity
...

Where we input all previous ligands and their binding affinities in the chain as *SMILES: Affinity*. The generated summary is placed into the following format, which becomes the full input for SFT:

Full SFT Input

We are collaborating on generating a ligand for a protein with high binding affinity. I will give you the output from docking software after each of your attempts. Provided below is a brief summary of past ligand modifications:
*****GENERATED SUMMARY****
First describe what you have learned from the above summary. Then based on that knowledge, generate a ligand that can bind to this protein with high binding affinity. Ensure that your generation is unique and is not found within the provided data. Follow this format for your final answer: \\box{MOLECULE}, where MOLECULE is your proposed ligand in SMILES format.
SMILES: Affinity
SMILES: Affinity
SMILES: Affinity
...

After this, we ask the LLM to generate reasoning that might lead an agent to generate the next (stronger affinity) ligand in the chain. This becomes the full desired output for SFT. In essence, we want to tune the model to carry out what we know to be a strong modification decision for a particular protein target based off information provided in the input.

Prompt to generate the full output for SFT:

We are collaborating on generating a ligand for a protein with high binding affinity. I will give you the output from docking software after each of your attempts. Provided below is a brief summary of past ligand modifications:
*****GENERATED SUMMARY****
Based on the summary, first pretend that you are thinking about what modifications to make to create a stronger molecule. In the end, choose to generate the molecule *NEXT_LIGAND*. Do not mention this molecule until you've explained what decisions and specific structural modifications would lead to this new molecule. It's crucial that you pretend to have came up with this generation yourself after some brief analysis of the provided summary. Provide the final ligand explictly in this exact format, character by character: \\box{*NEXT_LIGAND*}. Keep your response relatively short.
SMILES: Affinity
SMILES: Affinity
SMILES: Affinity
...

Notice that the first half of this prompt is the same as the actual input for SFT. However, we instead tell the LLM to pretend like it's doing the generation action, using the predetermined *NEXT_LIGAND* as the guide for its explanation, where *NEXT_LIGAND* is the next (higher affinity) ligand in the ligand chain. This results in an SFT output that not only contains the desired next ligand, but also provides a sample explanation for how you might modify previous ligands to create the new one. This is precisely what we want for the SFT dataset, as we want a model to learn how to make the best modifications for any given input to optimize toward a point of higher affinity. Here is a sample LLM output for this prompt:

Sample Full SFT Output

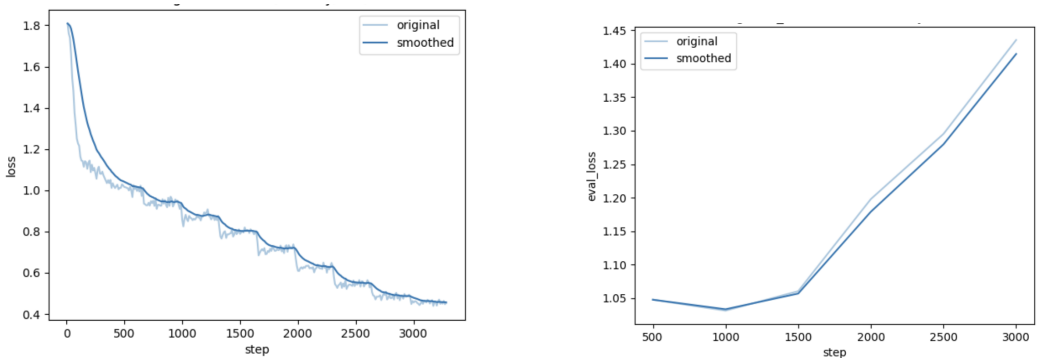
Considering the previous ligand's balance of hydrophobic aromatic rings, heterocycles, and hydrogen-bonding groups, I think introducing a more rigid and cyclic structure could enhance binding affinity. Replacing flexible linker regions with nitrogen-containing heterocycles might improve interactions through additional hydrogen bonds and electrostatic contacts. Additionally, modifying substituents to include carbonyl groups could further strengthen hydrogen bonding capacity. Incorporating a carbamate moiety at a strategic position can also influence binding strength. After analyzing these features, I realize that a cyclic structure incorporating these elements—specifically, a fused ring system with carbamate and aromatic groups—optimize interactions with the protein binding site. Therefore, I plan to generate a ligand with these features, resulting in the target molecule: \\box{*NEXT_LIGAND*}.

B.2 SFT Training Process

In this section, we provide information about the training process for our supervised fine-tuned model. We fine-tune using the unified open-source fine-tuning repository LLaMA-Factory [25] (Apache License). We utilize Low-Rank Adaption (LoRA) [10] to train a subset of the model parameters, saving a significant amount of time and computation. We utilize all default hyperparameters from the LlaMA-Factory repository (see the llama3_lora_sft.yaml example file in examples/train_lora/), except for modifying the train-validation split to be 0.95/0.05 instead of 0.90/0.10. We train for 10 epochs on a dataset with 2,500 samples, taking around 80 minutes on a NVIDIA H200.

Figure 2 provides the training and validation loss graphs for this process. As is evident from the figure, validation loss drops rapidly (initial model validation loss is not measured here, but we can assume it to be around 1.8 according to the start of the training loss graph), then rather quickly plateaus, and increases rapidly as the model overfits to the relatively small dataset. We let the training continue past the overfitting point just for the chance of any emergent behavior. However, we carefully select the

checkpoint for which the validation loss is at its minimum, which we evaluate to be the checkpoint at step 1,000. We merge the LoRA adapters at this checkpoint into the original base model to obtain the fine-tuned model used in MOLLEO optimization.



(a) Training Loss

(b) Validation Loss

Figure 3: Training and validation loss graphs for supervised fine-tuning. Validation loss appears to not decrease at all, but it's due to the large number of steps before the first evaluation; we can assume evaluation loss starts somewhere near where the training loss started (1.8)

C Demonstration for Lack of BindingDB Ligands

In this section, we demonstrate that for our supervised fine-tuning dataset, we have a workaround in the situation where we are optimizing for a protein target that is not well studied and has few results for experimentally-tested ligand binders.

Our original dataset for c-MET had around 2,500 samples, formed from around 7000 total ligand entries in BindingDB. This is a very small amount for a training dataset, but we still observe a significant drop in validation loss with such a dataset, which is also reflected by our model's performance within the MOLLEO optimization loop. We also formed another dataset comprising of around 30,000 samples. We did this by taking 20 protein targets that we determined to have structural similarities c-MET using the BLASTP tool [1]. By doing this, we expanded our total ligand pool to around 200,000 ligands, which resulted in a dataset of 30,000 samples. We trained the same small Llama model on this dataset, and its performance in MOLLEO is shown in Table 3.

Table 3: Boltz-2 Scores for Llama Tuned on 2.5k vs 30k Datasets

Method	Mean (filtered) $\pm$ SD
Llama (Untuned)	-11.2 $\pm$ 0.3
Llama (2.5 dataset)	**-11.6** $\pm$ 0.2
Llama (30k dataset)	**-11.6** $\pm$ 0.2

We see that increasing the size of the dataset using ligands from adjacent protein target does not change the model performance in any significant way. This shows that the size of the dataset is not necessarily a problem (at least at the size of this Llama model).

Importantly, it also shows that if the desired protein target does not have sufficient ligand entries, we can make up for it by identifying protein targets that are structurally similar to it and use their ligand entries instead. This guarantees some level of similarity in the input ligands, and as demonstrated experimentally, does not hurt performance relative to using a dataset comprised only of target-specific ligands.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our abstract clearly describes our three main optimizations, which are again repeated (in the same order) in the Introduction, Methodology, Results, and Discussion sections. Throughout the paper, we are focused narrowly on these 3 contributions regarding Boltz-2, BindingDB starting population, and our fine-tuned model, and do not deviate from the material introduced in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We dedicate a section of the Discussion and Conclusion section that discusses specific limitations of our methods and when they may not be applicable, as well as the low quantity of runs for our results. We also acknowledge particular limited results throughout the paper, such as current lack of ABFE results leading to low significance and the limitations of our fine-tuning method in its current inability to allow small models to surpass SOTA foundation models.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: We do not report any theoretical results, only concrete experimental ones. Our claims are based off statistical significance tests on these results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: For all 3 contributions, we describe how to recreate the experimental results; we describe our ABFE setup in Appendix A.2, provide detailed descriptions of prompt setups for the BindingDB dataset in Appendix B.1 (as well as our LLM fine-tuning setup and method), and describe exactly how we prune our starting pool from BindingDB in Methodology, including parameters for Butina clustering. We disclose exactly which GPT and Llama models are used, and describe exactly how Boltz-2 is incorporated into the MOLLEO algorithm, also in Methodology. In general, the Methodology and Appendix sections go into great detail about all of our experiment setup, ensuring reproducibility as best we can.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We do not introduce any truly novel code, our optimizations are rooted in modifying an oracle in an existing framework, and beyond that it is several Python scripts that process BindingDB data in a way that is thoroughly described throughout the paper. If accepted, we can break anonymity and include a repository link for these few scripts, but we don't consider it significant enough to include in this submission, especially considering the detail in which the methods are already described in Methodology and in the Appendix.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We report training details for our fine-tuning in Appendix B.2. We describe the method used (SFT with LoRA), and provide the training and validation results, as well as which checkpoint we utilized for our final model. We describe certain hyperparameters used in training, and link the specific repository used for this process. Further, we report our MOLLEO experimental setup and ABFE setup in the Results and Appendix A.2, respectively.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: For all important results (ABFE scores, mean Boltz-2 in MOLLEO), we report standard deviation as well as the number of data points that went into that mean / stdev. We also run independent t-tests on every result, reporting the p-value in the results. We do not claim statistical significance for cases where the p-value is not sufficiently low enough and acknowledge these limitations.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We report the time of execution for MOLLEO (in the Limitations section in the Conclusion), as well as the time it takes to train our SFT model in Appendix B.2. We also report the time of execution on ABFE results in Appendix A.2. In all sections, we disclose the NVIDIA GPUs used as well as the quantity.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, our research conforms with the NeurIPS code of ethics in every way described. We do not release any harmful data or models, and we include considerations for the societal impact of our work in the Conclusion.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We include an impact statement in our Conclusion that describes the potential dangers of improved molecular frameworks. We ultimately conclude that the scope of this work is too limited to contribute to the potential synthesis of truly harmful/dangerous compounds, mitigating the risk of negative societal impact stemming from this work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

Justification: As we do not release any particular generated datasets and models, we do not suffer from this risk. The methods described to reproduce these datasets only utilize commercial large language models and publicly released protein-ligand databases.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The code we build off (MOLLEO) and the models we employ (BAT.py, Boltz-2) as well as other frameworks (LLaMA-Factory) are all properly cited, credited, and referenced in this work. Their respective licenses are also included.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: No new assets are truly released by this paper. We do not include explicit code and only describe the methodology for the code, which itself is detailed and well documented both in the body of the paper and in the Appendix.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This work does not involve any crowdsourcing or research involving human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This work does not involve any crowdsourcing or research involving human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Appendix B.1 gives a detailed explanation of the entire process regarding using an LLM to generate our SFT dataset. It includes all prompts used, as well as sample outputs from LLMs in response to those prompts. All other uses of LLMs are those in MOLLEO, which use the same prompts as described in the original work and are not novel methodologies.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.

- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.