Guided Multi-objective Generative AI for Structure-based Drug Design

Amit Kadan[*†]
SandboxAQ
amit.kadan@sandboxaq.com

Kevin Ryczko[*†]
SandboxAQ
kevin.ryczko@sandboxaq.com

Erika Lloyd
SandboxAQ
erika.lloyd@sandboxaq.com

Adrian Roitberg
Department of Chemistry
University of Florida
roitberg@ufl.edu

Takeshi Yamazaki[†]
SandboxAQ
takeshi.yamazaki@sandboxaq.com

Abstract

Generative AI has the potential to revolutionize drug discovery. Yet, despite recent advances in deep learning, existing models cannot generate molecules that satisfy all desired physicochemical properties. Herein, we describe IDOLpro, a novel generative chemistry AI combining diffusion with multi-objective optimization for structure-based drug design. Differentiable scoring functions guide the latent variables of the diffusion model to explore uncharted chemical space and generate novel ligands *in silico*, optimizing a plurality of physicochemical properties. We demonstrate our platform's effectiveness by generating ligands with optimized binding affinities (measured by Vina score) and synthetic accessibility on two benchmark sets. IDOLpro produces ligands with binding affinities over 10%-20% higher than the next best state-of-the-art method on each test set, producing more drug-like molecules with generally better synthetic accessibility scores than other methods. We do a head-to-head comparison of IDOLpro against a classic virtual screen of a large database of drug-like molecules. We show that IDOLpro can generate molecules for a range of important disease-related targets with better binding affinity and synthetic accessibility than any molecule found in the virtual screen while being over $100\times$ faster and less expensive to run. On a test set of experimental complexes, IDOLpro is the first to produce molecules with better binding affinities than the experimentally observed ligands. IDOLpro can accommodate other scoring functions (e.g. ADME-Tox) to accelerate hit-finding, hit-to-lead, and lead optimization for drug discovery.

Introduction

The central goal of structure-based drug design (SBDD) is to develop ligands with high binding affinities for a given protein pocket [4]. SBDD is an inverse design problem where the desired properties are known, but designing a molecule to achieve these properties is challenging. Inverse design problems, common in materials [82, 53, 10], chemistry [54, 21, 66], and life sciences [34, 77,

[*]Equal contributions
[†]Corresponding author

Machine Learning for Structural Biology Workshop, NeurIPS 2024.

72], involve two main steps: sampling chemical space, and scoring compounds based on their ability to meet the desired properties. In drug discovery, chemical space is typically sampled by evaluating large databases of drug-like molecules, such as ZINC [27], Enamine [59], or GDB [52]. Despite these databases containing hundreds of billions of molecules, they represent only a fraction of the estimated $10^{20} - 10^{60}$ drug-like molecules in chemical space [49].

Recent literature has introduced several generative deep learning (DL) models to replace the virtual screening of large databases [41, 48, 38, 23, 57, 35]. These models generate molecules with lower average Vina docking scores [69] compared to molecules found via virtual screening while demonstrating greater efficiency. In addition, they can produce molecules not found in existing databases [35]. However, these models can produce unphysical structures [8, 76], or ligands with poor synthetic accessibility [19]. These problems can be addressed by pairing generative models with a feedback loop that conditions generation on measured physicochemical indicators [22, 21, 34]. Conditional generation is becoming a common practice in the drug-discovery literature [35, 57, 13], and can be performed efficiently by making use of gradient information from property predictors [72, 34].

In this work we present IDOLpro (**I**nverse **D**esign of **O**ptimal **L**igands for **Pro**tein pockets), a generative chemistry AI to produce optimized and chemically feasible ligands for a given protein pocket by guiding a state-of-the-art diffusion model. Specifically, we modify the latent variables of the generative model to optimize one or more objectives of interest simultaneously. The objectives considered evaluate properties of the generated ligands. In this report, we consider binding affinity (measured with Vina score) and synthetic accessibility. Our framework is highly modular and can easily incorporate alternative generators and additional scores. All metrics are written in Pytorch [47] and are fully differentiable with respect to the latent variables, allowing for the use of gradient-based optimization strategies to design optimal ligands.

Workflow

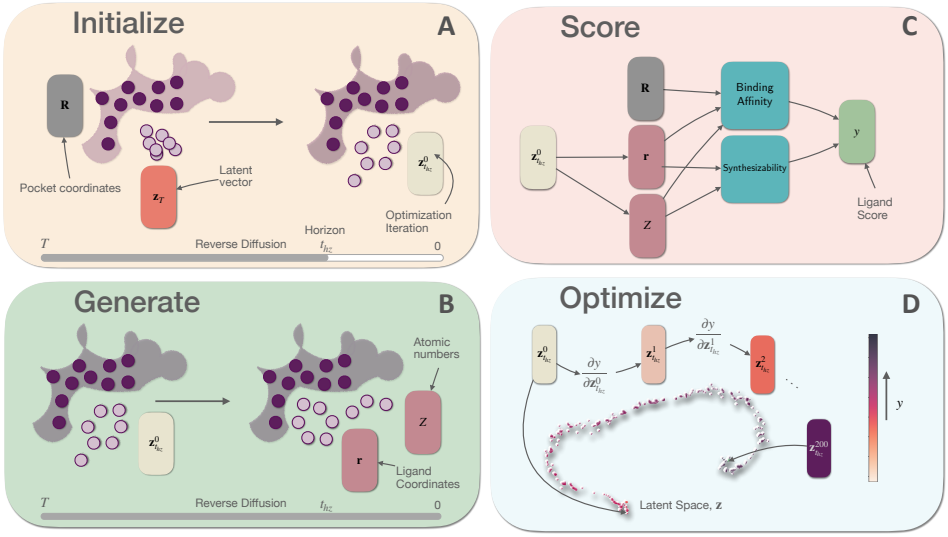

Figure 1: Visual overview. A: Random latent vectors are defined and the reverse diffusion is run for $T \to t_{\text{hz}}$. B: The rest of the reverse diffusion process is completed and a ligand is generated (with coordinates and atomic numbers). C: The ligand is scored by evaluating the binding affinity, synthesizability, or both. D: The ligand score is differentiated with respect to the latent vector $\mathbf{z}_t^0$, and $\mathbf{z}_t^1$ is defined by taking a single optimization step.

IDOLpro optimizes ligands by iteratively adjusting the predictions of a model that directly generates molecules into a given protein pocket. We achieve this by modifying the latent vectors of the generative model using gradients from property predictors. Once optimal ligands are generated, their binding poses are further refined through structural optimization within the pocket. The structural optimization uses the same differentiable scores for evaluating and adjusting the ligands' coordinates with gradients from the property predictors. A visual overview of is shown in Fig. 1.

In this report, we use DiffSBDD [57] as the baseline generative method for predicting ligands within a protein pocket. We use a specific variant of the model, DiffSBDD-Cond, which we found was able to generate ligands within the protein pocket without clashes more reliably than the alternative, DiffSBDD-inpaint. We assess the ability of our framework to discover novel ligands with improved binding affinity and synthetic accessibility. To estimate binding affinity, we have developed a torch-based version of Vina [69] that we call torchvina. When doing structural refinement we also use the ANI2x model [12]. To estimate synthesizability, we train an equivariant neural network [58] model [58] to predict the synthetic accessibility (SA) score reported from RDKit [1] and first proposed in Ref. [14], which we refer to as torchSA. Further details about our methods can be found in the Supplementary Information [30].

Datasets: We assess the performance of our platform on three different tests sets containing protein-ligands pairs – a subset of CrossDocked [16], a subset of the Binding MOAD (Mother of all Databases) [25], and on a test set first proposed in Ref. [17] consisting of disease-related proteins, which we refer to as the RGA test set. The CrossDocked test set contains 100 pocket-ligand pairs derived via re-docking ligands to non-cognate receptors with smina [32], and has been used to validate the performance of tools in several other papers [41, 48], including DiffSBDD [57]. The Binding MOAD contains 130 high resolution (<2.5 Å) experimentally derived pocket-ligand pairs extracted from the Protein Data Bank (PDB), and was also used to assess the performance of DiffSBDD. The RGA test set contains 10 experimentally derived disease-related protein targets with associated ligands. Targets in this test set include G-protein coupling receptors (GPCRs), kinases from the DUD-E dataset [43], and the SARS-CoV-2 main protease [80].

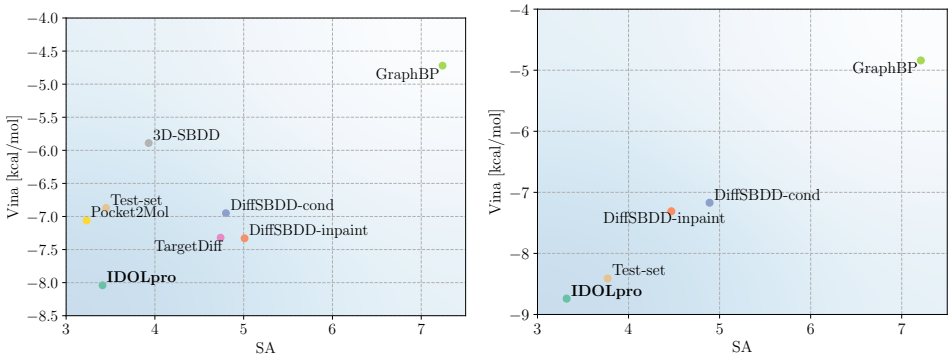

Figure 2: Performance of DL tools on two benchmark test sets. The scatter plot shows the average Vina and SA score for each method for targets in CrossDocked (left), and the Binding MOAD (right). IDOLpro is at the bottom left of each scatter plot, showing it can co-optimize Vina and SA for generated ligands.

Results

Comparison to deep learning: We compare our full pipeline, which includes both latent vector optimization and structural refinement, to other deep learning tools on the CrossDocked and Binding MOAD test sets. For each protein pocket, we use our platform to generate 100 optimized ligands. In addition to Vina score, top-10% Vina score, and SA score, we also report QED (quantitative estimate of drug-likeness) [6]. These metrics are evaluated across six other DL tools in the literature: 3D-SBDD [41], Pocket2Mol [48], GraphBP [38], TargetDiff [23], DiffSBDD-Cond, and DiffSBDD-inpaint [57] whose results are taken from Ref. [57] and shown for comparison with our worflow in Table S4. The performance of the models on Vina and SA is visualized in Fig. 2.

For CrossDocked, IDOLpro achieves improved Vina scores relative to other DL tools, with a 0.71 kcal/mol improvement in average Vina score and 1.03 kcal/mol improvement in top-10% Vina score compared to the next best tool in the literature, DiffSBDD-inpaint. Despite not optimizing for it directly, we find that IDOLpro achieves the best QED on the two benchmarks. IDOLpro ranks second for producing molecules with good SA scores, showing the ability of our tool to perform multi-objective optimization. Despite needing to run an entire optimization procedure for each ligand, IDOLpro is computationally tractable, achieving run times competitive with two other tools – TargetDiff and Pocket2Mol, while being faster than 3D-SBDD.

	Method	Vina [kcal/mol]	$\text{Vina}_{10\%}$ [kcal/mol]	SA	QED
CrossDocked	Test Set	-6.87 ± 2.32	-	3.45 ± 1.26	0.48 ± 0.20
	3D-SBDD [41]	-5.89 ± 1.91	-7.29 ± 2.34	3.93 ± 1.26	0.50 ± 0.17
	Pocket2Mol [48]	-7.06 ± 2.80	-8.71 ± 3.18	$\mathbf{3.23 \pm 1.08}$	0.57 ± 0.16
	GraphBP [38]	-4.72 ± 4.03	-7.17 ± 1.40	7.24 ± 0.81	0.50 ± 0.12
	TargetDiff [23]	-7.32 ± 2.47	-9.67 ± 2.55	4.74 ± 1.17	0.48 ± 0.20
	DiffSBDD-cond [57]	-6.95 ± 2.06	-9.12 ± 2.16	4.80 ± 1.17	0.47 ± 0.21
	DiffSBDD-inpaint [57]	-7.33 ± 2.56	-9.93 ± 2.59	5.01 ± 1.08	0.47 ± 0.18
	IDOLpro	$\mathbf{-8.04 \pm 2.55}$	$\mathbf{-10.96 \pm 3.02}$	3.41 ± 0.70	$\mathbf{0.63 \pm 0.06}$
MOAD	Test Set	-8.41 ± 2.03	-	3.77 ± 1.08	0.52 ± 0.17
	GraphBP [38]	-4.84 ± 2.24	-6.63 ± 0.95	7.21 ± 0.81	0.51 ± 0.11
	DiffSBDD-cond [57]	-7.17 ± 1.89	-9.18 ± 2.23	4.89 ± 1.08	0.44 ± 0.20
	DiffSBDD-inpaint [57]	-7.31 ± 4.03	-9.84 ± 2.18	4.47 ± 1.08	0.54 ± 0.21
	IDOLpro	$\mathbf{-8.74 \pm 2.59}$	$\mathbf{-11.23 \pm 3.12}$	$\mathbf{3.32 \pm 0.66}$	$\mathbf{0.63 \pm 0.08}$

Table 1: Evaluation of DL tools on targets from the CrossDocked and Binding MOAD datasets. The average, and standard deviation of each metric across the protein pockets in each dataset is reported. The top performing model on each metric is bolded in the corresponding column. Numbers for other models are taken from Ref. [57].

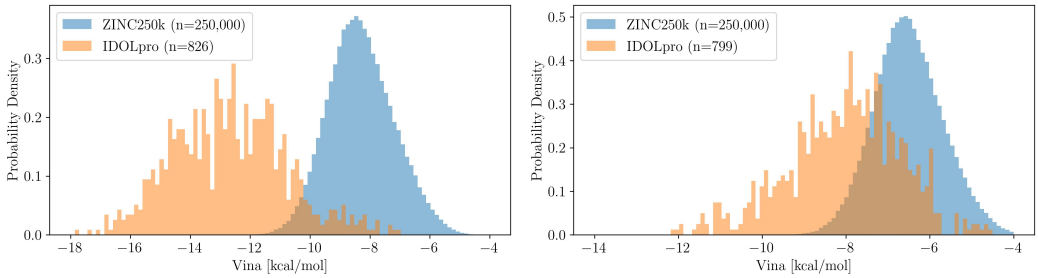

Figure 3: Comparison of our platform to virtually screening ZINC250K. Left: The distribution of Vina scores for generated and screened ligands for EGFR, an important oncology target (PDB ID 2rgp). Right: The distribution of Vina scores for generated and screened ligands for the SARS-Cov-2 main protease (PDB ID 7l11).

Method	N_{ligands}	Time [h]	Cost [$]
IDOLpro	47.20 ± 49.73	0.86 ± 0.71	0.87 ± 0.72
Virtual Screen	$250,000$	160.90 ± 24.12	109.41 ± 16.40

Table 2: Comparison of using our platform to find improved ligands relative to a virtual screen of ZINC250K. We note the average number of ligands, time, and cost it takes for our platform to find a ligand with both better Vina and SA than the best ligand from ZINC250K.

For the Binding MOAD test set, the advantage of IDOLpro is more pronounced, with a 1.43 kcal/mol improvement in average Vina score, and 1.40 kcal/mol improvement in top-10% Vina score compared to DiffSBDD-inpaint. In particular, IDOLpro is the first DL tool to generate molecules with a better average Vina score than those of the reference molecules in the Binding MOAD test set. This is noteworthy, because unlike molecules in CrossDocked, molecules in the Binding MOAD were derived through experiment. Out of the four methods compared, IDOLpro also achieves the best SA, improving upon the next best method by 1.15, while also achieving the best QED. The time to generate a single ligand for a protein pocket in the Binding MOAD test set is slower than for the CrossDocked test set, reflecting DiffSBDD's slowdown in generating molecules for targets in this test set.

IDOLpro is able to find ligands that improve upon both the Vina and SA scores relative to the reference ligand for 99/100 targets in the CrossDocked test set, and 126/130 targets for the Binding MOAD test set. In all of these cases we generate molecules with significantly better SA scores than the reference, but fail to find a molecule with a better Vina score. We believe this could be solved by re-weighting the optimization objective to more heavily favour Vina score. Overall our results show that IDOLpro can effectively co-optimize multiple objectives, generating ligands with state-of-the-art

Method	Vina [kcal/mol]	$\text{Vina}_{10\%}$ [kcal/mol]	SA	$\text{SA}_{10\%}$
Scaffolds	-4.65 $\pm$ 2.09	-	3.06 $\pm$ 1.40	-
Test Set	-5.58 $\pm$ 2.32	-	3.67 $\pm$ 1.23	-
IDOLpro	-7.17 $\pm$ 2.36	-8.96 $\pm$ 2.57	4.12 $\pm$ 1.10	2.90 $\pm$ 1.12

Table 3: Results for scaffold fixing on the 71 crossdocked data points with identifiable scaffolds using RDKit's Bemis-Murcko scaffold.

Vina and synthetic accessibility scores on two test sets. Improving other metrics is straightforward, simply requiring a differentiable score for evaluating the desired metric.

Comparison to virtual screening: We compare IDOLpro's ability to generate promising ligands against the virtual screening of the ZINC250K database, a collection of 250,000 commercially available compounds [22]. Using the 10 protein pockets from the RGA test set, we assess how rapidly we can generate a ligand with both superior binding affinity (Vina score) and synthetic accessibility (SA score) compared to the best ligand identified from the ZINC250K screening. To do the screening, we use QuickVina2 [3] to dock molecules to each of the 10 target protein pockets on an AWS compute-optimized instance with 8 CPU cores. We then use IDOLpro to generate optimized ligands, making note of the number of ligands, time, and cost. Results are shown in Table 2, where cost is based on the AWS pricing for the requested instances.

Screening ZINC250K using QuickVina2 takes an average of ≈ 161 hours per protein pocket, while IDOLpro is able to find a ligand with a better Vina and SA score than the virtual screen in under an hour (≈ 52 minutes) on average. This translates to $\approx 187\times$ speedup in terms of time, and $\approx 126\times$ reduction in cost. For 4/10 cases, IDOLpro is able to find a ligand with better Vina and SA score within the first 10 ligands generated, and for 9/10 cases within the first 100 ligands generated. For a single case (PDB ID 3eml) surpassing the performance of the virtual screen requires generating 152 ligands, taking 2.5 hours. This translates into a $> 60\times$ speedup in terms of time, and $> 40\times$ reduction in cost.

In general, for a given protein pocket, IDOLpro generates ligands with significantly better binding affinities than those found by virtually screening ZINC250K. In Fig. 3, we plot the distribution of Vina scores for both the generated ligands, and those found when screening ZINC250K for 2 of the targets from the RGA test set: 2rgp, a protein who's over-expression has been associated with human tumour growth [74], and 7l11 – the SARS-CoV-2 main protease [79].

Lead optimization: In addition to de novo generation, IDOLpro supports lead optimization by refining known ligands. This process involves fixing a large part of the molecule (the scaffold) while optimizing the remaining portion [26, 7]. Using DiffSBDD [57], we apply an inpainting method to incorporate the scaffold into the generation process by fixing certain atoms of the reference ligand. To test the framework's capability, we use protein pockets from the CrossDocked test set. For each reference ligand, RDKit is used to identify the Bemis-Murcko scaffold [5]. Targets are removed if no scaffold is found, if the scaffold makes up more than 90% of the ligand, or if it contains atoms unsupported by the ANI2x model [12]. The lead optimization results for the remaining 71 protein-scaffold pairs are shown in Table 3 and are compared to the original ligands and their scaffolds. We find that the average Vina scores of the optimized ligands greatly improve upon the seed scaffolds (2.52 kcal/mol) and the reference ligands (1.59 kcal/mol) from the test set. Although the average SA is higher, both the top-10% SA score and the top-10% Vina are significantly improved.

Conclusion

We developed a platform that generates optimized ligands for specific protein pockets by constructing a computational graph linking the latent variables of a diffusion model to key drug discovery metrics. Through gradient-based optimization, it improves Vina and synthetic accessibility (SA) scores simultaneously. The platform achieves the lowest Vina scores compared to other leading machine learning methods, with significantly improved the SA and QED metrics on both the CrossDocked and Binding MOAD test sets. It excels in hit-finding by identifying molecules with superior Vina and SA scores in less time and at lower cost compared to traditional virtual screening, and also supports lead optimization. Future plans include integrating additional metrics like toxicity and advanced binding affinity metrics such as free energy perturbation (FEP).

References

[1] Rdkit: Open-source cheminformatics. `https://www.rdkit.org`.

[2] Sungsoo Ahn, Junsu Kim, Hankook Lee, and Jinwoo Shin. Guiding deep molecular optimization with genetic exploration. *Advances in neural information processing systems*, 33:12008–12021, 2020.

[3] Amr Alhossary, Stephanus Daniel Handoko, Yuguang Mu, and Chee-Keong Kwoh. Fast, accurate, and reliable molecular docking with quickvina 2. *Bioinformatics*, 31(13):2214–2216, 2015.

[4] Amy C Anderson. The process of structure-based drug design. *Chemistry & biology*, 10(9):787–797, 2003.

[5] Guy W Bemis and Mark A Murcko. The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry*, 39(15):2887–2893, 1996.

[6] G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90–98, 2012.

[7] Hans-Joachim Böhm, Alexander Flohr, and Martin Stahl. Scaffold hopping. *Drug discovery today: Technologies*, 1(3):217–224, 2004.

[8] Martin Buttenschoen, Garrett M Morris, and Charlotte M Deane. Posebusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science*, 15:3130–3139, 2024.

[9] Lowik Chanussot, Abhishek Das, Siddharth Goyal, Thibaut Lavril, Muhammed Shuaibi, Morgane Riviere, Kevin Tran, Javier Heras-Domingo, Caleb Ho, Weihua Hu, et al. Open catalyst 2020 (oc20) dataset and community challenges. *Acs Catalysis*, 11(10):6059–6072, 2021.

[10] François Raymond J Cornet, Bardi Benediktsson, Bjarke Arnskjær Hastrup, Arghya Bhowmik, and Mikkel N Schmidt. Inverse-design of organometallic catalysts with guided equivariant diffusion. In *37th Conference on Neural Information Processing Systems*, 2023.

[11] Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. In *International Conference on Learning Representations*, 2023.

[12] Christian Devereux, Justin S Smith, Kate K Huddleston, Kipton Barros, Roman Zubatyuk, Olexandr Isayev, and Adrian E Roitberg. Extending the applicability of the ani deep learning molecular potential to sulfur and halogens. *Journal of Chemical Theory and Computation*, 16(7):4192–4202, 2020.

[13] Orion Dollar, Nisarg Joshi, Jim Pfaendtner, and David AC Beck. Efficient 3d molecular design with an e (3) invariant transformer vae. *The Journal of Physical Chemistry A*, 127(37):7844–7852, 2023.

[14] Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1:1–11, 2009.

[15] Dakota Folmsbee and Geoffrey Hutchison. Assessing conformer energies using electronic structure and machine learning methods. *International Journal of Quantum Chemistry*, 121(1):e26381, 2021.

[16] Paul G Francoeur, Tomohide Masuda, Jocelyn Sunseri, Andrew Jia, Richard B Iovanisci, Ian Snyder, and David R Koes. Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design. *Journal of chemical information and modeling*, 60(9):4200–4215, 2020.

[17] Tianfan Fu, Wenhao Gao, Connor Coley, and Jimeng Sun. Reinforced genetic algorithm for structure-based drug design. *Advances in Neural Information Processing Systems*, 35:12325–12338, 2022.

[18] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in neural information processing systems*, 33:1970–1981, 2020.

[19] Wenhao Gao and Connor W Coley. The synthesizability of molecules proposed by generative models. *Journal of chemical information and modeling*, 60(12):5714–5723, 2020.

[20] Xiang Gao, Farhad Ramezanghorbani, Olexandr Isayev, Justin S. Smith, and Adrian E. Roitberg. Torchani: A free and open source pytorch-based deep learning implementation of the ani neural network potentials. *Journal of chemical information and modeling*, 60(7):3408–3415, 2020.

[21] Niklas WA Gebauer, Michael Gastegger, Stefaan SP Hessmann, Klaus-Robert Müller, and Kristof T Schütt. Inverse design of 3d molecular structures with conditional generative neural networks. *Nature communications*, 13(1):973, 2022.

[22] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

[23] Jiaqi Guan, Wesley Wei Qian, Xingang Peng, Yufeng Su, Jian Peng, and Jianzhu Ma. 3d equivariant diffusion for target-aware molecule generation and affinity prediction. In *International Conference on Learning Representations*, 2023.

[24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[25] Liegi Hu, Mark L Benson, Richard D Smith, Michael G Lerner, and Heather A Carlson. Binding moad (mother of all databases). *Proteins: Structure, Function, and Bioinformatics*, 60(3):333–340, 2005.

[26] James P Hughes, Stephen Rees, S Barrett Kalindjian, and Karen L Philpott. Principles of early drug discovery. *British journal of pharmacology*, 162(6):1239–1249, 2011.

[27] John J Irwin and Brian K Shoichet. Zinc- a free database of commercially available compounds for virtual screening. *Journal of Chemical Information and Modeling*, 45(1):177–182, 2005.

[28] JH Jensen. A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. chem sci 10 (12): 3567–3572, 2019.

[29] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Multi-objective molecule generation using interpretable substructures. In *International conference on machine learning*, pages 4849–4859. PMLR, 2020.

[30] Amit Kadan, Kevin Ryczko, Erika Lloyd, Adrian Roitberg, and Takeshi Yamazaki. Supplementary information for guided multi-objective generative ai for structure-based drug design. URL-to-be-inserted-upon-publication.

[31] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[32] David Ryan Koes, Matthew P Baumgartner, and Carlos J Camacho. Lessons learned in empirical scoring with smina from the csar 2011 benchmarking exercise. *Journal of chemical information and modeling*, 53(8):1893–1904, 2013.

[33] Ask Hjorth Larsen, Jens Jørgen Mortensen, Jakob Blomqvist, Ivano E Castelli, Rune Christensen, Marcin Dułak, Jesper Friis, Michael N Groves, Bjørk Hammer, Cory Hargus, et al. The atomic simulation environment—a python library for working with atoms. *Journal of Physics: Condensed Matter*, 29(27):273002, 2017.

[34] Seul Lee, Jaehyeong Jo, and Sung Ju Hwang. Exploring chemical space with score-based out-of-distribution generation. In *Proceedings of the 40th International Conference on Machine Learning*, pages 18872–18892. PMLR, 2023.

[35] Yibo Li, Jianfeng Pei, and Luhua Lai. Structure-based de novo drug design using 3d deep generative models. *Chemical science*, 12(41):13664–13675, 2021.

[36] Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E Gonzalez, and Ion Stoica. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018.

[37] Haitao Lin, Yufei Huang, Meng Liu, Xuanjing Li, Shuiwang Ji, and Stan Z Li. Diffbp: Generative diffusion of 3d molecules for target protein binding. *arXiv preprint arXiv:2211.11214*, 2022.

[38] Meng Liu, Youzhi Luo, Kanji Uchino, Koji Maruhashi, and Shuiwang Ji. Generating 3D molecules for target protein binding. In *Proceedings of the 39th International Conference on Machine Learning*, pages 13912–13924. PMLR, 2022.

[39] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[40] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022.

[41] Shitong Luo, Jiaqi Guan, Jianzhu Ma, and Jian Peng. A 3d generative model for structure-based drug design. *Advances in Neural Information Processing Systems*, 34:6229–6239, 2021.

[42] Andrew T McNutt, Paul Francoeur, Rishal Aggarwal, Tomohide Masuda, Rocco Meli, Matthew Ragoza, Jocelyn Sunseri, and David Ryan Koes. Gnina 1.0: molecular docking with deep learning. *Journal of cheminformatics*, 13(1):1–20, 2021.

[43] Michael M Mysinger, Michael Carchia, John J Irwin, and Brian K Shoichet. Directory of useful decoys, enhanced (dud-e): better ligands and decoys for better benchmarking. *Journal of medicinal chemistry*, 55(14):6582–6594, 2012.

[44] AkshatKumar Nigam, Pascal Friederich, Mario Krenn, and Alán Aspuru-Guzik. Augmenting genetic algorithms with deep neural networks for exploring the chemical space. *arXiv preprint arXiv:1909.11655*, 2019.

[45] Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9:1–14, 2017.

[46] Shyue Ping Ong, William Davidson Richards, Anubhav Jain, Geoffroy Hautier, Michael Kocher, Shreyas Cholia, Dan Gunter, Vincent L Chevrier, Kristin A Persson, and Gerbrand Ceder. Python materials genomics (pymatgen): A robust, open-source python library for materials analysis. *Computational Materials Science*, 68:314–319, 2013.

[47] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[48] Xingang Peng, Shitong Luo, Jiaqi Guan, Qi Xie, Jian Peng, and Jianzhu Ma. Pocket2mol: Efficient molecular sampling based on 3d protein pockets. In *Proceedings of the 39th International Conference on Machine Learning,*, pages 17644–17655. PMLR, 2022.

[49] Pavel G Polishchuk, Timur I Madzhidov, and Alexandre Varnek. Estimation of the size of drug-like chemical space based on gdb-17 data. *Journal of computer-aided molecular design*, 27:675–679, 2013.

[50] Rodrigo Quiroga and Marcos A Villarreal. Vinardo: A scoring function based on autodock vina improves scoring, docking, and virtual screening. *PloS one*, 11(5):e0155183, 2016.

[51] Matthew Ragoza, Tomohide Masuda, and David Ryan Koes. Generating 3d molecules conditional on receptor binding sites with deep generative models. *Chemical science*, 13(9):2701–2713, 2022.

[52] Lars Ruddigkeit, Ruud Van Deursen, Lorenz C Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of Chemical Information and Modeling*, 52(11):2864–2875, 2012.

[53] Kevin Ryczko, Pierre Darancet, and Isaac Tamblyn. Inverse design of a graphene-based quantum transducer via neuroevolution. *The Journal of Physical Chemistry C*, 124(48):26117–26123, 2020.

[54] Benjamin Sanchez-Lengeling, Carlos Outeiral, Gabriel L Guimaraes, and Alan Aspuru-Guzik. Optimizing distributions over molecular space. an objective-reinforced generative adversarial network for inverse-design chemistry (organic). *ChemRxiv*, 2017.

[55] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *Proceedings of the 38th International Conference on Machine Learning conference on machine learning*, pages 9323–9332. PMLR, 2021.

[56] Arne Schneuing. Diffsbdd, 2023.

[57] Arne Schneuing, Yuanqi Du, Charles Harris, Arian Jamasb, Ilia Igashov, Weitao Du, Tom Blundell, Pietro Lió, Carla Gomes, Max Welling, et al. Structure-based drug design with equivariant diffusion models. In *International Conference on Learning Representations*, 2023.

[58] Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *International Conference on Machine Learning*, pages 9377–9388. PMLR, 2021.

[59] AN Shivanyuk, SV Ryabukhin, A Tolmachev, AV Bogolyubsky, DM Mykytenko, AA Chupryna, W Heilman, and AN Kostyuk. Enamine real database: Making chemical diversity real. *Chemistry today*, 25(6):58–59, 2007.

[60] Grzegorz Skoraczyński, Mateusz Kitlas, Błażej Miasojedow, and Anna Gambin. Critical assessment of synthetic accessibility scores in computer-assisted synthesis planning. *Journal of Cheminformatics*, 15(1):6, 2023.

[61] Justin S Smith, Olexandr Isayev, and Adrian E Roitberg. Ani-1: an extensible neural network potential with dft accuracy at force field computational cost. *Chemical science*, 8(4):3192–3203, 2017.

[62] Justin S Smith, Ben Nebgen, Nicholas Lubbers, Olexandr Isayev, and Adrian E Roitberg. Less is more: Sampling chemical space with active learning. *The Journal of chemical physics*, 148(24), 2018.

[63] Justin S Smith, Benjamin T Nebgen, Roman Zubatyuk, Nicholas Lubbers, Christian Devereux, Kipton Barros, Sergei Tretiak, Olexandr Isayev, and Adrian E Roitberg. Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning. *Nature communications*, 10(1):2903, 2019.

[64] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[65] Jacob O Spiegel and Jacob D Durrant. Autogrow4: an open-source genetic algorithm for de novo drug design and lead optimization. *Journal of cheminformatics*, 12:1–16, 2020.

[66] Bhuvanesh Sridharan, Manan Goel, and U Deva Priyakumar. Modern machine learning for tackling inverse problems in chemistry: molecular design to realization. *Chemical Communications*, 58(35):5316–5331, 2022.

[67] Amol Thakkar, Veronika Chadimová, Esben Jannik Bjerrum, Ola Engkvist, and Jean-Louis Reymond. Retrosynthetic accessibility score (rascore)–rapid machine learned synthesizability classification from ai driven retrosynthetic planning. *Chemical Science*, 12(9):3339–3349, 2021.

[68] Richard Tran, Janice Lan, Muhammed Shuaibi, Brandon M Wood, Siddharth Goyal, Abhishek Das, Javier Heras-Domingo, Adeesh Kolluru, Ammar Rizvi, Nima Shoghi, et al. The open catalyst 2022 (oc22) dataset and challenges for oxide electrocatalysts. *ACS Catalysis*, 13(5):3066–3084, 2023.

[69] Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

[70] Renxiao Wang, Xueliang Fang, Yipin Lu, Chao-Yie Yang, and Shaomeng Wang. The pdbbind database: methodologies and updates. *Journal of medicinal chemistry*, 48(12):4111–4119, 2005.

[71] Zechen Wang, Liangzhen Zheng, Sheng Wang, Mingzhi Lin, Zhihao Wang, Adams Wai-Kin Kong, Yuguang Mu, Yanjie Wei, and Weifeng Li. A fully differentiable ligand pose optimization framework guided by deep learning and a traditional scoring function. *Briefings in Bioinformatics*, 24(1):bbac520, 2023.

[72] Tomer Weiss, Eduardo Mayo Yanes, Sabyasachi Chakraborty, Luca Cosmo, Alex M Bronstein, and Renana Gershoni-Poranne. Guided diffusion for inverse molecular design. *Nature Computational Science*, 3(10):873–882, 2023.

[73] Yutong Xie, Chence Shi, Hao Zhou, Yuwei Yang, Weinan Zhang, Yong Yu, and Lei Li. Mars: Markov molecular sampling for multi-objective drug discovery. *arXiv preprint arXiv:2103.10432*, 2021.

[74] Guozhang Xu, Marta C Abad, Peter J Connolly, Michael P Neeper, Geoffrey T Struble, Barry A Springer, Stuart L Emanuel, Niranjan Pandey, Robert H Gruninger, Mary Adams, et al. 4-amino-6-arylamino-pyrimidine-5-carbaldehyde hydrazones as potent erbb-2/egfr dual kinase inhibitors. *Bioorganic & medicinal chemistry letters*, 18(16):4615–4619, 2008.

[75] Jiahui Yu, Jike Wang, Hong Zhao, Junbo Gao, Yu Kang, Dongsheng Cao, Zhe Wang, and Tingjun Hou. Organic compound synthetic accessibility prediction based on the graph attention mechanism. *Journal of chemical information and modeling*, 62(12):2973–2986, 2022.

[76] Yuejiang Yu, Shuqi Lu, Zhifeng Gao, Hang Zheng, and Guolin Ke. Do deep learning models really outperform traditional approaches in molecular docking? *arXiv preprint arXiv:2302.07134*, 2023.

[77] Shehtab Zaman, Denis Akhiyarov, Mauricio Araya-Polo, and Kenneth Chiu. Stride: Structure-guided generation for inverse design of molecules. *arXiv preprint arXiv:2311.06297*, 2023.

[78] Barbara Zdrazil, Eloy Felix, Fiona Hunter, Emma J Manners, James Blackshaw, Sybilla Corbett, Marleen de Veij, Harris Ioannidis, David Mendez Lopez, Juan F Mosquera, et al. The chembl database in 2023: a drug discovery platform spanning multiple bioactivity data types and time periods. *Nucleic acids research*, 52(D1):D1180–D1192, 2024.

[79] CH Zhang, EA Stone, M Deshmukh, JA Ippolito, MM Ghahremanpour, J Tirado-Rives, KA Spasov, S Zhang, Y Takeo, SN Kudalkar, et al. Potent noncovalent inhibitors of the main protease of sars-cov-2 from molecular sculpting of the drug perampanel guided by free energy perturbation calculations. acs cent. sci. 2021, 7 (3), 467–475. *DOI: https://doi. org/10.1021/acscentsci. 1c00039. PMID: https://www. ncbi. nlm. nih. gov/pubmed/33786375*, pages 467–475.

[80] Chun-Hui Zhang, Elizabeth A Stone, Maya Deshmukh, Joseph A Ippolito, Mohammad M Ghahremanpour, Julian Tirado-Rives, Krasimir A Spasov, Shuo Zhang, Yuka Takeo, Shalley N Kudalkar, et al. Potent noncovalent inhibitors of the main protease of sars-cov-2 from molecular sculpting of the drug perampanel guided by free energy perturbation calculations. *ACS central science*, 7(3):467–475, 2021.

[81] Zhenpeng Zhou, Steven Kearnes, Li Li, Richard N Zare, and Patrick Riley. Optimization of molecules via deep reinforcement learning. *Scientific reports*, 9(1):10752, 2019.

[82] Alex Zunger. Inverse design in search of materials with target functionalities. *Nature Reviews Chemistry*, 2(4):0121, 2018.

Supplementary Information

S.1 Methods

S.1.1 Generator Module

When optimizing latent vectors, we utilize a state-of-the-art denoising diffusion probabilistic model (DDPM) [24], DiffSBDD [57], for generating novel ligands with high binding affinity. DDPMs generate samples from a target distribution by learning the reverse of a noising process. Gaussian random noise is iteratively injected into samples from the target distribution until no information from the original sample remains. During generation the model reverses this process, transforming random noise into samples from the target distribution. In particular, a diffusion model generates samples by denoising a random initial latent vector for T steps. The initial latent vector is drawn from a normal distribution,

$$\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Afterwards, the model generates consecutive latent vectors by predicting the noise at time t, $\epsilon_\theta(\mathbf{z}_t, t)$, where θ are the model weights. The noise is removed from $\mathbf{z}_t$ in order to generate $\mathbf{z}_{t-1}$. $\mathbf{z}_0$ is the final prediction of the model.

DiffSBDD is an *SE(3)*-equivariant [18] 3D-conditional DDPM which respects translation, rotation, and permutation symmetries. DiffSBDD was trained to create ligands with high binding affinity given a target protein pocket. In DiffSBDD, data samples consist of protein pocket and ligand point clouds (atomic numbers and coordinates), i.e., $\mathbf{z} = [\mathbf{r}, \mathbf{h}]$ where $\mathbf{r} \in \mathbb{R}^{N\times 3}$ is a tensor of atomic coordinates, and $\mathbf{h} \in \mathbb{R}^{N\times 10}$ is a tensor of atomic probabilities over the atom types which the model can generate. Within the model, each $\mathbf{z}_t$ is converted to a graph, and processed by an EGNN [55] to produce a prediction of $\epsilon_\theta(\mathbf{z}_t, t)$. DiffSBDD contains two different models for 3D pocket conditioning – a conditional DDPM that receives a fixed pocket representation as the context in each denoising step, and a model that is trained to approximate the joint distribution of ligand-protein pocket pairs and is combined with a modified sampling procedure, inpainting [64, 40], at inference time to yield conditional samples of ligands given a fixed protein pocket. In this work, we focus only on the conditional DDPM for ligand generation. Our framework requires that generated ligands do not overlap with the target protein pocket, as the ligands are later docked using structural optimization. We found that the conditional DDPM model is the only model consistently capable of generating ligands satisfying this constraint.

DiffSBDD was trained on a subset of the CrossDocked [16], and the Binding MOAD [25] datasets. For training, Ref. [57] used the same train/test splits as in Ref. [41] and Ref. [48], resulting in 100,000 complexes for training, and 100 protein pockets for testing. Ref. [57] filtered the database to contain only molecules with atom types compatible with their model, and removed corrupted entries, resulting in 40,344 complexes for training, and 130 protein pockets for testing.

DiffSBDD was shown to achieve state-of-the-art performance on both test sets. In particular, DiffSBDD achieved the best average, and best top-10% Vina score when compared with other state-of-the-art models in the literature – 3D-SBDD [41], Pocket2Mol[48], GraphBP [38], and TargetDiff [23]. We note, that although DiffSBDD is used as our baseline model in this report, our framework is not limited to the use of this specific model – any other generative model which makes use of latent vectors as intermediate representations during generation can take its place.

S.1.2 Ligand Validity Checks

When generating ligands using DiffSBDD, we perform several chemical and structural checks to ensure that the generated ligand is valid. A number of these checks are done using RDKit [1]. These include verifying that hydrogens can be added to the ligand and assigned a Cartesian coordinate (using the *addCoords* option in *Chem.AddHs*), that the ligand is not fragmented, and that the ligand can be sanitized. All of these except for the valency check can also be done within DiffSBDD [56].

In addition, we have four more checks to ensure the structural validity of the ligand. These four checks are necessary to be able to run structural refinement with IDOLpro, which makes use of the ANI2x model [12]. Structural refinement is described in the Structural Refinement Section S.1.5. We first make sure that the ligand contains only atoms compatible with ANI2x. DiffSBDD can generate ligands with four atom types that are incompatible with ANI2x – B, P, Br, and I. We also make sure

that the bond lengths in the ligand are correct by referring to covalent radii, and that the ligand does not overlap with the protein pocket. This is done via ASE's [33] (Atomic Simulation Environment) *NeighborList* class. Lastly, We make sure that the atoms do not have significant overlap within the ligand itself. This is done via pymatgen's [46] *Molecule* class.

S.1.3 Scoring Module

After generating a set of ligands, we pass them to a scoring module. In this work, we include a custom torch-based Vina score [69] which we refer to as torchvina, an equivariant neural network trained to predict the synthetic accessibility of molecules with 3D information [14] which we refer to as torchSA, the scoring module from DiffDock [11], and the ANI2x model [12]. These objectives are all written using Pytorch [47] with differentiable operations and hence can be differentiated automatically using autograd.

torchvina We re-implement the Vina force field [69] using Pytorch to allow for automatic differentiation with respect to the latent parameters of the generator. Our work is not the first to produce a Pytorch-based version of Vina to facilitate automatic differentiation, a similar implementation was presented by Ref. [71]. Our motivation for implementing a differentiable Vina score is that docking with Vina was shown to outperform state-of-the-art ML models such as DiffDock [11] when stricter chemical and physical validity checks were enforced on docked molecules, or when these procedures were evaluated on a dataset composed of examples distinct from the ML models' training data [8].

The Vina force field is composed of a weighted sum of atomic interactions. Steric, hydrophobic, and hydrogen bonding interactions are calculated and weighted according to a nonlinear fit to structural data [69]. The final score is re-weighted by the number of rotatable bonds to account for entropic penalties [42]. The Vina score is composed of a sum of intramolecular and intermolecular terms, both of which are integrated into our implementation. Although not used in the study, we have added the ability to score molecules with the Vinardo score [50], a re-weighted version of the Vina score which was shown to outperform the Vina scores for docking and virtual screening on a number of tests.

torchSA To have an evaluator model capable of estimating synthesizability, we train an equivariant neural network to predict the synthetic accessibility (SA) score. SA score was first proposed by Ref. [14], ranges from 1 (easy to make) and 10 (very difficult to make), and shown to be effective for biasing generative pipelines towards synthesizable molecules [19, 60]. Moreover, it was used directly in DiffSBDD to measure the performance of the pipeline [57]. To be able to guide latent parameters in DiffSBDD towards generating ligands with high synthesizability required designing a model that can handle the outputs of DiffSBDD in a differentiable manner. In particular, we constructed a machine learning model that can take in atomic point clouds, $\mathbf{z} = [\mathbf{r}, \mathbf{h}]$. We accomplish this by constructing a dataset of atomic point clouds of ligands labelled with SA score. To allow for predictions on probability distributions of atom types, we encode atom types as one-hot vectors. For more details, we refer the reader to Section S.3.

ANI2x ANI2x is a neural network ensemble model that is part of the ANI suite of models [20]. The ANI models are trained on quantum chemistry calculations (at the density functional theory level) and they predict the total energy of a target system. The ANI models are trained on millions of organic molecules and are accurate across different domains [61, 62, 12, 63]. In addition, they have been shown to outperform many common force fields in terms of accuracy [15]. The ANI models make use of atomic environment descriptors, which probe their local environment, as input vectors. An individual ANI model contains multiple neural networks, each specialized for a specific atom type, predicting the energy contributed by atoms of that type in the molecular system. The total energy of the system is obtained by performing a summation over the atomic contributions [61]. The ANI2x model is an ensemble model consisting of 8 individual ANI models. Each sub-model is trained on a different fold of the ANI2x dataset, composed of gas-phase molecules containing seven different atom types – H, C, N, O, F, Cl, and S [12]. These seven atom types cover $\approx 90\%$ of drug-like molecules, making ANI2x a suitable ML model for usage in our framework.

S.1.4 Latent Vector Optimization

The main optimization in IDOLpro occurs via the modification of latent vectors used by the generator to generate novel ligands. We do this by repeatedly evaluating generated ligands with an objective composed of a set of differentiable scores, calculating the gradient of the objective with respect to the latent vectors (facilitated by automatic differentiation with Pytorch [47]), and modifying the latent vectors via a gradient-based optimizer.

When optimizing latent vectors in DiffSBDD, we do not modify the initial latent vectors used by the model. Instead, we define an optimization horizon, t_{hz}. First latent vectors are generated up to the optimization horizon $\mathbf{z}_T, \ldots, \mathbf{z}_{t_{\text{hz}}}$. This latent vector is saved, and the remaining latent vectors, $\mathbf{z}_{t_{\text{hz}}-1}, \ldots, \mathbf{z}_0$, are generated. The gradient of the objective with respect to $\mathbf{z}_{t_{\text{hz}}}$ is evaluated, and $\mathbf{z}_{t_{\text{hz}}}$ is modified using the Adam optimizer [31]. When re-generating ligands, rather than starting from $\mathbf{z}_T$, only latent vectors proceeding the optimization horizon are generated, i.e., $\mathbf{z}_{t_{\text{hz}}-1}, \ldots, \mathbf{z}_0$.

In this work, we focus on using two combinations of evaluators: torchvina on its own, and torchvina in combination with torchSA. We use the Adam optimizer with a learning rate of 0.1, $\beta_1 = 0.5$ and $\beta_2 = 0.999$ to modify latent vectors. We perform hyperparameter optimization to choose the optimization horizon, described in Section S.4.2.

S.1.5 Structural Refinement

Structural refinement, in the form of local coordinate optimization, proceeds similarly to latent vector optimization. The scoring module is used to repeatedly evaluate ligands, and the derivatives concerning the ligand's coordinates are used to modify the ligand's coordinates with a gradient-based optimizer. We use the L-BFGS optimizer in Pytorch [47] to perform coordinate optimization. Our optimization algorithm is implemented with Pytorch and is parallelizable on a GPU. In this work, we only use one combination of evaluators to perform coordinate optimization: torchvina and ANI2x. We discuss the selection of different inter and intra-molecular forces in Section S.4.3.

S.2 Additional experiments

S.2.1 Validating latent vector optimization

To assess the ability of our platform to augment the performance of the baseline model via the optimization of latent vectors, we run IDOLpro to optimize torchvina and torchSA, and analyze its capability to improve the Vina and SA scores relative to DiffSBDD-Cond. For each of the protein pockets in the CrossDocked and Binding MOAD test sets, we generate 100 optimized ligands using IDOLpro. We calculate the Vina and SA scores of ligands before and after latent vector optimization with IDOLpro. We report the average Vina and SA scores, and the top-10% Vina and SA scores for each method on each test set. We also report the average percentage of synthesizable molecules generated. In this work, we define a ligand as synthesizable if it achieves an SA score of less than 3.5. Although the inventors of the SA score suggest 6 as the cutoff for synthesizability, a number of papers have found SA scores between 3.5 and 6 to be ambiguous [19, 67]. A cutoff of 3.5 was also used to determine synthesizability in Ref. [75].

The results are shown in Table S1. We find that IDOLpro generates ligands with better Vina and SA scores than DiffSBDD-cond, yielding molecules with $\approx 20\%$ better Vina scores and $\approx 21\%$ better SA scores on the CrossDocked dataset, and $\approx 26\%$ better Vina scores and $\approx 21\%$ better SA scores on the Binding MOAD dataset. Furthermore, IDOLpro yields more than double the amount of synthesizable ligands compared with DiffSBDD-cond for each dataset (51.2 % vs 23.5 % and 56.9 % vs 22.6 %). Overall, IDOLpro is able to co-optimize Vina and SA, producing molecules with significantly better binding affinities and synthetic accessibility when compared to the baseline model, DiffSBDD-cond.

S.2.2 Comparisons to DL and non-DL methods

Lastly, we compare IDOLpro to various non-deep learning-based methods in the literature. These methods include genetic algorithms [44, 28, 65, 17], reinforcement learning [81, 2, 45, 29], and an MCMC method [73]. We evaluate these methods across the 10 protein pockets in the RGA test set. For each target, as was done in Ref. [17], we generate 1000 ligands with IDOLpro and calculate the average top-100, top-10, and top-1 Vina score. We also record the average SA and QED of molecules,

Dataset	Method	Vina [kcal/mol]	SA	Synth [%]
CrossDocked	DiffSBDD-cond [57]	-5.37 ± 1.93	4.30 ± 0.50	23.5 ± 15.5
	IDOLpro	-6.47 ± 2.10	3.41 ± 0.70	51.2 ± 22.7
MOAD	DiffSBDD-cond [57]	-5.38 ± 2.55	4.18 ± 0.50	26.5 ± 17.8
	IDOLpro	-6.77 ± 2.24	3.32 ± 0.66	56.9 ± 22.6

Table S1: Performance of IDOLpro when used to optimize torchvina and torchSA relative to the baseline model, DiffSBDD-cond on the CrossDocked and Binding MOAD test sets. The average Vina score, SA score, and the average percent of synthesizable molecules are reported.

Method	$\text{Vina}_{\text{top-100}}$ [kcal/mol]	$\text{Vina}_{\text{top-10}}$ [kcal/mol]	$\text{Vina}_{\text{top-1}}$ [kcal/mol]	SA
MARS [73]	-7.76 ± 0.61	-8.8 ± 0.71	-9.26 ± 0.79	2.69 ± 0.08
MolDQN [81]	-6.29 ± 0.40	-7.04 ± 0.49	-7.50 ± 0.40	5.83 ± 0.18
GEGL [2]	-9.06 ± 0.92	-9.91 ± 0.99	-10.45 ± 1.04	2.99 ± 0.05
REINVENT [45]	-10.81 ± 0.44	-11.23 ± 0.63	-12.01 ± 0.83	2.60 ± 0.12
RationaleRL [29]	-9.23 ± 0.92	-10.83 ± 0.86	-11.64 ± 1.10	2.92 ± 0.13
GA+D [44]	-8.69 ± 0.45	-9.29 ± 0.58	-9.83 ± 0.32	3.45 ± 0.12
Graph-GA [28]	-10.48 ± 0.86	-11.70 ± 0.93	-12.30 ± 1.91	3.50 ± 0.37
Autogrow 4.0 [65]	-11.37 ± 0.40	-12.21 ± 0.62	-12.47 ± 0.84	2.50 ± 0.05
RGA [17]	-11.87 ± 0.17	-12.56 ± 0.29	-12.89 ± 0.47	$\mathbf{2.47 \pm 0.05}$
IDOLpro	$\mathbf{-14.59 \pm 1.51}$	$\mathbf{-16.26 \pm 1.66}$	$\mathbf{-17.35 \pm 2.10}$	3.77 ± 0.33

Table S2: Comparison of IDOLpro to various score and sample-based methods on 10 disease-related protein targets. The average top-100, top-10, and top-1 Vina scores across the targets are reported, along with the average SA. The top-performing model on each metric is bolded in the corresponding column. Numbers for other methods are taken from Ref. [17].

Method	QED	Diversity
MARS [73]	0.71 ± 0.01	$\mathbf{0.88 \pm 0.00}$
MolDQN [81]	0.17 ± 0.02	$\mathbf{0.88 \pm 0.01}$
GEGL [2]	0.64 ± 0.01	0.85 ± 0.00
REINVENT [45]	0.45 ± 0.06	0.86 ± 0.01
RationaleRL [29]	0.32 ± 0.02	0.72 ± 0.03
GA+D [44]	0.70 ± 0.02	0.87 ± 0.01
Graph-GA [28]	0.46 ± 0.07	0.81 ± 0.04
Autogrow 4.0 [65]	$\mathbf{0.75 \pm 0.02}$	0.85 ± 0.01
RGA [17]	0.74 ± 0.04	0.86 ± 0.02
IDOLpro	0.64 ± 0.06	0.72 ± 0.04

Table S3: Comparison of IDOLpro to various score and sample-based methods on 10 disease-related protein targets for QED and diversity. The top-performing model on each metric is bolded in the corresponding column. Numbers for other methods are taken from Ref. [17].

along with the average diversity per protein pocket. Numbers for other methods are taken from Ref [17]. Results are shown in Table S2 and Table S3.

IDOLpro greatly outperforms non-DL techniques in terms of Vina score, improving on the next best method by $\approx 23\%$, $\approx 29\%$, and $\approx 35\%$ in terms of average top-100, top-10, and top-1 Vina score respectively. Unlike when compared to other DL methods, IDOLpro ranks behind most of these methods in terms of average SA, ranking 9^{th} out of the 10 methods compared. IDOLpro is middle-of-the-pack in terms of QED, ranking 5th out of the 10 methods compared. This shows that IDOLpro, and deep learning methods in general, have a ways to go before they can produce molecules with the same synthesizability and drug-likeness as other advanced non-DL methods in the literature. This is an ongoing area of research [19, 8], and is an aspect that we would like to improve in IDOLpro.

S.3 More details of the SA model

To train the SA model, we prepare a dataset consisting of all molecules with structural information in ChemBL [78] (2,409,270 structures), and ligands used to train DiffSBDD [57] on CrossDocked2020

	Method	Diversity	Time [s/ligand]
CrossDocked	3D-SBDD [41]	0.74 ± 0.09	328.13 ± 245.43
	Pocket2Mol [48]	0.74 ± 0.15	41.79 ± 36.84
	GraphBP [38]	$\mathbf{0.84 \pm 0.01}$	$\mathbf{0.17 \pm 0.02}$
	TargetDiff [23]	0.72 ± 0.09	~ 57.22
	DiffSBDD-cond [57]	0.73 ± 0.07	2.27 ± 0.86
	DiffSBDD-inpaint [57]	0.76 ± 0.05	2.67 ± 1.22
	IDOLpro	0.79 ± 0.07	58.80 ± 32.97
MOAD	GraphBP [38]	$\mathbf{0.83 \pm 0.01}$	$\mathbf{0.23 \pm 0.03}$
	DiffSBDD-cond [57]	0.71 ± 0.08	5.61 ± 1.42
	DiffSBDD-inpaint [57]	0.74 ± 0.05	6.17 ± 2.08
	IDOLpro	0.77 ± 0.07	82.30 ± 45.07

Table S4: Evaluation of DL tools on targets from the CrossDocked and Binding MOAD datasets for diversity and time. The average, along with the standard deviation of each metric across the protein pockets in each dataset is reported. The top performing model on each metric is bolded in the corresponding column. Numbers for other models are taken from Ref. [57]. Time is based on running DiffSBDD-cond on our hardware, and adjusting the times reported in Ref. [57] accordingly.

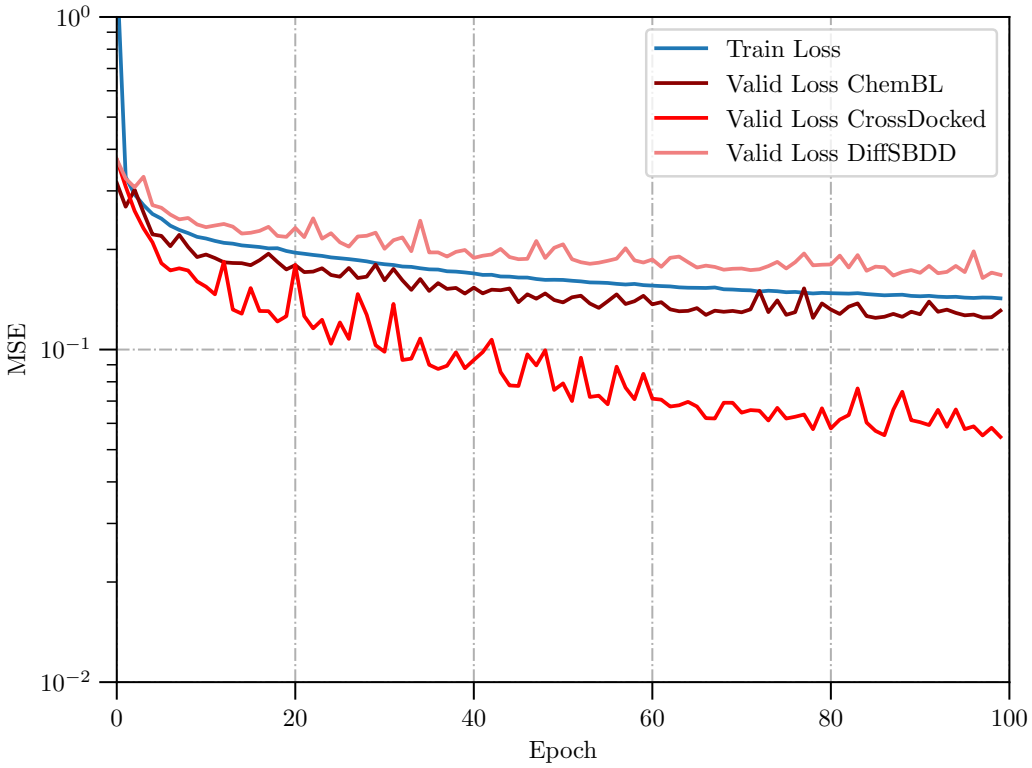

Figure S1: Training and validation curves for training PaiNN to predict the SA score on 3D atomic point clouds. The MSE at each epoch is plotted for the training set, the ChemBL validation set, the DiffSBDD validation set, and the CrossDocked validation set.

[16] (183,468 structures). Although the SA score is fully determined by the chemical graph of a molecule, we keep molecules with different conformations from CrossDocked2020 to aid the model in learning the redundancy of pose in determining the SA score. To improve the model's performance on ligands produced by DiffSBDD, we generate nearly 1,000,000 (877,284) ligands with DiffSBDD which are included in the training data. We generate several ligands for each of the protein pockets in the DiffSBDD training set and then filter them using the same validity checks described in the Methods section. We put a higher emphasis on modelling ligands from CrossDocked2020 and

DiffSBDD, sampling from one of these datasets during training with a $5\times$ higher likelihood than ChEMBL.

We train the polarizable atomic interaction neural network [58] (PaiNN) from the Open Catalyst Project [9, 68] to predict the SA score given the atomic coordinates and atom types. To allow PaiNN to make predictions on atom types coming out of DiffSBDD, we encode atom types as one-hot vectors. We first optimize the hyperparameters of the model using Ray Tune [36]. The hyperparameters chosen were *num_rbf* = 64, *num_layers* = 4, *max_neighbor* = 30, *cutoff* = 8.0, *hidden_channels* = 256. We use a 95%/5% training/validation split for each dataset. The model is trained for 100 epochs to minimize the MSE loss with the AdamW optimizer [39] with a learning rate of 5×10^{-4}. Training and validation curves are plotted in Fig. S1.

S.4 Hyperparameter Tuning

PDB ID	Ligand ID
2ah9	cto
5lvq	p2l
5g3n	u8d
1u0f	g6p
4bnw	fxe
4i91	cpz
2ati	ihu
2hw1	lj9
1bvr	geq
1zyu	k2q

Table S5: Validation set used to choose hyper-parameters in IDOLpro. All proteins from the test set of LiGAN [51] were used, and a single protein pocket for each protein was selected at random.

We perform several hyperparameter tuning experiments to fine-tune the performance of our pipeline. To avoid overfitting on our two benchmark sets, we perform all experiments using a non-overlapping validation set. This validation set is composed of 10 targets taken from the test set of LiGAN [51], which was used to validate the performance of several other works in the literature [38, 37]. For each of these ten targets, a single pocket is randomly selected to be included in the validation set. The set of receptor-ligand combinations used in these experiments is included in Table S5. We generate 20 ligands per pocket for each experiment. Each experiment is run on an NVIDIA A10G GPU with 24 GB of GPU memory.

S.4.1 Accelerating Diffusion

diffusion steps	Vina [kcal / mol]	SA	QED	Time [s]
5	-6.25 ± 2.08	3.72 ± 0.46	0.53 ± 0.10	60.60 ± 49.75
50	-6.72 ± 2.45	3.92 ± 0.58	0.54 ± 0.10	196.96 ± 102.97

Table S6: Results when reducing the number of rollout steps from 50 to 5. The average Vina, SA, and QED across the validation set is reported.

In DiffSBDD, the models are trained to generate ligands over 500 reverse diffusion steps according to some noise schedule. At each time step an equivariant network takes in the noised coordinates and atom types, as well as the time step, and returns a denoised representation of the atoms and coordinates [57]. One can reduce the number of reverse diffusion steps by skipping time steps in the noise schedule.

In IDOLpro, the majority of the reverse diffusion process is run to generate $\mathbf{z}_{t_{hz}}$, i.e., to seed the initial latent vectors. When optimizing $\mathbf{z}_{t_{hz}}$, the rollout of $\mathbf{z}_{t_{hz}}, \ldots, \mathbf{z}_0$ needs to be repeated many times. We run an experiment to determine whether we can reduce the number of steps during this final rollout 10-fold without a significant degradation in performance. We run an experiment with the smallest horizon considered $t_{hz} = 50$. Results are shown in table Table S6. Running the rollout with reduced diffusion steps results in over a $3\times$ speedup, with a slight decrease in average Vina score (<

0.5 kcal /mol), while preserving average SA and QED. We adopt this setting in our pipeline for this reason.

S.4.2 Tuning Optimization Horizon

We tune the value of the optimization horizon, t_{hz}, to optimize both the Vina and SA scores of generated molecules. We consider $t_{hz} \in 50, 100, 200$. For each setting of the optimization horizon, we track the difference in Vina score, SA score, and QED. Results are reported in Table S7. Based on these results we set the optimization horizon to 200, since that setting resulted in by far the best difference in Vina score and QED, albeit a slightly worse improvement in SA score relative to setting $t_{hz} = 100$.

t_{hz}	Δ Vina	Δ SA	Δ QED
50	-1.21	-0.34	-0.028
100	-1.17	-0.82	0.004
200	-1.84	-0.76	0.048

Table S7: Results when varying the optimization horizon t_{hz} in IDOLpro. The difference in Vina, SA, and QED for the final optimized ligands produced by IDOLpro relative to the initial ligands produced by DiffSBDD are reported.

S.4.3 Structural Refinement with Torchvina and ANI2x

Here we discuss how we tune the performance of our structural refinement procedure. We consider various combinations of intra-molecular and inter-molecular forces derived from the torchvina and ANI2x [12] potentials. We use the following parameters in the L-BFGS optimization algorithm: *max_iter*=100, *tolerance_grad*=10^{-3}, and *line_search_fn*="strong_wolfe". For each experiment, we keep track of the average Vina score, the top%10 Vina score, the average time taken to optimize each ligand, and the percent of valid structures that are output by the procedure based on our validity checks. We tabulate these results and include the same metrics when QuickVina [3] is used for docking for comparison. QuickVina was used to dock structures after generation in DiffSBDD [57].

When analyzing the reason for invalid molecules, we find that a large number of failed cases are due to atoms becoming too far during the structural relaxation, causing the molecule to become disconnected. To remedy this, we add an $L1$ penalty for violating the bonds in the molecule produced by IDOLpro. To do so, we use ASE's [33] natural cutoffs. We find that setting a weight of 0.01 on this $L1$ penalty results in the best balance of Vina score and validity.

Method	Torchvina	ANI2x	$L1$ bond penalty	Vina [kcal/mol]	$\text{Vina}_{10\%}$ [kcal/mol]
QVina	-	-	-	-8.51	-9.51
IDOLpro	Inter+Intra	Inter+Intra	✗	-9.26	-10.35
IDOLpro	Inter+Intra	Intra	✗	-9.33	-10.41
IDOLpro	Inter	Inter+Intra	✗	-9.38	-10.53
IDOLpro	Inter	Intra	✗	-9.53	-10.70
IDOLpro	Inter	Intra	✓	-9.39	-10.68

Table S8: Results when running structural refinement various combinations of inter-molecular and intra-molecular forces derived from the Vina and ANI2x potentials. We also make note of whether an $L1$ penalty for encforcing bonds was used. For each experiment, we note the Vina score, top-10 Vina score.

S.4.4 Stopping Criteria, Backtracking, and Decaying Learning Rate

We use per-parameter options in Pytorch [47] to allow for individualized learning rates for different ligands. For each ligand, we optimize it with Adam with the chosen hyperparameters. We optimize each latent vector for 10-200 optimization steps. Often, during latent vector optimization, a ligand will be pushed to a part of latent space such that it becomes invalid. In such a case, we attempt to generate a ligand 10 times with the given latent vector. If after 10 attempts, reverse diffusion has not produced a valid ligand, we backtrack to the previous latent vector in the optimization trajectory, reduce the learning rate by a factor of 10, and restart the optimization. If at another point in the

Method	Torchvina	ANI2x	$L1$ bond penalty	Validity [%]	Time [s]
QVina	-	-	-	95.0	75.0
IDOLpro	Inter+Intra	Inter+Intra	✗	80.1	19.34
IDOLpro	Inter+Intra	Intra	✗	77.2	22.49
IDOLpro	Inter	Inter+Intra	✗	82.1	18.53
IDOLpro	Inter	Intra	✗	81.6	23.90
IDOLpro	Inter	Intra	✓	86.6	24.45

Table S9: Results when running structural refinement various combinations of inter-molecular and intra-molecular forces derived from the Vina and ANI2x potentials. We make note of whether an $L1$ penalty for encforcing bonds was used, the percent of valid structures produced with the combination of potentials. Validity checks are performed according to the checks described in Section Validity Checks S.1.2.

optimization, with the reduced learning rate, another latent vector fails to generate a valid ligand over 10 attempts, the optimization of that trajectory is stopped.

S.5 An Additional Scoring Function: DiffDock

We include the scoring module from DiffDock [11] in our evaluator module. DiffDock is composed of two modules, a docking module and a scoring module, which together can dock ligands to a target protein without pocket information. The DiffDock docking module was trained to predict the experimental binding pose of ligands in the PDBBind dataset [70]. The DiffDock scoring module was trained on experimental data where the goal of the model was to classify whether or not a candidate ligand is < 2 Å of the experimental binding pose. DiffDock docks ligands by producing many binding poses for a target ligand with the docking module, and returning these poses as a ranked list using the scoring module. The node from the final classification layer of the scoring network, indicating the likelihood that a docked ligand is < 2 Å from an experimentally derived binding pose, can be used as a scoring function.

DiffDock was shown to have state-of-the-art performance on a blind docking task for protein-ligand pairs extracted from the PDBBind dataset, significantly outperforming other state-of-the-art ML-based docking procedures [11].

S.6 Visualization of Latent Vectors

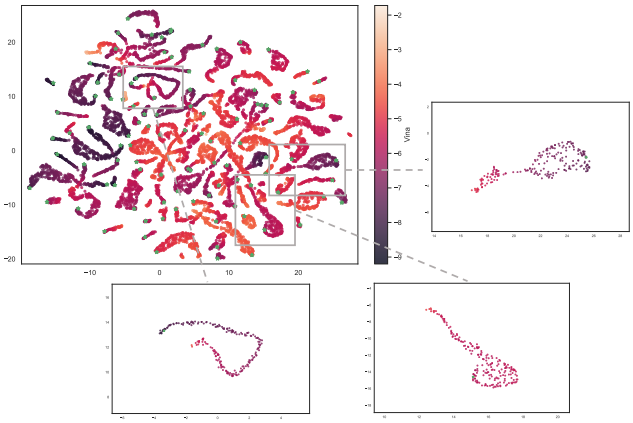

Figure S2: Latent vector visualizations of IDOLpro when generating ligands for 14gs. The points are coloured by Vina score (darker implies lower scores), and a green star marks the end of the optimization trajectory.