Target-Aware Variational Auto-Encoders for Ligand Generation with Multi-Modal Protein Modeling

Nhat Khang Ngo *
FPT Software AI Center
Hanoi, Vietnam
khangnn3@fsoft.com.vn

Truong Son Hy * †
Indiana State University
Terre Haute, IN 47809, USA
TruongSon.Hy@indstate.edu

Abstract

Without knowledge of specific pockets, generating ligands based on the global structure of a protein target plays a crucial role in drug discovery as it helps reduce the search space for potential drug-like candidates in the pipeline. However, contemporary methods require optimizing tailored networks for each protein, which is arduous and costly. To address this issue, we introduce TargetVAE, a target-aware variational auto-encoder that generates ligands with high binding affinities to arbitrary protein targets, guided by a novel prior network that learns from entire protein structures. We showcase the superiority of our approach by conducting extensive experiments and evaluations, including the assessment of generative model quality, ligand generation for unseen targets, docking score computation, and binding affinity prediction. Empirical results demonstrate the promising performance of our proposed approach. Our source code in PyTorch is publicly available at `https://github.com/HySonLab/Ligand_Generation`.

1 Introduction

Drug discovery is a complex and expensive process that involves multiple stages and often takes years of development, with costs running into billions of dollars [15]. The first stage is to design novel drug-like compounds that have high binding affinities to protein targets. This process consists of two sub-tasks: searching for candidates and measuring drug-target affinities (DTA). Searching for potential candidates in a huge database of roughly 10^{33} chemically valid molecules is a daunting task as current methods often rely on virtual screenings, professional software, and expert evaluation [40, 3]. Besides, drug-target affinities (DTA) are critical measurements for identifying potential candidates, as well as avoiding those that are inefficient for clinical trials. The most reliable technique for predicting DTA involves atomistic molecular dynamics simulations. However, these methods are computationally expensive and time-consuming, making them infeasible for large-scale sets of protein-ligand complexes. Our ultimate objective is to accelerate and automate these two sub-tasks in the first stage of the drug development process, using computational methods and machine-learning techniques.

To effectively design probable drug-like candidates, deep generative models [46, 17, 18, 25, 36, 18, 7, 17, 26, 25] have been proposed as a potential approach to reduce the amount of work for wet-lab experiments [11, 3, 40]. These methods demonstrate remarkable results in the unconditional generation or optimization for simple molecular properties (e.g., QED, SA, etc.). However, when enhancing binding affinity or other computationally expensive molecular properties, these generative models are prohibitively slow. They need to be trained in reinforcement learning frameworks where the generated molecular graph is modified based on the reward. It is worth noting that this reward

*Co-first authors
†Correspondent author

NeurIPS 2023 Generative AI and Biology (GenBio) Workshop.

function is determined by calling a property network that estimates the binding affinities. Albeit effective and powerful, these approaches require that specific property networks are trained for each protein target, which is not trivial due to the vast amount of (un)-known proteins [1, 3]. Furthermore, binding scores (labels) for supervision training are not widely available, and computing them via software like Autodock or Vina is time-consuming.

Contributions In summary, our contributions are three-fold as follows:

- We build a conditional VAE model that can generate chemically valid, drug-like molecules with high binding scores to an arbitrarily given protein structure. Apart from other methods, ours can directly condition the entire structure of any protein target and design multiple candidates that can bind to it, without requiring the training of a specific property network for each target.
- To diversify the generated results, we adapt previous works in computer vision domains. Specifically, we aim at transferring weights of a pre-trained unconditional VAE, which is trained on a large dataset of drug-like molecules, to a conditional VAE which is trained on a small, well-aligned dataset of protein-ligand pairs, allowing us to generate diverse sets of molecules while keeping relevant to the reference targets.

2 Method

2.1 Problem Setup

Given a well-aligned dataset D of protein-ligand pairs, our objective is to predict the binding affinity and generate novel drug-like ligands that have the potential to bind to a conditioning protein structure. We cast the former as a prediction task based on geometric and relational reasoning on protein and ligand structures, whereas the latter is regarded as a protein-structure conditioned ligand generation. Let $(l, p, s) \in D$ be a pair of protein-ligand where l and p denote the representations of ligands and proteins, respectively, and s indicates the binding score between them. We define such representations for proteins and ligands that best fit with the corresponding objectives in the following sections. Additionally, Figure 4 depicts the overview of our approaches in both tasks.

2.2 Target-aware Ligand Generation

Although there exist many machine-learning approaches that generate drug-like molecules, it is challenging for graph-based or smiles-based methods to generate chemically valid ligands with high probability. Meanwhile, SELFIES (SELF-referencIng Embedded Strings) [24] is a string-based representation of molecules that is 100% robust to molecular validity. A ligand l can be defined as a string of $l = (l_1, l_2, ..., l_n)$ in which l_i is a SELFIES token, which belongs to a predefined symbol set S derived from the training dataset. We generate ligands $\hat{l} = (\hat{l}_1, \hat{l}_2, ..., \hat{l}_n)$ by computing n independent probability vectors $y = (y_1, y_2, ..., y_n)$, $y_i \in \mathbb{R}^{|S|}$. Each new token $\hat{l}_i$ is defined as $\hat{l}_i = S_j$ where $j = \underset{0 \leq j < |S|}{\text{argmax}} (y_i)$.

Let ϕ, θ, and ψ denote the encoder, decoder, and prior network in a conditional VAE framework, respectively. According to Figure 4b, in this work, $\phi : \mathbb{R}^{n \times |S|} \rightarrow \mathbb{R}^d$ and $\theta : \mathbb{R}^d \rightarrow \mathbb{R}^{n \times |S|}$ are multiplayer perceptrons (MLPs), and ψ is the PMN described in Section A where the language modeling part is excluded for computational efficiency. All the networks are jointly optimized based on Equation 12. After training, given a protein structure p, a ligand $\hat{l}$ is generated by sampling a latent vector $z \sim \mathcal{N}(\mu_\psi(p), \sigma_\psi(p))$, which is fed to the decoder θ to decode into a SELFIES representation.

Conditional Inference with Pretrained Unconditional VAE In addition to validity, the diversity of generated sets of ligands is also an important criterion in drug discovery. While classical conditional VAE trained on protein-ligand pairs can generate novel and valid molecules, the diversity and uniqueness of these samples are relatively low due to the limited amount of available data. We address this issue by adapting the work proposed in [13] from the computer vision domain to diversify the latent variables. In this framework, the decoder θ of the generative model is independent with the condition y as $p_{\theta,\psi}(x, z|y) = p_\theta(x|z)p_\psi(z|y)$, allowing θ to re-use weights of the decoder θ^* of

an unconditional VAE as both have the identical architecture. We train the model to optimize the objective as:

$$\log p_{\theta,\psi}(x|y) \geq O_{\text{for}} \triangleq \mathbb{E}_{q_\phi}[\log p_{\theta,\psi}(x|z)] - \text{KL}[q_\phi(z|x)||p_\psi(z|y)] \tag{1}$$

Different from Eq 12, both q_ϕ and p_θ in Eq 1 are not conditioned by the auxiliary covariate y. This allows conditional VAEs to use weights of ϕ^* and θ^* of a pre-trained VAE, which is trained on a diverse set of unconditional molecules, to make amortized inferences on a smaller aligned dataset of protein-ligand pairs.

3 Experiments

3.1 Binding Affinity Prediction

3.1.1 Experimental Setup

We evaluate the capability of our models on two ligand-binding datasets, DAVIS and KIBA. Our empirical results suggest that modeling long-range interactions on invariant features and leveraging sequence information provide promising performance on the task of protein-ligand affinity prediction which requires neural networks to reason on large regions of 3D structures of receptors. Both datasets contain proteins and ligands:

- **Davis** [6] has 442 proteins and 68 ligands, making up 30,056 protein-ligand binding pairs, and the binding scores are measured as K_D constants.
- **Kiba** [38] has 229 proteins and 2,111 ligands, making up 118,254 protein-ligand binding pairs, and binding affinities are measured by KIBA scores.

For fair comparisons, we follow the same train-test split settings in [30]. We use mean-squared errors (MSE), concordance index (CI), and r_m^2 to evaluate the performance. Baseline methods include KronRLS, SimBoost, SimCNN-DTA, DeepDTA, WideDTA, AttentionDTA, MATT-DTI, GraphDTA, FusionDTA, BiCompDTA, and their results are taken from [20].

3.1.2 Experimental Results

We conduct a five-fold validation (given in the dataset) to select the optimal weights for PMN. According to Table 1, our method outperforms the baselines on the DAVIS dataset by a large margin and achieves comparable performance to the best competitor on the KIBA dataset. Similar to PMN, FusionDTA also augments the representations of proteins by adopting ESM-1b [32] Transformer encoder for producing representation vectors of protein sequences. However, instead of leveraging efficient Transformers like ours, the approach utilizes full-rank Transformers for learning on long protein sequences, which requires excessive computational resources for training and fine-tuning. This can explain the trade-off between performance and training efficiency between our method and FusionDTA. On the other hand, while BiCompDTA demands carefully processed features to encode protein sequences, our approach can learn this information directly from raw structures and sequences in a data-driven manner.

3.2 Target-aware Drug Design

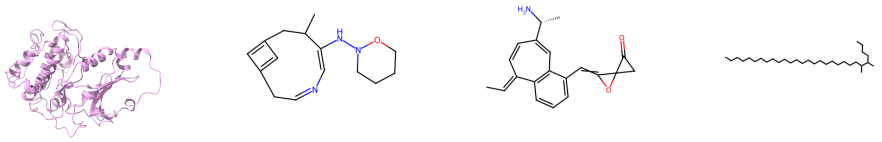
Dataset We utilize the dataset KIBA [38] for conditional molecule generation. KIBA contains 229 proteins and 2,111 ligands, and there are 118,254 protein-ligand pairs in total. For unconditional pre-training, we train a VAE on the ZINC250K dataset, which contains about 250,000 drug-like molecules. As both datasets provide SMILES as representations for the molecules, we convert them to SELFIES representations. We filter out SMILES that can not be converted to SELFIES and build a vocabulary of SELFIES blocks, which consists of 108 tokens. We split the dataset into 90 $\%$ of proteins for training and 10% of targets for testing.

3.2.1 Experimental Results

Approximation of real distributions This experiment aims to explore the capabilities of conditional VAE in generating real-world molecules given their corresponding targets. Figure **??** illustrates the FCD scores of TargetVAE trained with objectives O_{cond} in Eq. 12 and O_{for} in Eq. 1. For each

Table 1: Experimental Results on DAVIS and KIBA dataset. Results are averaged over five runs.

Approach	DAVIS			KIBA		
	MSE ↓	CI ↑	r_m^2 ↑	MSE ↓	CI ↑	r_m^2 ↑
KronRLS [28]	0.379	0.871	0.407	0.411	0.782	0.342
SimBoost [14]	0.282	0.872	0.644	0.222	0.836	0.629
SimCNN-DTA [35]	0.319	0.852	0.595	0.274	0.821	0.573
DeepDTA [52]	0.261	0.878	0.63	0.194	0.863	0.673
WideDTA [30]	0.886	0.262	—	0.875	0.179	—
AttentionDTA [49]	0.216	0.893	0.677	0.155	0.882	0.755
MATT-DTI [48]	0.227	0.891	0.683	**0.150**	0.882	0.756
GraphDTA [41]	0.258	0.884	0.656	0.162	0.879	0.736
FusionDTA [47]	0.220	0.903	0.666	0.167	**0.891**	0.699
BiCompDTA [20]	0.237	0.904	0.696	0.167	**0.891**	0.757
PMN (ours)	**0.202**	**0.906**	**0.739**	0.153	0.874	**0.767**
std	± 0.007	± 0.003	± 0.011	± 0.002	± 0.003	± 0.003



Target O_{for} O_{for} O_{cond}

Figure 1: 2D illustration of molecules generated by TargetVAE trained with two objectives O_{for} and O_{cond}

target, lower FCD scores show that TargetVAE trained by O_{for} approximate the distributions of real-world molecules better than that trained by O_{cond}. We further explore this phenomenon by visualizing the generated molecules and recognize that posterior collapse happens when TargetVAE is trained by O_{cond}. According to Figure 5, samples generated by TargetVAE trained by O_{cond} collapse to simple molecules, while the model trained by O_{for} can generate diverse samples for each protein target. Moreover, Table 2 shows the average scores of the top ten molecules in three properties (i.e. QED, SA, and pLogP). In particular, TargetVAE trained by O_{for} can generate molecules with high QED (>90) and low SA (<2), which are uncommon in drug discovery [2]. In contrast, the model trained by O_{cond} converges to molecules with high pLogP values, while possesing very low QED, indicating that they are not drug-like molecules.

Zero-shot generation to arbitrary targets As shown in Table 3, TargetVAE can generate molecules with higher binding affinities (lower K_D, in nanomoles/liter) than prior state-of-the-art RL-based or iterative methods like GCPN, MOLDQN, GraphDF, and MARS. We achieve comparable performance with LIMO, another VAE-based approach. While effective, LIMO shares a similarity with RL-based methods in that it requires training a specific property network for each protein target, resulting in inefficiency and limitations when dealing with a large number of targets. Moreover, optimizing molecules for high binding affinities may compromise other critical properties such as QED and SA, leading to sub-optimal overall performance. To prove this fact, we select two molecules having the highest QED scores and make comparisons with those produced by LIMO and GCPN in [9]. Table 4 demonstrates that while having the lowest K_D, ligands generated by LIMO are not likely

Table 2: Comparison between O_{cond} and O_{for}

Objective	QED ↑	SA ↓	pLogP ↑
O_{cond}	0.118	2.48	9.57
O_{for}	0.913	1.29	3.81

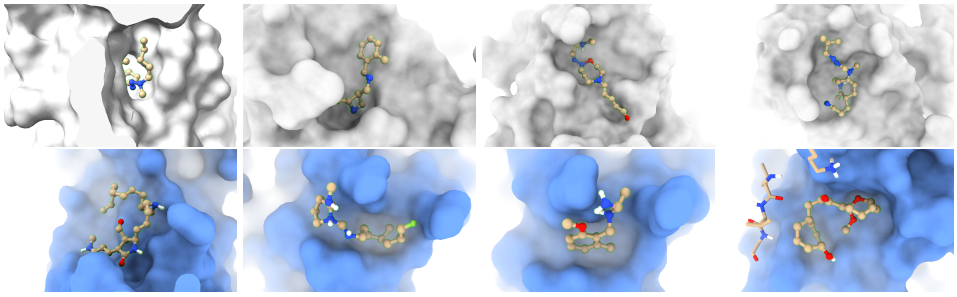


Figure 2: 3D visualizations of how ligands bind to ERR1 (first row) and ACC1 (second row). 3D conformations of generated molecules are calculated by the Obabel software [29].

drug-like molecules. In contrast, GCPN can generate drug-like molecules with QED up to 0.80, yet the method fails at producing ligands with high binding affinities. TargetVAE, on the other hand, offers the advantage of maintaining a balance among properties. Our method excels at generating ligands that possess desirable drug-like qualities, and synthetic accessibility, while still exhibiting reasonably favorable binding affinities. Finally, Figure 2 shows how the generated ligands bind to their targets.

Table 3: Top-three generated molecules with high binding affinities (shown as $K_D \downarrow$) for ESR1 and ACAA1.

Method	ESR1			ACAA1		
	1ST	2ND	3RD	1ST	2ND	3RD
GCPN [46]	6.4	6.6	8.5	75	83	84
MOLDQN [51]	373	588	1062	240	337	608
GraphDF [26]	25	47	51	370	520	590
MARS [45]	17	64	69	163	203	236
LIMO [9]	0.72	0.89	1.4	37	37	41
TargetVAE (ours)	0.55	2.7	5.1	87.3	165	177

Table 4: Trade-off between binding affinities and pharmaceutical properties (i.e. QED and SA).

Ligand	ESR1			ACAA1		
	$K_D(\downarrow)$	QED $\uparrow$	SA $\downarrow$	$K_D(\downarrow)$	QED $\uparrow$	SA $\downarrow$
LIMO #1	4.6	0.43	4.8	28	0.57	5.5
LIMO #2	2.8	0.64	4.9	31	0.44	4.9
GCPN #1	810	0.43	4.2	8500	0.69	4.2
GCPN #2	2.7×10^4	0.80	3.7	8500	0.54	4.2
TargetVAE (ours) # 1	100	0.79	6.0	420	0.77	5.82
TargetVAE (ours) # 2	40.2	0.72	5.9	662	0.71	7.64

4 Conclusion

We present two novel techniques named TargetVAE and Protein Multimodal Network (PMN) approaches to addressing protein-ligand binding prediction and target-aware ligand generation. Our proposed methods outperform other baselines in binding affinity prediction and can generate diverse sets of ligands with high binding affinities to arbitrary targets. We expect this work can help accelerate the process of drug discovery in the future.

References

[1] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[2] G. R. Bickerton, G. V. Paolini, J. Besnard, S. Muresan, and A. L. Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90–98, 2012.

[3] S. K. Burley, H. M. Berman, C. Bhikadiya, C. Bi, L. Chen, L. Di Costanzo, C. Christie, K. Dalenberg, J. M. Duarte, S. Dutta, et al. Rcsb protein data bank: biological macromolecular structures enabling research and education in fundamental biology, biomedicine, biotechnology and energy. *Nucleic acids research*, 47(D1):D464–D474, 2019.

[4] C. Cai, T. S. Hy, R. Yu, and Y. Wang. On the connection between mpnn and graph transformer. *International Conference of Machine Learning*, 2023.

[5] K. M. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Q. Davis, A. Mohiuddin, L. Kaiser, D. B. Belanger, L. J. Colwell, and A. Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=Ua6zuk0WRH`.

[6] M. I. Davis, J. P. Hunt, S. Herrgard, P. Ciceri, L. M. Wodicka, G. Pallares, M. Hocker, D. K. Treiber, and P. P. Zarrinkar. Comprehensive analysis of kinase inhibitor selectivity. *Nature Biotechnology*, 29(11):1046–1051, Nov. 2011.

[7] N. De Cao and T. Kipf. MolGAN: An implicit generative model for small molecular graphs. *ICML 2018 workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018.

[8] V. P. Dwivedi, L. Rampášek, M. Galkin, A. Parviz, G. Wolf, A. T. Luu, and D. Beaini. Long range graph benchmark. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 22326–22340. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/8c3c666820ea055a77726d66fc7d447f-Paper-Datasets_and_Benchmarks.pdf`.

[9] P. Eckmann, K. Sun, B. Zhao, M. Feng, M. K. Gilson, and R. Yu. Limo: Latent inceptionism for targeted molecule generation. 2022.

[10] M. Fréchet. Sur la distance de deux lois de probabilité. In *Annales de l'ISUP*, volume 6, pages 183–198, 1957.

[11] V. Gapsys, D. F. Hahn, G. Tresadern, D. L. Mobley, M. Rampp, and B. L. de Groot. Pre-exascale computing of protein-ligand binding free energies with open source software for drug design. *Journal of chemical information and modeling*, 62(5):1172–1177, 2022.

[12] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1263–1272. JMLR.org, 2017.

[13] W. Harvey, S. Naderiparizi, and F. Wood. Conditional image generation by conditioning variational auto-encoders. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=7MV6uLzOChW`.

[14] T. He, M. Heidemeyer, F. Ban, A. Cherkasov, and M. Ester. Simboost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines. *Journal of Cheminformatics*, 9(1):24, Apr 2017. ISSN 1758-2946. doi: 10.1186/s13321-017-0209-z. URL `https://doi.org/10.1186/s13321-017-0209-z`.

[15] J. P. Hughes, S. Rees, S. B. Kalindjian, and K. L. Philpott. Principles of early drug discovery. *British Journal of Pharmacology*, 162(6):1239–1249, 2011.

[16] O. Ivanov, M. Figurnov, and D. Vetrov. Variational autoencoder with arbitrary conditioning. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=SyxtJh0qYm`.

[17] W. Jin, R. Barzilay, and T. Jaakkola. Junction tree variational autoencoder for molecular graph generation. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2323–2332. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/jin18a.html.

[18] W. Jin, D. Barzilay, and T. Jaakkola. Hierarchical generation of molecular graphs using structural motifs. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4839–4848. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/jin20a.html.

[19] B. Jing, S. Eismann, P. Suriana, R. J. L. Townshend, and R. Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=1YLJDvSx6J4.

[20] M. Kalemati, M. Zamani Emani, and S. Koohi. Bicomp-dta: Drug-target binding affinity prediction through complementary biological-related and compression-based featurization approach. *PLOS Computational Biology*, 19(3):1–28, 03 2023. doi: 10.1371/journal.pcbi.1011036. URL https://doi.org/10.1371/journal.pcbi.1011036.

[21] J. Kim, D. T. Nguyen, S. Min, S. Cho, M. Lee, H. Lee, and S. Hong. Pure transformers are powerful graph learners. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=um2BxfgkT2_.

[22] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[23] N. Kitaev, L. Kaiser, and A. Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgNKkHtvB.

[24] M. Krenn, F. Häse, A. Nigam, P. Friederich, and A. Aspuru-Guzik. Self-referencing embedded strings (selfies): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, oct 2020. doi: 10.1088/2632-2153/aba947. URL https://dx.doi.org/10.1088/2632-2153/aba947.

[25] S. Luo, J. Guan, J. Ma, and J. Peng. A 3d generative model for structure-based drug design. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=yDwfVD_odRo.

[26] Y. Luo, K. Yan, and S. Ji. Graphdf: A discrete flow model for molecular graph generation. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7192–7203. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/luo21a.html.

[27] A. Mayr, G. Klambauer, T. Unterthiner, M. Steijaert, J. K. Wegner, H. Ceulemans, D.-A. Clevert, and S. Hochreiter. Large-scale comparison of machine learning methods for drug target prediction on chembl. *Chemical science*, 9(24):5441–5451, 2018.

[28] A. C. Nascimento, R. B. Prudêncio, and I. G. Costa. A multiple kernel learning algorithm for drug-target interaction prediction. *BMC bioinformatics*, 17:1–16, 2016.

[29] N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, and G. R. Hutchison. Open babel: An open chemical toolbox. *Journal of cheminformatics*, 3(1):1–14, 2011.

[30] H. Öztürk, E. Ozkirimli, and A. Özgür. Widedta: prediction of drug-target binding affinity. *arXiv preprint arXiv:1902.04166*, 2019.

[31] K. Preuer, P. Renz, T. Unterthiner, S. Hochreiter, and G. Klambauer. Fréchet chemnet distance: A metric for generative models for molecules in drug discovery. *Journal of Chemical Information and Modeling*, 58(9):1736–1741, Sep 2018. ISSN 1549-9596. doi: 10.1021/acs.jcim.8b00234. URL https://doi.org/10.1021/acs.jcim.8b00234.

[32] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

[33] D. Rogers and M. Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

[34] A. Roy*, M. T. Saffar*, D. Grangier, and A. Vaswani. Efficient content-based sparse attention with routing transformers, 2020. URL `https://openreview.net/forum?id=B1gjs6EtDr`.

[35] J. Shim, Z.-Y. Hong, I. Sohn, and C. Hwang. Prediction of drug–target binding affinity using similarity-based convolutional neural network. *Scientific Reports*, 11(1):4416, 2021.

[36] M. Simonovsky and N. Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. *ArXiv*, abs/1802.03480, 2018.

[37] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL `https://proceedings.neurips.cc/paper_files/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf`.

[38] J. Tang, A. Szwajda, S. Shakyawar, T. Xu, P. Hintsanen, K. Wennerberg, and T. Aittokallio. Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis. *Journal of Chemical Information and Modeling*, 54(3):735–743, 2014.

[39] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rJXMpikCZ`.

[40] G. M. Verkhivker, D. Bouzida, D. K. Gehlhaar, P. A. Rejto, S. Arthurs, A. B. Colson, S. T. Freer, V. Larson, B. A. Luty, T. Marrone, et al. Binding energy landscapes of ligand-protein complexes and molecular docking: Principles, methods, and validation experiments. In *Combinatorial Library Design and Evaluation*, pages 177–216. CRC Press, 2001.

[41] T. Voitsitskyi, R. Stratiichuk, I. Koleiev, L. Popryho, Z. Ostrovsky, P. Henitsoi, I. Khropachov, V. Vozniak, R. Zhytar, D. Nechepurenko, S. Yesylevskyy, A. Nafiiev, and S. Starosyla. 3dprotdta: a deep learning model for drug-target affinity prediction based on residue-level protein graphs. *RSC Adv.*, 13:10261–10272, 2023. doi: 10.1039/D3RA00281K. URL `http://dx.doi.org/10.1039/D3RA00281K`.

[42] Z. Wan, J. Zhang, D. Chen, and J. Liao. High-fidelity pluralistic image completion with transformers. *arXiv preprint arXiv:2103.14031*, 2021.

[43] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma. Linformer: Self-attention with linear complexity, 2020.

[44] L. N. Wasserstein et al. Markov processes over denumerable products of spaces describing large systems of automata. *Problems of Information Transmission*, 5(3):47–52, 1969.

[45] Y. Xie, C. Shi, H. Zhou, Y. Yang, W. Zhang, Y. Yu, and L. Li. Mars: Markov molecular sampling for multi-objective drug discovery. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=kHSu4ebxFXY`.

[46] J. You, B. Liu, Z. Ying, V. Pande, and J. Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper_files/paper/2018/file/d60678e8f2ba9c540798ebbde31177e8-Paper.pdf`.

[47] W. Yuan, G. Chen, and C. Y.-C. Chen. FusionDTA: attention-based feature polymerizer and knowledge distillation for drug-target binding affinity prediction. *Briefings in Bioinformatics*, 23(1), 12 2021. ISSN 1477-4054. doi: 10.1093/bib/bbab506. URL `https://doi.org/10.1093/bib/bbab506`. bbab506.

[48] Y. Zeng, X. Chen, Y. Luo, X. Li, and D. Peng. Deep drug-target binding affinity prediction with multiple attention blocks. *Briefings in Bioinformatics*, 22(5), 04 2021. ISSN 1477-4054. doi: 10.1093/bib/bbab117. URL https://doi.org/10.1093/bib/bbab117. bbab117.

[49] Q. Zhao, F. Xiao, M. Yang, Y. Li, and J. Wang. Attentiondta: prediction of drug–target binding affinity using attention model. In *2019 IEEE international conference on Bioinformatics and Biomedicine (BIBM)*, pages 64–69. IEEE, 2019.

[50] C. Zheng, T.-J. Cham, and J. Cai. Pluralistic image completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[51] Z. Zhou, S. Kearnes, L. Li, R. N. Zare, and P. Riley. Optimization of molecules via deep reinforcement learning. *Scientific reports*, 9(1):1–10, 2019.

[52] H. Öztürk, A. Özgür, and E. Ozkirimli. DeepDTA: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 09 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty593. URL https://doi.org/10.1093/bioinformatics/bty593.

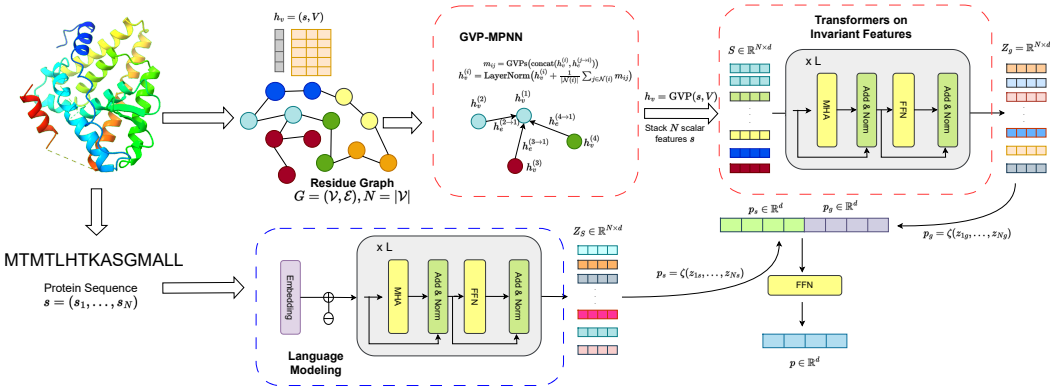


Figure 3: Overview of our Protein Multimodal Network (PMN)

.

A Protein Multimodal Network (PMN)

Proteins are complex structures that consist of long chains of residues/amino acids. Each amino acid is a molecule with 3D structures, and a combination of hundreds to thousands of residues determines the unique 3D structure of a specific protein and its functions. It is worth noting that while two residues are distant along the protein sequence, they could be close to each other in three-dimensional space. This is our key observation to design a novel framework that can unify different representations of proteins in an end-to-end learning manner. In the field of graph learning, the conventional graph neural networks based on the message passing scheme [12] that propagates and aggregates information of each node to and from its local neighborhoods have been shown to be incapable of capturing the long-range interactions in a large-diameter graph [8]. Meanwhile, the graph Transformers that considers all pairwise node interactions via the self-attention mechanism can successfully capture the long-range dependencies [21, 4]. Since proteins can be seen as long-range graphs, we utilize sequential and graph Transformers to encode both sequences and 3D graphs of residues and combine them to create a unified representation for a large protein, making our model operate on multi-modalities of proteins.

Long-range Modeling on 3D Structures According to Figure 3, there are three major components in the 3D modeling part, including a local encoder, a GVP module, and a global Transformers encoder (Trans). We use a message-passing network (MPNN) in which dense layers are replaced by GVP to operate on invariant features [19]:

$$m_{ij} = \text{GVPs}\big(\text{concat}(h_v^{(i)}, h_e^{(j\to i)})\big) \tag{2}$$

$$h_v^{(i)} = \text{LayerNorm}\big(h_v^{(i)} + \frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)} m_{ij}\big) \tag{3}$$

Where m_{ij} computed by a module of three GVP layers denotes the message propagated from node j to i. Also, $h_v^{(i)}$ and $h_e^{(j\to i)}$ indicate the embeddings of node i and edge $(j \to i)$ and are tuples of scalar and vector features as described in Section **??**. The local encoder outputs a tuple of scalar and vector features for each residue node, which are rotationally invariant and equivariant, respectively. We utilize a GVP module to update the tuple $h_v = (s, V)$ as $(s', V') = \text{GVP}\big((s, V)\big)$, and we take the invariant scalar feature $s' \in \mathbb{R}^d$ as the node embedding for successor modules. The resulting tensor $S \in \mathbb{R}^{N\times d}$, in which row i indicates a d-dimensional scalar feature s_i of node i, is passed to a L-layer Transformers encoder:

$$Q_l = Z_{l-1}W_l^Q, K_l = Z_{l-1}W_l^K, V_l = Z_{l-1}W_l^V \tag{4}$$

$$H_l = \text{MultiheadAttention}(Q_l, K_l, V_l) \tag{5}$$

$$Z_l = \text{LayerNorm}(Z_{l-1} + \text{FFN}(H_l)) \tag{6}$$

Here, $Z_0 \triangleq S$ and $\{W_l^Q, W_l^K, W_l^V\}_{l=1}^L \in \mathbb{R}^{d\times d_k}$, and $Z_g \triangleq Z_L$ denotes the final node embeddings produced by the network. Notably, this global encoder allows residue nodes to attend to other nodes

on a large protein graph, especially those that are distant from them (i.e. long-range modeling). Finally, we aggregate node embeddings by a row-wise *Aggregator* ζ (e.g., mean, max, sum, etc.) to produce an embedding for the protein structure $p_g = \zeta(Z_g) \in \mathbb{R}^d$.

Language Modeling on Protein Sequence A protein can be represented as a sequence $s = (s_0, s_1, ..., s_n)$ in which $s_i \in \mathbb{R}^{20}$ is a one-hot vector indicating one in a total of 20 types of residues. We utilize Transformer-based language models, where the layers are the same as in Eq. (4, 5, 6), to compute the text representation of this protein sequence with the initial embeddings $Z_0 = [z_1, z_2, ..., z_n] \in \mathbb{R}^{n \times d}$ with $z_i \in \mathbb{R}^d$ is calculated as $z_i = \text{Embed}(s_i) + p_i$.

Here, p_i is the positional encoding feature added at each token i. Then, we define $p_s = \zeta(Z_s) \in \mathbb{R}^d$ as the global representation for the entire protein sequence. Notably, there may be hundreds to thousands of residues in a long-chain protein, so we utilize efficient Transformers [34, 5, 23] to reduce the computational complexity. At the end of the network, we calculate a unified representation of the protein $p = W_2\text{ReLU}(W_1(\text{concat}(p_g, p_s)) + b_1) + b_2$.

A.1 Binding Affinity Prediction

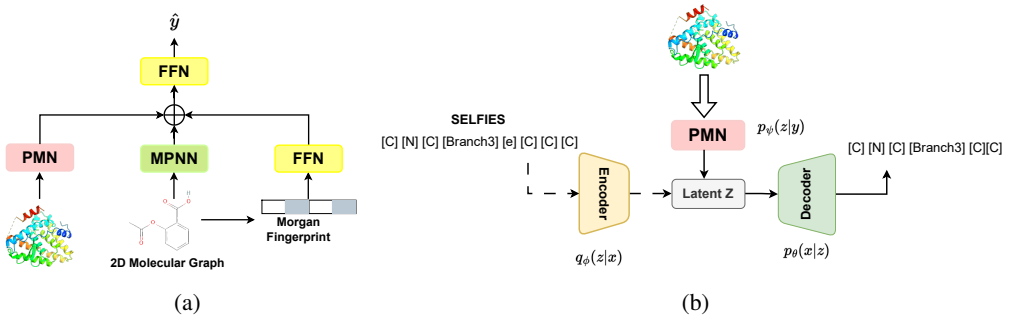


Figure 4: Figure 4a is a framework for predicting binding affinities between proteins and ligands. Figure 4b is TargetVAE with an encoder, decoder, and a prior network. The PMN prior network computes the conditions from protein structures for constructing the latent space of the VAE framework, which learns to generate SELFIES representations of molecules

Figure 4a illustrates our proposed approach to predicting the binding affinities between ligands l and protein targets p. A drug-like ligand is represented by a 2D molecular graph G_l and a binary Morgan Fingerprint vector $v_m \in \mathbb{R}^{2,048}$[33], which embodies critical properties of chemical structures. G and v_m are passed to a message-passing neural network (MPNN) and feed-forward network (FFN). Also, the given protein structure p is sent to the protein multimodal network (PMN) mentioned in Section A:

$$z_{l1} = \text{MPNN}(G_l) \tag{7}$$

$$z_{l2} = W_{m2}\text{ReLU}(W_{m1}v_m + b_{m1}) + b_{m2} \tag{8}$$

$$z_p = \text{PMN}(p) \tag{9}$$

Then, z_{l1}, z_{l2}, and z_p are combined to yield a unified input for the top FFN to output a scalar value $\hat{y}$ denoting the predicted binding affinity score:

$$\hat{y} = W_{u2}\text{ReLU}(W_{u1}\text{concat}(z_{l1}, z_{l2}, z_p) + b_{u1}) + b_{u2} \tag{10}$$

B Additional Experiments

C Variational Auto-Encoders

A variational auto-encoder (VAE) is regarded as an auto-encoding variational Bayes model [22] that comprises two components, including a generative model and an inference model (also known as probabilistic encoder). The former uses a probabilistic decoder $p_\theta(x|z)$ and a prior $p_\psi(z)$ to define a joint distribution $p_{\theta,\psi}(x, z) = p_\theta(x|z)p_\psi(z)$ between latent variables z and data x; in addition,

Kingma and Welling [22] let $p_\psi(z)$ be isotropic Gaussian. An ideal generative model should learn to maximize the log-likelihood $\log p_{\theta,\psi}(x) = \log \int p_{\theta,\psi}(x,z)dz$. However, this is intractable as marginalization over the latent space is usually infeasible with realistic data. VAE alleviates this issue by using an encoder $q_\phi(z|x)$ to approximate the true posterior distribution of the latent space and maximize the evidence lower bound (ELBO) over each training sample x:

$$\log p_{\theta,\psi}(x) \geq \mathbb{E}_{q_\phi}[\log p_{\theta,\psi}(x|z)] - \text{KL}[q_\phi(z|x)||p_\psi(z)] \tag{11}$$

In conditional VAE, the generative component is augmented by auxiliary covariates y. Given a condition y, the generative model defines a conditional joint distribution of z and x as $p_{\theta,\psi}(x,z|y) = p_\theta(x|y,z)p_\psi(z|y)$. Similarly, the condition inputs are integrated into the encoder as $q_\phi(z|x,y)$. These two extensions establish a prominent conditional VAE model [37, 50, 16, 42] that is trained to maximize the conditional ELBO as:

$$\log p_{\theta,\psi}(x|y) \geq O_{\text{cond}} \triangleq \mathbb{E}_{q_\phi}[\log p_{\theta,\psi}(x|y,z)] - \text{KL}[q_\phi(z|x,y)||p_\psi(z|y)] \tag{12}$$

D Geometric Vector Perceptron

Jing et al. [19] propose Geometric Vector Perceptron (GVP) as a simple module for learning vector-valued and scalar-valued functions over geometric vectors and scalars. The module transforms an input tuple (s, V) of scalar features $s \in \mathbb{R}^n$ and vector features $V \in \mathbb{R}^{\mu\times 3}$ into a new tuple $(s', V') \in \mathbb{R}^m \times \mathbb{R}^{\nu\times 3}$. According to Algorithm 1, GVP consists of two separate linear transformations W_m and W_h that work on the scalar and vector features respectively, followed by nonlinearities σ and σ^+. Before being transformed, the scalar feature s is concatenated with the L_2norm of the vector feature V. This enables GVP to extract the rotation-invariant information from the input vector V. Moreover, an additional transformation W_μ is used to control the dimensionality of the output vector V', making it independent of the number of norms extracted. Albeit simple, GVP is an effective module that guarantees desired properties of invariance/equivariance and expressiveness. The scalar and vector outputs of GVP are invariant and equivariant respectively, with respect to an arbitrary composition R of rotations and reflections in 3D Euclidean space. In other words, if GVP$(s, V) = (s', V')$, then GVP$(s, R(V)) = (s', R(V'))$.

Algorithm 1 Geometric Vector Perceptron

Input: Scalar and vector features $(s, V) \in \mathbb{R}^n \times \mathbb{R}^{\mu\times 3}$
Output: Scalar and vector features $(s', V') \in \mathbb{R}^m \times \mathbb{R}^{\nu\times 3}$
$h \leftarrow \max(\mu, \nu)$
GVP:
$V_h \leftarrow W_h V \quad \in \mathbb{R}^{h\times 3}$
$V_\mu \leftarrow W_\mu V_h \quad \in \mathbb{R}^{\mu\times 3}$
$s_h \leftarrow ||V_h||_2$ (row-wise) $\quad \in \mathbb{R}^h$
$v_\mu \leftarrow ||V_\mu||_2$ (row-wise $\quad \in \mathbb{R}^\mu$
$s_{h+n} \leftarrow \text{concat}(s_h, s) \quad \in \mathbb{R}^{h+n}$
$s_m \leftarrow W_m s_{h+n} + b \quad \in \mathbb{R}^m$
$s' \leftarrow \sigma(s_m) \quad \in \mathbb{R}^m$
$V' \leftarrow \sigma^+(v_\mu) \odot V_\mu$ (row-wise multiplication) $\quad \in \mathbb{R}^{\mu\times 3}$
return (s', V')

E Implementation Details

Binding Affinity Prediction The proposed model was developed in PyTorch, and experiments were carried out on an NVIDIA A100 GPU. For each experiment, we trained the model in 700 epochs. The models are optimized by Adam optimizer, with a learning rate of 0.0001, and a batch size of 128. We use a two-layer graph attention network (GAT) [39] to learn the representations of 2D molecular graphs. For the protein multimodal network, we reuse a three-layer message passing network proposed in [19], followed by two layers of Performer [5] to learn the global context of protein structures. Hidden dimensions equal 256 in all layers. To learn sequential information, we use four layers of Linformer [43] with 8 heads and embedding dimensions of 128.

Ligand Generation We trained an unconditional VAE based on setup in [9] and reused the pre-trained weights of its encoder and decoder as initialization for TargetVAE. For the prior network, we use PMN with the same setting used in the binding affinity prediction task, but the language modeling component is detached for efficiency purposes. The models were trained with the coefficient $\beta = 0.1$, which controls the KL term in ELBO, in 30 epochs with a learning rate of 0.0001, and batch size equals 256.

F Metrics

F.1 Binding Affinity Prediction

- Mean squared error (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{13}$$

  where n is the number of samples, y_i is the observed value, and $\hat{y_j}$ is the predicted value.

- Concordance Index:

$$\text{CI} = \frac{1}{Z}\sum_{\delta_i > \delta_j} h(\hat{y}_i - \hat{y}_j) \tag{14}$$

  $\hat{y}_i$ denotes the prediction for the larger affinity δ_i, $\hat{y}_j$ is the predicted value for the smaller affinity δ_j. Z is the normalization constant, and $h(x)$ is defined as:

$$h(x) = \begin{cases} 1 & x > 0 \\ 0.5 & x = 0 \\ 0 & x < 0 \end{cases} \tag{15}$$

- r_m^2 Index:

$$r_m^2 = r^2 \times \left(1 - \sqrt{r^2 - r_0^2}\right) \tag{16}$$

  where r^2 and r_0^2 are the squared correlation coefficients with and without intercepts respectively.

F.2 Ligand Generation

Fréchet ChemNet Distance (FCD) [31] calculates the distance between the distribution of $p_w(\cdot)$ of real-world molecules and the distribution of $p_g(\cdot)$ of molecules generated by the model. Numerical representations of the molecules are obtained by the activations of the penultimate layer of ChemNet [27]. For each distribution p, the mean and covariance are computed from the activations, which correspond to the molecules in p, assuming that the hidden representations follow a multi-dimensional Gaussian. Let (μ_w, Σ_w) and (μ_g, Σ_g) denote means and covariances of p_w and p_g respectively. Then, the Fréchet Distance [10] (i.e. Wasserstein-2 distance [44]) is used to calculate the two Gaussians:

$$d^2(p_g, p_w) = ||\mu_g - \mu_w||^2 + \text{Tr}(\Sigma_g + \Sigma_w - 2(\Sigma_g\Sigma_w)^{1/2}) \tag{17}$$

Our paper reports the FCD as $d^2(p_g, p_w)$.

G Discussions

In this work, we seek an effective approach to generate ligands by conditioning the entire structures of proteins, while also introducing a novel architecture for operating on these structures. For protein modeling, using efficient transformers can reduce the time complexity to a linear scale with respect to the number of residues. It is worthwhile to explore methods that can improve the modeling of long-range interactions on large protein graphs while maintaining a subquadratic complexity. In the context of target-aware ligand generation, although TargetVAE can perform on par with expensive RL-based methods, there still exist gaps between theoretical results and practical applications in the field of drug discovery. To bridge these gaps, wet lab experiments should be integrated into future work, allowing better evaluations of the models and enhancing their practical applicability.

H Visualization

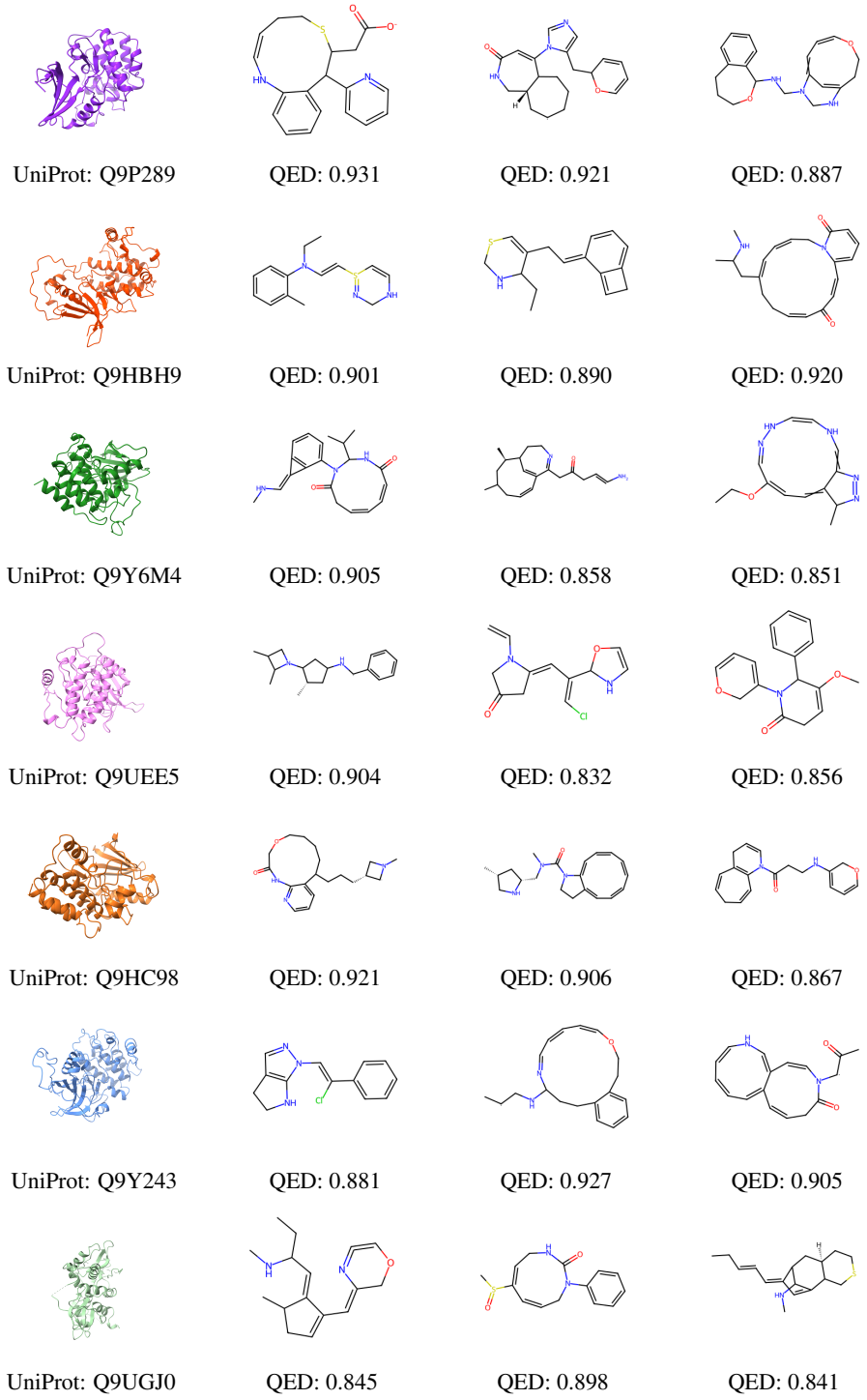


Figure 5: More visualizations of molecules generated by TargetVAE, given the corresponding targets.