Appendix

1 Method

1.1 Sketching

In Pocket-based Sketching, we present an algorithm to obtain the molecule shape, which is of the appropriate size and complementary to protein pockets. The algorithm uses a seed shape to intersect with the pocket gradually.

To obtain the seed shape, we first randomly sample several molecules from ZINC, then use the overlapping of their shapes as the seed shape. By using the overlapping strategy, we make the sketched pseudo molecular shapes more native-molecule-like and not overly dependent on one specific molecule. Algorithm 1 is the pseudo-code.

We set the volume threshold to 300Å^3, the average volume of molecules. The step size is the same as the voxel resolution, i.e., 0.5Å. We initially place the seed shape in a random position as long as the seed shape and the pocket shape do not overlap.

1.2 Generating

Pilot Experiments of Tokenizing Methods We evaluate different molecule tokenizing methods based on three principles (preserving functional groups, appropriate vocabulary size, and no circles structures) and propose our approach in the context of pre-training. There are several popular tokenizing methods for drug design: Atom-based [1, 2] shatters a molecule into atoms. Cut Single Bond [3] cuts all single bonds in a molecule. Both RECAP [4] and BRICS [5] use chemical reaction rules for preserving functional groups. The results in Table 1 show: (1) Atom-based and Cut Single Bond can not preserve functional groups. Because they do not distinguish whether a chemical bond belongs to a functional group, they ruin the molecular structures that are vital for determining molecule properties. (2) Two chemical-reaction-aware methods behave differently. As RECAP has more conservative rules, it produces a vocabulary at least one magnitude larger than its counterparts.

Because BRICS shows a better balance among these principles, we adapt it to serve our pre-trained model. Furthermore, because over 60% of the out-of-vocabulary problem, an essential factor affecting the pre-trained model quality [6], is caused by the combinations between rings and other structures through single bonds, we add an extra rule to BRICS: cut all single bonds attached to a ring.

Details of Discretization To stabilize the training process, we discretize two continuous variables, i.e., translation vector and rotation quaternion. Specifically, we map them into two discrete spaces.

For the translation vector, its discrete space is grids in 3D space (see Figure 1a). Briefly speaking, we represent this continuous vector with the discrete index of the grid. We represent the i-th translation bin as the coordinate of its centre $t_i^{\text{bin}} \in \mathbb{R}^3$. The discretization of any continuous translation operator $t \in \mathbb{R}^3$ can be computed by $\arg\min_i \|t_i^{\text{bin}}, t\|_2$. The total number of grids is $21,952$.

Because the rotation axis and rotation angle can determine the rotation quaternion, the discrete space of the quaternion contains two parts. As shown in Figure 1b, we first enumerate rotation axes (x, y, z) in 3D space, then for each axis, we list the rotation angle every θ degrees. We represent the i-th rotation bin (stands for rotating θ_i degrees around an axis (x_i, y_i, z_i)) as a quaternion

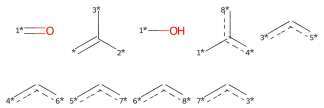
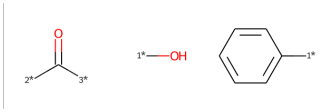
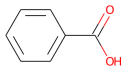
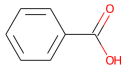
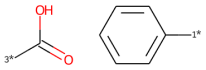
36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Algorithm 1 Pocket-based Sketching

Input: a function λ measuring the volume of a shape, a threshold t, a step size α
a pocket shape S_p located at C_p, a seed shape S_s located at C_s where $\lambda\left(\bigcap\left(S_p, S_s\right)\right) = 0$
Output: The sampled molecule shape S_d.

```
while True do
    V ← λ(⋂(S_p, S_s))
    if V ≥ t then
        S_d ← ⋂(S_p, S_s)
        break
    end if
    C_s = αC_p + (1 − α)C_s
end while
```

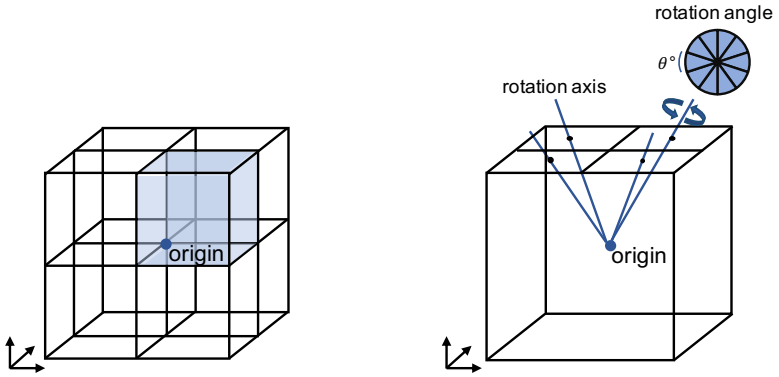
Table 1: Comparison of different tokenizing methods. Coverage is the percentage of novel molecules that the vocabulary can handle.

Method	Coverage	Vocabulary Size	Case
Atom-based	100.0	49	
Cut Single Bond	74.3	16,920	
RECAP	40.1	289,188	
BRICS	61.2	49,339	
Ours	70.4	23,896	

$q_i^{\text{bin}} = (x_i, y_i, z_i, \theta_i) \in \mathbb{R}^4$. The discretization of any continuous rotation operator $q \in \mathbb{R}^4$ can be computed by $\arg\min_i \|q_i^{\text{bin}}, q\|_2$. The total number of bins is $8,763$. To be precise, we enumerate 363 axes and 24 angles (i.e., 15 degrees per angle).

Another reason for the discretization is to avoid the discontinuity of quaternions when optimizing them [7, 8]. Specifically, there are several approaches to parameterize the rotation operator: quaternion [9], euler-angle [10], and SO(3) group (i.e., the rotation matrix) [11]. Quaternion and euler-angle are sometimes ambiguous and discontinuous. For example, the rotation operator is periodic, rotating $180°$ is equal to rotating $-180°$ and rotating $179.9°$ is very close to rotating $-179.9°$. Without discretization, we will excessively penalize the model according to the mean-square-error (i.e., $[179.9 - (-179.9)]^2 = 359.8^2$). While in this paper, we convert a regression problem into a classification one with discretization, thus avoiding the above issues. Instead of discretizing quaternions, AlphaFold [12] avoids such issues by regarding the quaternion as an intermediate variable and not optimizing the quaternion directly. We leave adopting AlphaFold's strategy for future work.

Greedy Algorithm for Connecting Fragments Algorithm 2 is responsible for converting the separated fragments into a complete 3D molecule. To be specific, we first place all the fragments in 3D space according to the predictions of SHAPE2MOL. Then, for each time, we greedily chose the



(a) Discrete space for translation vector (b) Discrete space for rotation quaternion

Figure 1: Discretization of translation vector and rotation quaternion. Both origins locate at the centroid of the fragment.

two closest breakpoints from different fragments and connect the fragments through the breakpoints. By repeating the process, the fragments get larger and larger. When there are not enough breakpoints to connect, we return the largest fragment as the final molecule. For potential residual breakpoints, we attach carbon atoms to them for molecular validity.

Algorithm 2 Greedy Algorithm for Connecting Fragments

Input: Fragment sequence $S = [(C_i, P_i, R_i)]_{i=0}^{l_s}$, fragment vocabulary V.
Output: A whole molecule M formed by the fragments.

```
B ← {}
F ← {}
for i ← 0 to l_s do
    (C_i, P_i, R_i) ← S[i]
    f ← query_vocabulary(V, C_i)
    f ← rotate(translate(f, P_i), R_i)          ▷ Place the fragment in 3D space
    b ← get_breakpoints(f)                      ▷ Return all the breakpoints in current fragment
    F ← F ∪ {f}
    B ← B ∪ {b}
end for
while |B| ≥ 2 do
    b_1, b_2 ← get_breakpoints(B)               ▷ Chose two nearest breakpoints from different fragments
    if b_1 is None or b_2 is None then
        break
    end if
    f_1 ← get_fragment(F, b_1)                  ▷ Retrieve the corresponding fragment of the breakpoint
    f_2 ← get_fragment(F, b_2)
    M_partial ← attach_fragments(f_1, b_1, f_2, b_2)   ▷ Connect fragments through the breakpoints
    B ← B \ {b_1, b_2}
    F ← F \ {f_1, f_2}
    F ← F ∪ {M_partial}
end while
M ← get_largest_fragment(F)                     ▷ Chose the largest fragment as the output molecule
if |B| ≠ 0 then
    M ← attach_carbon(M, B)                     ▷ Handle the remaining breakpoints by attaching carbon atoms
end if
```

1.3 More Details of SHAPE2MOL

Output of Shape Encoder The output of the shape encoder is the continuous representation of each 3D patch, which contain the geometric information of input molecular shape. It will serve as the context of the decoder to constrain the shape of generated molecules.

Input of Shape Decoder The input of the shape decoder at the time step t is the fragment category, rotation quaternion, and translation vector from the output of the previous step $t-1$. We use them to tell the model how exactly a fragment is placed in 3D space so that the model can generate the next fragment connected with it. The output of the shape encoder is fed into the decoder as the geometric context, through the cross-attention module.

Hyperparameters of SHAPE2MOL Both encoder and decoder of SHAPE2MOL have 12 stacked 8-head Transformer layers, whose model size is 1024 and feedforward dimension is 4096. We use ReLu as the activation function and set the position embedding learnable in the training process.

For the encoder, following liGAN, we set the resolution of the voxelized shape to 0.5Å and the side length of the spanned cube to 14Å. The size of 3D patches is $4 \times 4 \times 4$, which allows us to handle a large number of voxels. Specifically, we reshape $21,952$ voxels into 343 3D patches in the paper.

The decoder has 3 different embeddings, denoting the fragment category, the discretized rotation, and the discretized translation, respectively. Similarly, we also use 3 different output heads to predict the category, rotation, and translation. The output heads share their weights with the embeddings. The total number of model parameters is 650M.

2 Experiments

2.1 Metrics

- Uniqueness **(Uniq)** is the percentage of unique molecules among all generated results.
- Novelty **(Nov)** is the percentage of generated molecules with Tanimoto similarity [13] less than 0.4 compared to its nearest neighbor in existing ligands [14].
- Diversity **(Div)** is the internal diversity of the generated molecules and calculated as $\frac{2}{n(n-1)} \sum_{x \neq x'} 1 - sim(x, x^{'})$.
- Success rate **(Succ)** is the percentage of generated molecules that pass the predefined thresholds for the desired properties. Specifically, we based on a widely used rule of thumb [15] to set the thresholds as $\mathrm{QED} \geq 0.25$, $\mathrm{SA}_{\mathrm{score}} \geq 0.59$ and $\mathrm{Vina}_{\mathrm{score}} \leq -8.18$ kcal/mol, where QED is the index of drug-likeness [16] and $\mathrm{SA}_{\mathrm{score}}$ is the index of synthetic accessibility [17].
- Product **(Prod)** is the product of the four metrics above and serves as a comprehensive evaluation.
- Median Vina Score **(Median)** is the median of $\mathrm{Vina}_{\mathrm{score}}$. $\mathrm{Vina}_{\mathrm{score}}$ is the binding energy evaluation using the global energy optimization algorithm in Vina [18], which can reflect the binding affinity between proteins and molecules.

2.2 More Ablation Study of Generating

Model Variants As shown in Figure 3, we study two variants in the model design: discretization and robust training. The discretization consistently improves the result, while the continuous variant is worse. A possible reason is a non-linear relationship between quaternions and rotation angles. [1] We observe minor improvement for finer granularity and leave it for future exploration. We also study the ability of handling shape noise caused by a possibly large pocket size. The model trained with shape noise (i.e. robust training) shows better performance when test data are noisy, while the performance under standard training drops clearly.

[1] For example, if a molecule rotates 90° along an axis, the mean squared error of quaternion is 1; when the angle is 10°, the error is 0.98.

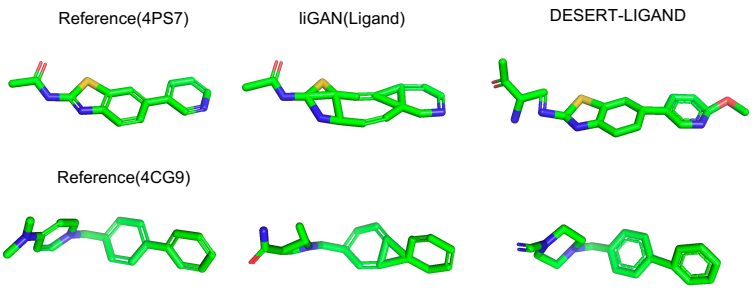


Figure 2: More cases from liGAN and DESERT.

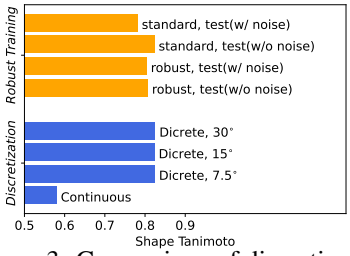


Figure 3: Comparison of discretization and robust training.

Table 2: Comparison of difference decoding strategy.

Sampling Method	Div	Prod	Median
Beam Search (Beam=10)	0.70	0.00	-5.87
Greedy Decoding	0.65	0.00	-5.83
Top K (K=10)	0.91	0.04	-5.99
Top P (P=0.95)	0.93	0.06	-6.01
+ post-processing	0.92	0.17	-7.34

Table 3: Combine protein chemical information.

Method	Prod	Median
DESERT-POCKET	0.55	-9.32
+ chemical(Weak)	0.53	-9.19
+ chemical(Strong)	0.52	-9.03

Table 4: Comparison of difference seed shape.

Seed Shape	Succ	Prod	Median
None	82.0	0.74	-8.85
Sphere	75.5	0.68	-9.13
Molecules	61.0	0.55	-9.32

Decoding Strategy We replace the Nucleus sampling method, i.e., Top P sampling, in the decoding strategy with Beam Search [19], Greedy Decoding [20], and Top K sampling [20] to study the influence of different sampling methods on the performance of DESERT. As shown in Table 2, Beam Search and Greedy Decoding are poor at providing diverse molecules, which suggests that the output probability is concentrated. They can only get several fragments with high output probability. When low probability fragments are obtained, the ranking mechanism in Beam Search drops them in the subsequent decoding step. Due to their internal annealing mechanism, both Top K and Top P can provide relatively diverse results.

Chemical Information Driven Design We also explore the potential of integrating chemical information of proteins into drug design. Briefly speaking, based on hydrogen bond acceptor-donor rules [21], we put the fragment with more hydrogen atoms into the pocket region with more oxygen, nitrogen, and fluorine atoms. The results in Table 3 show that the strategy does not achieve better performance when we increase the effect of chemical information on the output probability. The reason might be not considering detailed chemical information, like the bond length. We leave a more thoughtful method for combining the chemical information for future work.

Atom-based Pre-training We pre-train an atom-based SHAPE2MOL to analyze the contribution of our fragment-based generation style. Specifically, we keep most pre-training processes the same but replace our tokenizing method with the atom-based strategy (both mentioned in Section 1.2). As drug molecules mainly consist of carbon atoms, the atom-based SHAPE2MOL ignores input shapes and keeps outputting carbon atoms. The atom-based variant can not design any valid molecules, which indicates the fragment-based style is crucial for our pre-training process.

Invariance of SHAPE2MOL Following liGAN, when training SHAPE2MOL, we randomly rotate and translate the molecules to make our model have rotation and translation invariance ability. We compare the similarity of generated molecules based on different or the same molecular shapes (both are randomly rotated and translated). The similarity rises from 0.092 to 0.508, which shows that with the same molecular shape as input, the model produces similar molecules.

2.3 More Ablation Study of Sketching

Seed Shape Type We evaluate the influence of different seed shape in Table 4. We find that directly using the pocket as the shape may contribute to suboptimal result in providing highly active molecules. Through the sketching stage, we obtain more realistic and diverse molecular shapes. These shapes benefit the molecular activity but result in more complex structures and lower synthetic accessibility in generated molecules.

3 Discussion of Related Work

3.1 Shape-based Drug Design

Here we discuss the relationship between our method and existing shape-based drug design [22] approaches. Some previous work designed new drugs based on the shape of a known ligand. Traditional approaches work in a retrieval way – finding molecules whose shape is most similar to a known one [23, 24]. Modern deep learning models can decode a molecule from its shape [1, 25]. Such ligand-based generation can not generalize to unseen pockets. Luo et al. [2] directly generates molecules from the pocket shape, which is mostly close to our work. However, our model works in a fragment-based fashion, while theirs works in an atomic way. What makes our model especially different is that we utilize the power of the pre-training model to make pocket-based drug design more promising.

3.2 Tokenization and Linearization

Here we discuss how the tokenization and linearization procedures relate to fragment-based drug design (FBDD) [26, 27].

Fragment-Based Drug Design Briefly, there are two approaches in FBDD: growing the fragment synthetically to a proximal binding site or by linking two fragments together [27]. Our method can be classified as the previous type since the linearization generates a molecule in a one-by-one fashion.

Tokenization The procedure is carefully designed for deep generative models to avoid loops and preserve functionalities, which is in spirit to the principle of [3]. Some FBDD work, such as [28] works in a discriminative approach. Thus there are not many constraints when cutting the molecules. Podda et al. [29] uses the SMILES-fragment rather than a real molecule-fragment. Thus it can not utilize rich structured features.

Linearization The procedure aims at traversing (or generating) a structured object in a left-to-right approach [30], which is tractable and scalable. It is borrowed from the area of computational linguistics, more specifically, syntactic parsing and structured generation [31, 32]. For micro-molecules, linearization (such as SMILES [33]) has been adopted for several decades. There is a line of research work for SMILES-based generation [34, 35, 36]. Similarly, in the area of macro-molecule, Huang et al. [37] designs a linear structure for estimating the likelihood of the structure of RNA.

Comparison Compared with traditional linearized sequences such as SMILES [33], our method utilizes structural information to segment the molecules to preserve their functionality. Compared with topological generation based on a graph [3, 38, 39], our method is more scalable to big data since generating the variable graph topology is not friendly to large-batch training in neural networks.

4 More Cases

We show more cases from liGAN and DESERT in Figure 2. Similarly, liGAN struggles to produce realistic molecular structures, while DESERT generates better results.

References

[1] Tomohide Masuda, Matthew Ragoza, and David Ryan Koes. "Generating 3D Molecular Structures Conditional on a Receptor Binding Site with Deep Generative Models". In: *CoRR* abs/2010.14442 (2020). arXiv: 2010.14442. URL: https://arxiv.org/abs/2010.14442.

[2] Shitong Luo et al. "A 3D Generative Model for Structure-Based Drug Design". In: *Advances in Neural Information Processing Systems* 34 (2021).

[3] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. "Junction tree variational autoencoder for molecular graph generation". In: *International conference on machine learning*. PMLR. 2018, pp. 2323–2332.

[4] Xiao Qing Lewell et al. "Recap retrosynthetic combinatorial analysis procedure: a powerful new technique for identifying privileged molecular fragments with useful applications in combinatorial chemistry". In: *Journal of chemical information and computer sciences* 38.3 (1998), pp. 511–522.

[5] Jrg Degen et al. "On the Art of Compiling and Using'Drug-Like'Chemical Fragment Spaces". In: *ChemMedChem: Chemistry Enabling Drug Discovery* 3.10 (2008), pp. 1503–1507.

[6] Wen Tai et al. "exBERT: Extending pre-trained models with domain-specific vocabulary under constrained training resources". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. 2020, pp. 1433–1439.

[7] Yi Zhou et al. "On the continuity of rotation representations in neural networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5745–5753.

[8] Luca Falorsi et al. "Explorations in homeomorphic variational auto-encoding". In: *arXiv preprint arXiv:1807.04689* (2018).

[9] Wikipedia contributors. *Quaternion — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Quaternion&oldid=1112663286. [Online; accessed 4-October-2022]. 2022.

[10] Wikipedia contributors. *Euler angles — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Euler_angles&oldid=1112943781. [Online; accessed 4-October-2022]. 2022.

[11] Wikipedia contributors. *3D rotation group — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=3D_rotation_group&oldid=1101034704. [Online; accessed 4-October-2022]. 2022.

[12] John Jumper et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589.

[13] David Rogers and Mathew Hahn. "Extended-connectivity fingerprints". In: *Journal of chemical information and modeling* 50.5 (2010), pp. 742–754.

[14] Marcus Olivecrona et al. "Molecular de-novo design through deep reinforcement learning". In: *Journal of cheminformatics* 9.1 (2017), pp. 1–14.

[15] Oleg Ursu et al. "DrugCentral 2018: an update". In: *Nucleic acids research* 47.D1 (2019), pp. D963–D970.

[16] G Richard Bickerton et al. "Quantifying the chemical beauty of drugs". In: *Nature chemistry* 4.2 (2012), pp. 90–98.

[17] Peter Ertl and Ansgar Schuffenhauer. "Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions". In: *Journal of cheminformatics* 1.1 (2009), pp. 1–11.

[18] Oleg Trott and Arthur J Olson. "AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading". In: *Journal of computational chemistry* 31.2 (2010), pp. 455–461.

[19] Markus Freitag and Yaser Al-Onaizan. "Beam search strategies for neural machine translation". In: *arXiv preprint arXiv:1702.01806* (2017).

[20] Ari Holtzman et al. “The curious case of neural text degeneration”. In: *arXiv preprint arXiv:1904.09751* (2019).

[21] Roderick E Hubbard and Muhammad Kamran Haider. “Hydrogen bonds in proteins: role and strength”. In: *eLS* (2010).

[22] Rob LM Van Montfort and Paul Workman. “Structure-based drug design: aiming for a perfect fit”. In: *Essays in biochemistry* 61.5 (2017), pp. 431–437.

[23] Ashutosh Kumar and Kam YJ Zhang. “Advances in the development of shape similarity methods and their application in drug discovery”. In: *Frontiers in chemistry* 6 (2018), p. 315.

[24] Camila Cardoso Santos et al. “Drug screening using shape-based virtual screening and in vitro experimental models of cutaneous Leishmaniasis”. In: *Parasitology* 148.1 (2021), pp. 98–104.

[25] Miha Skalic et al. “Shape-based generative modeling for de novo drug design”. In: *Journal of chemical information and modeling* 59.3 (2019), pp. 1205–1214.

[26] Philine Kirsch et al. “Concepts and core principles of fragment-based drug design”. In: *Molecules* 24.23 (2019), p. 4309.

[27] Christopher W Murray and David C Rees. “The rise of fragment-based drug discovery”. In: *Nature chemistry* 1.3 (2009), pp. 187–192.

[28] Harrison Green, David R Koes, and Jacob D Durrant. “DeepFrag: a deep convolutional neural network for fragment-based lead optimization”. In: *Chemical Science* 12.23 (2021), pp. 8036–8047.

[29] Marco Podda, Davide Bacciu, and Alessio Micheli. “A deep generative model for fragment-based molecule generation”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 2240–2250.

[30] Yijia Liu et al. “Transition-based syntactic linearization”. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2015, pp. 113–122.

[31] Yoon Kim et al. “Unsupervised recurrent neural network grammars”. In: *arXiv preprint arXiv:1904.03746* (2019).

[32] Oriol Vinyals et al. “Grammar as a foreign language”. In: *Advances in neural information processing systems* 28 (2015).

[33] David Weininger. “SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules”. In: *Journal of chemical information and computer sciences* 28.1 (1988), pp. 31–36.

[34] Matt J Kusner, Brooks Paige, and Jos é Miguel Hern á ndez-Lobato. “Grammar variational autoencoder”. In: *International conference on machine learning*. PMLR. 2017, pp. 1945–1954.

[35] Hanjun Dai et al. “Syntax-directed variational autoencoder for structured data”. In: *arXiv preprint arXiv:1802.08786* (2018).

[36] Seokho Kang and Kyunghyun Cho. “Conditional molecular design with deep generative models”. In: *Journal of chemical information and modeling* 59.1 (2018), pp. 43–52.

[37] Liang Huang et al. “LinearFold: linear-time approximate RNA folding by 5’-to-3’dynamic programming and beam search”. In: *Bioinformatics* 35.14 (2019), pp. i295–i304.

[38] Binghong Chen et al. “Molecule Optimization by Explainable Evolution”. In: *International Conference on Learning Representations*. 2020.

[39] Yutong Xie et al. “Mars: Markov molecular sampling for multi-objective drug discovery”. In: *arXiv preprint arXiv:2103.10432* (2021).