DyMol: Dynamic Many-Objective Molecular Optimization with Objective Decomposition and Progressive Optimization

Dong-Hee Shin[†], **Young-Han Son**[†], **Deok-Joong Lee, Ji-Wung Han & Tae-Eui Kam**[*]
Department of Artificial Intelligence, Korea University
{dongheeshin,yhson135,deokjoong,danielhan,kamte}@korea.ac.kr

Abstract

Molecular discovery has received significant attention across various scientific fields by enabling the creation of novel chemical compounds. In recent years, the majority of studies have approached this process as a multi-objective optimization problem. Despite notable advancements, most methods optimize only up to four molecular objectives and are mainly designed for scenarios with a predetermined number of objectives. However, in real-world applications, the number of molecular objectives can be more than four (many-objective) and additional objectives may be introduced over time (dynamic-objective). To fill this gap, we propose DyMol, the first method designed to tackle the dynamic many-objective molecular optimization problem by utilizing a novel divide-and-conquer approach combined with a decomposition strategy. We validate the superior performance of our method using the practical molecular optimization (PMO) benchmark.

1 Introduction

Molecular discovery is foundational to progress in a variety of scientific fields, ranging from the development of new pharmaceuticals to the creation of innovative materials (Bilodeau et al., 2022). At its core, molecular discovery is a complex process that seeks to identify molecules with specific, desirable properties. In essence, this process is fundamentally a constrained multi-objective optimization problem, where the objectives are to simultaneously maximize or minimize certain attributes of molecules (Fromer & Coley, 2023). Unlike single-objective optimization, the multi-objective optimization problem introduces distinct challenges that arise from the necessity to balance multiple and often conflicting objectives (Marler & Arora, 2004). Therefore, it becomes infeasible to identify a single optimal solution that satisfies all objectives. Instead, the focus shifts to finding Pareto optimal solution sets that represent various trade-offs among these objectives (Gunantara, 2018).

To tackle the multi-objective molecular optimization (MOMO) problem, much prior work has employed a range of generative models, including sampling-based methods (Fu et al., 2021; Xie et al., 2021), genetic algorithms (Jensen, 2019; Tripp et al., 2021), probabilistic models (Bengio et al., 2021), and reinforcement learning (Olivecrona et al., 2017; Jin et al., 2020). However, given the necessity of simultaneously optimizing multiple objectives, several studies have commonly adopted the scalarization method, which transforms multiple objectives into a single objective function by aggregating them using weighted sums or other combining strategies (Gunantara, 2018). Alternatively, the Bayesian optimization (BO) methods have been employed to address multiple objectives concurrently by leveraging acquisition functions to navigate the optimization landscape without needing to quantify the relative weights of each objective (Fromer & Coley, 2023). While prior frameworks have shown effectiveness, many of these methods have primarily focused on optimizing a limited number of objectives, typically up to four, and are often designed for a fixed number of objectives.

In real-world drug discovery, the significance of dynamic many-objective molecular optimization setting becomes evident (Luukkonen et al., 2023). From a many-objective perspective, the drug development process is complex and multifaceted, typically requiring optimization of more than four objectives. In particular, a new drug must meet various criteria, including potency, safety, solubility,

[*] Corresponding Author, † Equal Contribution

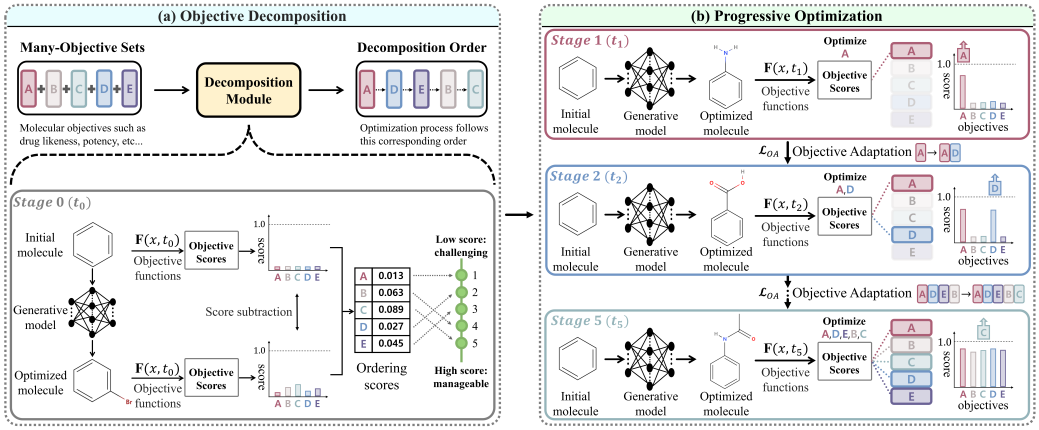


Figure 1: The overview of our DyMol method for dynamic many-objective molecular optimization problem, specifically exemplifies the scenario involving five molecular objectives. (a) Our method begins by decomposing complex many-objective sets into more manageable sub-problems facilitated by our decomposition module. (b) The optimization then progresses from a single objective, systematically incorporating additional objectives over time according to the decomposition order, thereby enabling progressive optimization.

bioavailability, and stability (Sadybekov & Katritch, 2023). From a dynamic-objective standpoint, the changing demands of the pharmaceutical market coupled with the emergence of new scientific insights during the drug development process may require the adjustment or the introduction of new optimization objectives. However, to our knowledge, no prior studies have tackled these issues.

The dynamic many-objective molecular optimization problem presents distinct challenges that set it apart from the typical MOMO problem. The many-objective aspect introduces an enormously large search space, which hinders efficient exploration and convergence to optimal Pareto solutions (Yuan et al., 2015). The dynamic-objective nature of this problem further exacerbates difficulties by continually altering the optimization landscape through the introduction of new objectives. To tackle these challenges, our approach diverges from most previous works, which typically aim to find Pareto optimal solutions for all objectives in a high-dimensional joint search space.

In this paper, we propose DyMol, which is a divide-and-conquer approach that decomposes the complex many-objective task into a series of manageable sub-problems, allowing us to address each of them sequentially. This progressive optimization scheme systematically unfolds the inherent complexities associated with the many-objective settings. Moreover, we incrementally introduce objectives over time in our method, allowing it to seamlessly adapt to the dynamic-objective settings with the incorporation of new objectives. To enhance our method's adaptability for new objectives, we have also developed an objective adaptation technique that detects changes in the optimization landscape and helps the model to identify effective Pareto solutions.

2 METHODS

2.1 OBJECTIVE DECOMPOSITION

As shown in Figure 1, at time stage t_0, our method begins by decomposing complex many-objective sets into more manageable sub-problems. This decomposition process is facilitated by our decomposition module, which analyzes the complexities of each objective and automatically determines the order in which they should be prioritized during optimization. Specifically, the generative model initially optimizes all objectives jointly for a limited number of iterations. The model uses an initial molecule as a starting point and produces an optimized molecule. Once complete, both the optimized and initial molecules are evaluated by objective functions $\mathbf{F}(x, t_0)$, also referred to as oracle functions. From this evaluation, we calculate ordering scores by subtracting the objective scores of the initial molecules from the optimized molecules. These scores offer insights into the extent of improvement in objective scores accomplished by the model. A lower ordering score indicates that optimizing a specific objective is more challenging, implying the need for early prioritization.

2.2 PROGRESSIVE OPTIMIZATION

The main idea behind our method is to employ a divide-and-conquer approach to provide adaptability and efficiency when dealing with dynamic many-objective optimization. Consider a scenario where we have a total of five molecular objectives: objectives A through E, as shown in Figure 1. Following the decomposition order of A $\rightarrow$ D $\rightarrow$ E $\rightarrow$ B $\rightarrow$ C, our proposed model begins by solely optimizing objective A as follows:

$$\begin{aligned} \underset{x \in \mathcal{X}_1}{\text{Maximize}} \quad & \mathbf{F}(x, t_1) = \{f_{\mathrm{A}}(x)\}, \\ \text{subject to} \quad & g_j(x, t_1) \leq 0, \quad j = 1, 2, \ldots, k; \\ & h_j(x, t_1) = 0, \quad j = 1, 2, \ldots, l; \end{aligned} \tag{1}$$

where x denotes the molecular vector, $\mathbf{F}(x, t_1)$ represents the set of molecular objective functions at time stage $t_1 \in T$, and $f_{\mathrm{A}}(x)$ is the objective function A that maps the molecule x to a real number. Note that $\mathcal{X}_1$ represents the feasible set in the decision space specific to t_1, potentially different from the general decision space $\mathcal{X}$ due to the dynamic nature of the problem. The $g_j(x, t_1)$ and $h_j(x, t_1)$ represent inequality and equality constraints, respectively, derived from the physical, biological, or chemical requirements that a molecule must meet to be viable in a real-world environment.

When the model satisfies a certain score threshold related to objective A or reaches a predetermined number of iterations, it progresses to the next time stage t_2 and incrementally incorporates additional objective D. Consequently, the optimization problem is expanded to maximize:

$$\mathbf{F}(x, t_2) = \{f_{\mathrm{A}}(x), f_{\mathrm{D}}(x)\}. \tag{2}$$

However, the introduction of a new objective function $f_{\mathrm{D}}(\cdot)$ necessarily alters the optimization landscape by expanding the dimensions of the objective space. In this context, our objective adaptation technique plays a crucial role by enabling the model to adapt to this evolving optimization landscape. Specifically, it detects changes in the composition of objective scores, which provide learning feedback for model training and updates. For instance, at stage t_1, the objective scores are solely based on the value of objective A. However, at t_2, they evolve to encompass a composite value of both objectives A and D. The major role of the objective adaptation technique is to retrain the model using these updated objective scores, enabling the model to adjust to the evolving Pareto front $\mathbf{PF}(t)$, which is defined as the set of optimal trade-offs among objectives such as follows:

$$\mathbf{PF}(t) = \{\mathbf{f}(x^*) \mid x^* \in \mathcal{X}, \nexists x \in \mathcal{X} : x \succ x^* \text{ w.r.t. } \mathbf{F}(x, t)\}, \tag{3}$$

where x^* represents the Pareto optimal solution, $\mathbf{f}(x^*)$ denotes the vector of objective function values for this solution, and $\mathbf{F}(x, t)$ indicates the set of objective functions that evolves over time.

As time stages progress, the model systematically incorporates each new subsequent objective in line with the decomposition order and sequentially adjusts to the evolving Pareto front. Eventually, at the end of the time stage, the model can address the complete set of objectives. Thus, our method can be considered as a divide-and-conquer approach, as it strategically divides the complex optimization task into a series of simpler sub-problems, each focusing on a specific subset of the objectives. However, distinct from conventional divide-and-conquer methods that solve sub-problems independently and then combine their solutions, our approach is characterized by its sequential adaptation and refinement of solutions. As new objectives are introduced, the model dynamically adjusts its search process and integrates the incremental sub-problem solutions into a comprehensive solution that addresses all objectives. This adaptive nature of our method can make it effective in the dynamic-objective settings, where the optimization landscape progressively evolves over time.

2.3 PARETO SAMPLING AND OBJECTIVE ADAPTATION

Our proposed model employs likelihood P_θ to autoregressively generate molecule x_g, from initial molecule sequence x_0 up to a maximum length L as:

$$P_\theta(x_g = x_L) = \prod_{j=0}^{L} P_\theta(x_j | x_{j-1}, x_{j-2}, \ldots, x_0). \tag{4}$$

These generated molecules are evaluated by $\mathbf{F}(x, t)$ to obtain objective scores. To provide learning feedback, we compute reward scores $R(x, t)$ at each time stage t by taking the weighted sum of

objective scores from $\mathbf{F}(x,t)$. This is represented as $R(x,t) = \sum_{i=1}^{n} w_i(t) f_i(x)$, where n is the total number of objective functions and $w_i(t)$ is the relative weight for $f_i(x)$ at time stage t. To further enhance the training efficiency, we utilize experience replay $\mathcal{B}$ that stores previously optimized molecules with high reward scores. In contrast to traditional approaches that primarily emphasize score-convergence, we develop the Pareto sampling technique to also consider Pareto diversity.

As shown in Figure 2, we perform two types of sampling: convergence sampling, where we sample molecules x_c with high reward scores from $\mathcal{B}$ to promote score-convergence, and Pareto sampling, where we sample molecules x_p from the Pareto front to encourage Pareto diversity. Finally, we optimize the model parameters θ using the following loss function:

$$\mathcal{L}(\theta, t) = [-\log P_\theta(\mathbf{x}) + \log P_{\text{prior}}(\mathbf{x}) + R(\mathbf{x}, t)]^2 \tag{5}$$

where P_{prior} is the likelihood of a pre-trained model that imposes additional constraints based on the chemical grammar. It should be noted that $\mathbf{x}$ encompasses a set of molecules x_g, x_c, and x_p, represented as $\mathbf{x} = \{x_g, x_c, x_p\}$.

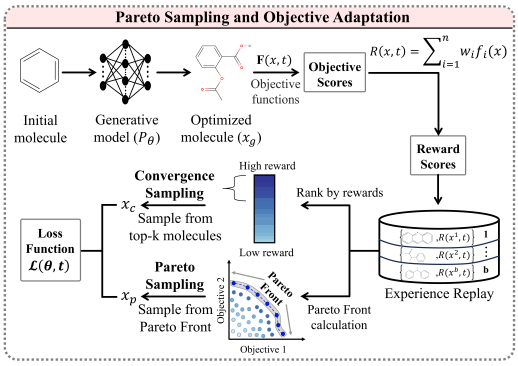


Figure 2: Molecules and their reward scores are stored in experience replay, which serves two major roles: ranking molecules by rewards for convergence sampling, and extracting the Pareto front for Pareto sampling. These sampled molecules are then integrated into the loss function for training the generative model.

As time stages advance $t \rightarrow t+1$, the introduction of new objective changes the composition of objective scores and the reward scores. Although the generative model is initially unaware of these changes, we introduce the objective adaptation technique to update θ. This involves retraining the model using updated reward scores to account for the impact of new objectives. The objective adaptation loss can be expressed as:

$$\mathcal{L}_{\text{OA}}(\theta, t) = [-\log P_\theta(x_b) + \log P_{\text{prior}}(x_b) + R(x_b, t+1)]^2, \tag{6}$$

where x_b denotes all molecules from $\mathcal{B}$. Note that we employ REINVENT (Olivecrona et al., 2017) as our backbone generative model due to its superior performance.

3 Experiments

3.1 Experimental Setup & Competing Methods

We evaluated the performance of our proposed method using the practical molecular optimization (PMO) benchmark (Gao et al., 2022). In this setup, oracle call budgets are strictly limited to 10,000 evaluations to reflect the real-world constraints of molecular discovery. For the oracle functions in our experiments, we adopted the most commonly used molecular objective functions in previous MOMO studies (Jin et al., 2020; Xie et al., 2021). These include biological objectives GSK3β and JNK3, which represent inhibition scores against two target proteins related to Alzheimer's disease, as well as non-biological objectives like QED and SA that quantify drug-likeness and synthesizability, respectively. To extend our approach to many-objective settings, we included further objectives such as DRD2, associated with dopamine receptor binding affinity, as well as Osimertinib MPO objective for discovering new therapeutics that optimize existing drugs with multiple desirable attributes.

We compared the performance of our method against a range of competing methods, including Random ZINC (Sterling & Irwin, 2015), SMILES-VAE (Gómez-Bombarelli et al., 2018), MIMOSA (Fu et al., 2021), GFlowNet (Bengio et al., 2021), and GraphGA (Jensen, 2019). Additionally, we evaluated against BO methods such as GPBO (Tripp et al., 2021), LaMBO (Stanton et al., 2022), and HN-GFN (Zhu et al., 2023); well-known many-objective optimization algorithms like MOEA/D (Zhang & Li, 2007) and NSGA-III (Verhellen, 2022); and RL-based methods, including REINVENT (Olivecrona et al., 2017), REINVENT BO (Tripp et al., 2021), and AugMem (Guo & Schwaller, 2023). Note that REINVENT was acknowledged as the best-performing algorithm for molecular optimization, as evidenced by the original PMO benchmark results (Gao et al., 2022).

Table 1: Comparative performance in scenarios with Four objectives (GSK3β+JNK3+QED+SA), Five objectives (GSK3β+JNK3+QED+SA+DRD2), and Six objectives (GSK3β+JNK3+QED+SA+DRD2+Osimertinib MPO) using 10 different seeds. The evaluation metrics used are the hypervolume (HV) and R2 indicators.

	Four objectives		Five objectives		Six objectives	
Method	HV(↑)	R2(↓)	HV(↑)	R2(↓)	HV(↑)	R2(↓)
Random ZINC	0.065±0.013	4.809±0.243	0.005±0.003	8.867±0.202	0.001±0.000	14.456±0.344
SMILES-VAE	0.073±0.023	4.839±0.435	0.004±0.001	9.226±0.512	0.001±0.000	15.109±0.548
GFlowNet	0.063±0.011	4.952±0.196	0.011±0.004	9.011±0.314	0.004±0.001	14.462±0.422
MIMOSA	0.082±0.028	4.905±0.410	0.016±0.012	9.195±0.761	0.005±0.005	15.454±1.701
LaMBO	0.123±0.006	4.496±0.148	0.009±0.001	9.302±0.159	Out-of-memory	Out-of-memory
HN-GFN	0.120±0.000	4.013±0.060	0.004±9.861	9.861±0.096	0.002±0.000	15.349±0.102
MOEA/D	0.176±0.123	4.615±1.193	0.094±0.052	8.105±1.561	0.025±0.018	15.054±2.028
NSGA-III	0.234±0.107	3.477±0.837	0.071±0.047	7.130±1.349	0.016±0.020	14.366±2.393
GraphGA	0.254±0.069	3.379±0.666	0.100±0.057	7.676±1.312	0.051±0.030	11.814±1.787
GPBO	0.275±0.091	3.311±0.757	0.091±0.031	7.670±0.761	0.026±0.024	12.840±1.811
REINVENT BO	0.309±0.021	2.795±0.103	0.071±0.014	7.537±0.620	0.033±0.019	10.929±1.019
REINVENT	0.338±0.030	2.770±0.116	0.099±0.054	7.578±1.187	0.062±0.028	10.032±0.922
AugMem	0.395±0.038	2.496±0.192	0.090±0.043	8.011±0.955	0.071±0.072	12.472±3.758
DyMol (Ours)	**0.422±0.023**	**2.297±0.095**	**0.247±0.087**	**4.943±0.990**	**0.143±0.056**	**8.842±1.632**

3.2 Experimental Results

The performances of our proposed method and the competing methods were assessed by two evaluation metrics: the hypervolume indicator (HV) (Zitzler et al., 2003) and the R2 indicator (R2) (Brockhoff et al., 2012). The HV measures the volume of the objective space dominated by the Pareto front, while the R2 evaluates the quality of a solution set based on user-defined reference points. A higher HV value indicates a better solution set, while a lower R2 value is more desirable.

As shown in Table 1, our method outperforms all competing methods across all scenarios. Notably, in scenarios with Five and Six objectives, our method demonstrates a substantial performance improvement. This highlights the effectiveness of our divide-and-conquer approach that successfully handles the inherent complexity of many-objective problems by decomposing them into manageable sub-problems. However, other methods struggle with exponential increases in complexity.

To assess our method in dynamic-objective scenarios, we propose a novel experimental setup where a model has initially been fully optimized for a set of Five objectives. Subsequently, a new, sixth objective (Osimertinib MPO) is introduced, requiring additional optimization. Instead of re-optimizing all objectives from scratch, we implement a fine-tuning approach that leverages the model already optimized for the initial Five objectives, and further optimizing the new objective. As depicted in Figure 3, our method effectively reaches the baseline performance of the Six objectives within fine-tuning 2000 oracle calls and continues to improve beyond that. This achievement can be attributed to our objective adaptation technique and the incremental nature of adding objectives within our method.

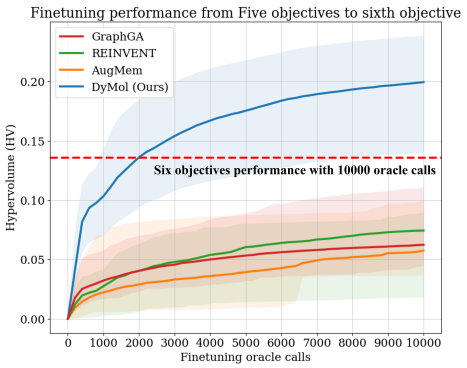


Figure 3: Finetuning performance of the top 4 methods in dynamic-objective scenarios.

4 Conclusion

In this work, we propose DyMol as a novel and first method to address the dynamic many-objective molecular optimization problem by leveraging the divide-and-conquer approach. DyMol decomposes complex many-objective sets into manageable sub-problems for progressive optimization. Our results demonstrate that DyMol outperforms competing methods in both many-objective and dynamic-objective scenarios. Future work can include extending research to material science.

5 ACKNOWLEDGEMENT

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant (No. 2019-0-00079, Artificial Intelligence Graduate School Program(Korea University), No. 2022- 0-00871, Development of AI Autonomy and Knowledge Enhancement for AI Agent Collaboration), and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS202300212498).

REFERENCES

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. *Advances in Neural Information Processing Systems*, 34:27381–27394, 2021.

Camille Bilodeau, Wengong Jin, Tommi Jaakkola, Regina Barzilay, and Klavs F Jensen. Generative models for molecular discovery: Recent advances and challenges. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 12(5):e1608, 2022.

Dimo Brockhoff, Tobias Wagner, and Heike Trautmann. On the properties of the R2 indicator. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, pp. 465–472, 2012.

Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *IEEE transactions on evolutionary computation*, 18(4):577–601, 2013.

Jinliang Ding, Cuie Yang, Qiong Xiao, Tianyou Chai, and Yaochu Jin. Dynamic evolutionary multiobjective optimization for raw ore allocation in mineral processing. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 3(1):36–48, 2018.

Jenna C Fromer and Connor W Coley. Computer-aided multi-objective optimization in small molecule discovery. *Patterns*, 4(2), 2023.

Tianfan Fu, Cao Xiao, Xinhao Li, Lucas M Glass, and Jimeng Sun. Mimosa: Multi-constraint molecule sampling for molecule optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 125–133, 2021.

Wenhao Gao, Tianfan Fu, Jimeng Sun, and Connor Coley. Sample efficiency matters: a benchmark for practical molecular optimization. *Advances in Neural Information Processing Systems*, 35: 21342–21357, 2022.

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.

Nyoman Gunantara. A review of multi-objective optimization: Methods and its applications. *Cogent Engineering*, 5(1):1502242, 2018.

Jeff Guo and Philippe Schwaller. Augmented memory: Capitalizing on experience replay to accelerate de novo molecular design. *arXiv preprint arXiv:2305.16160*, 2023.

Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *arXiv preprint arXiv:2102.09548*, 2021.

Jan H Jensen. A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. *Chemical science*, 10(12):3567–3572, 2019.

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

Marco Laumanns and Jiri Ocenasek. Bayesian optimization algorithms for multi-objective optimization. In *International Conference on Parallel Problem Solving from Nature*, pp. 298–307. Springer, 2002.

Xi Lin, Zhiyuan Yang, Xiaoyuan Zhang, and Qingfu Zhang. Pareto set learning for expensive multi-objective optimization. *Advances in Neural Information Processing Systems*, 35:19231–19247, 2022.

Ruochen Liu, Ping Yang, and Jiangdi Liu. A dynamic multi-objective optimization evolutionary algorithm for complex environmental changes. *Knowledge-Based Systems*, 216:106612, 2021.

Sohvi Luukkonen, Helle W van den Maagdenberg, Michael TM Emmerich, and Gerard JP van Westen. Artificial intelligence in multi-objective drug design. *Current Opinion in Structural Biology*, 79:102537, 2023.

R Timothy Marler and Jasbir S Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26:369–395, 2004.

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

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.

Zhen Quan, Yan Wang, and Zhicheng Ji. Multi-objective optimization scheduling for manufacturing process based on virtual workflow models. *Applied Soft Computing*, 122:108786, 2022.

Anastasiia V Sadybekov and Vsevolod Katritch. Computational approaches streamlining drug discovery. *Nature*, 616(7958):673–685, 2023.

Samuel Stanton, Wesley Maddox, Nate Gruver, Phillip Maffettone, Emily Delaney, Peyton Greenside, and Andrew Gordon Wilson. Accelerating bayesian optimization for biological sequence design with denoising autoencoders. In *International Conference on Machine Learning*, pp. 20459–20478. PMLR, 2022.

Teague Sterling and John J Irwin. Zinc 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55(11):2324–2337, 2015.

Austin Tripp, Gregor NC Simm, and José Miguel Hernández-Lobato. A fresh look at de novo molecular design benchmarks. In *NeurIPS 2021 AI for Science Workshop*, 2021.

Jonas Verhellen. Graph-based molecular pareto optimisation. *Chemical Science*, 13(25):7526–7535, 2022.

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.

Yuan Yuan, Hua Xu, Bo Wang, Bo Zhang, and Xin Yao. Balancing convergence and diversity in decomposition-based many-objective optimizers. *IEEE Transactions on Evolutionary Computation*, 20(2):180–198, 2015.

Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731, 2007.

Yiheng Zhu, Jialu Wu, Chaowen Hu, Jiahuan Yan, Chang-Yu Hsieh, Tingjun Hou, and Jian Wu. Sample-efficient multi-objective molecular optimization with GFlownets. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=uoG1fLIK2s`.

Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M Fonseca, and Viviane Grunert Da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation*, 7(2):117–132, 2003.

APPENDIX

A PSEUDO-CODE

This section provides the DyMol pseudo-code for the entire training process.

Algorithm 1: Decomposition Module

Input: Generative model P_θ, Prior model P_{prior}, Decomposition oracle calls N_{order}, Number of batch size N_{batch}, Objective function f, Number of objectives n, Initial joint objective sets F_{joint}

Initialize Generative model P_θ = P_{prior}
Experience replay buffer $\mathcal{B}$= $\{\}$
Length of replay buffer $N = 0$
Sample batch of initial molecules $x_{init} \sim P_\theta$
while $N < N_{order}$ **do**
  Sample batch of molecule $x_g \sim P_\theta$
  Calculate the objective scores $F_{joint}(x_g, 0)$
  Compute the reward scores $R(x_g, 0) = \sum_{i=1}^{n} \frac{f_i(x_g)}{n}$
  Update replay buffer $\mathcal{B} \cup (x_g, R(x_g, 0))$
  Sample x_c from TopK high scoring molecules from buffer $x_c \sim TopK(\mathcal{B})$
  $\mathbf{x} = x_g \cup x_c$
  Update model $\mathcal{L}(\theta, 0) = [-\log P_\theta(\mathbf{x}) + \log P_{\text{prior}}(\mathbf{x}) + R(\mathbf{x}, 0)]^2$
  $N = N + N_{batch}$
end
Sample batch of prototype molecules $x_{proto} \sim P_\theta$
$Ordering\ scores = \sum \frac{F_{joint}(x_{proto}) - F_{joint}(x_{init})}{N_{batch}}$
$Ordering = Argsort(Ordering\ scores)$
Return $Ordering$

Algorithm 2: Progressive Optimization

Input: Generative model P_θ, Prior model P_{prior}, Experience replay Buffer $\mathcal{B}$, Score threshold s_{thre}, Objective function f, Number of objectives n, Objective order $Ordering$

Objective $Ordering \rightarrow \{f_1, f_2, ..., f_n\}$
Initialize objective function sets $F = \{\}$
for *t=1 to* $t = n$ **do**
  Objective set update $F \cup f_t$
  Relative reward weight $\{w_1, w_2, ..., w_t\} = \frac{1}{t}$
  $w_t = w_t \times 1.5$
  while $f_t(x_g) < s_{thre}$ **do**
    Sample batch of molecule $x_g \sim P_\theta$
    Calculate the objective scores $F(x_g, t)$
    Calculate Pareto front $\mathbf{PF}$
    Compute the reward scores $R(x_g, t) = \sum_{i=1}^{t} w_i f_i(x_g)$
    Update replay buffer $\mathcal{B} \cup (x_g, R(x_g, t))$
    Sample TopK high scoring molecules from buffer $x_c \sim TopK(\mathcal{B})$
    Sample molecules from Pareto front $x_p \sim \mathbf{PF}$
    $\mathbf{x} = x_g \cup x_c \cup x_p$
    Update model $\mathcal{L}(\theta, t) = [-\log P_\theta(\mathbf{x}) + \log P_{\text{prior}}(\mathbf{x}) + R(\mathbf{x}, t)]^2$
  end
  Objective adaptation with entire TopK buffer $x_b = TopK(\mathcal{B})$
  $\mathcal{L}_{\text{OA}}(\theta, t) = [-\log P_\theta(x_b) + \log P_{\text{prior}}(x_b) + R(x_b, t+1)]^2$
end

B Related Work

B.1 Generative Models for Molecular Discovery

In recent years, there has been a growing interest in the use of various generative models for molecular discovery. Generative models employed in molecular discovery can be broadly classified into four categories: 1) sampling-based methods, 2) genetic algorithms, 3) reinforcement learning (RL), and 4) probabilistic models. The sampling-based methods (Xie et al., 2021; Fu et al., 2021) focus on sampling from a distribution of possible molecules with desirable properties. The genetic algorithms (Jensen, 2019; Tripp et al., 2021) employ a population-based approach that evolves molecules through iterative selection, crossover, and mutation guided by a fitness function. The RL-based methods (Olivecrona et al., 2017; Jin et al., 2020) involve an agent that interacts with an environment to generate molecular structures. In this paradigm, the agent receives rewards for producing chemical compounds with desired properties, thereby gradually refining its strategy towards achieving optimal molecular designs. The probabilistic models, GFlowNets (Bengio et al., 2021), generate molecular structures by identifying high-potential regions using probability distributions from data.

B.2 Multi-Objective Molecular Optimization

In the context of the MOMO problem, the challenge lies in simultaneously optimizing multiple molecular objectives, which often conflict with each other (Luukkonen et al., 2023). To address this, two prominent multi-objective optimization techniques have been widely adopted: scalarization and Bayesian optimization (BO). For instance, in the case of scalarization, MIMOSA (Fu et al., 2021) has employed straightforward linear scalarization techniques to handle the MOMO problem. These techniques aim to aggregate multiple objectives into a single objective function by using weighted sums or Tchebycheff methods (Lin et al., 2022). On the other hand, BO offers a black box optimization approach that has been integrated into various molecular generative models to enhance sample efficiency (Laumanns & Ocenasek, 2002). In particular, GPBO (Tripp et al., 2021) exemplifies the integration of BO within the framework of GraphGA (Jensen, 2019) as the backbone model. Similarly, LaMBO (Stanton et al., 2022) leverages BO on top of denoising autoencoders to address multi-objective sequence design problems. Recently, HN-GFN (Zhu et al., 2023) proposes a multi-objective BO algorithm that leverages hypernetwork-based GFlowNets.

B.3 Dynamic Many-Objective Optimization

Dynamic many-objective optimization, while yet to be widely explored in molecular optimization, has found application in diverse fields such as manufacturing (Quan et al., 2022), environmental management (Liu et al., 2021), and mineral processing (Ding et al., 2018). Existing approaches in these domains have predominantly employed decomposition-based MOEA/D (Zhang & Li, 2007) and non-dominated sorting NSGA-III (Deb & Jain, 2013) frameworks due to their effectiveness in navigating high-dimensional search space.

C Experimental Details

C.1 Competing Methods

Here, we provide a detailed overview of the competing methods, outlining their key principles, methodologies, and how they stand in comparison to our proposed method.

- Random ZINC (Sterling & Irwin, 2015) functions as a baseline, employing a straightforward approach of randomly sampling molecules from the ZINC dataset. It demonstrates the basic level of effectiveness that can be achieved by merely sampling from an existing dataset, without the application of any advanced optimization or generation strategies.
- SMILES-VAE (Gómez-Bombarelli et al., 2018) is a sampling-based method using a variational autoencoder model to generate molecules. These molecules are represented as SMILES strings, a textual format that encodes molecular structures using concise strings of characters to denote atoms and their connections.

- MIMOSA (Fu et al., 2021) is a sampling-based method that utilizes Markov Chain Monte Carlo for efficient sampling from a targeted molecular distribution. It begins with an input molecule and progressively samples subsequent molecules from the specified distribution.
- GFlowNets (Bengio et al., 2021) represents one of the unique classes of probabilistic models that integrate the principles from both RL and sampling-based methods. Specifically, this model is designed to sample more frequently from areas of higher rewards by leveraging a probabilistic approach in its training process.
- GraphGA (Jensen, 2019) is a genetic algorithms-based method that evolves molecules in a population through iterative selection, crossover, and mutation, guided by a fitness function. It leverages chemical domain knowledge to design molecular mutation and crossover rules that efficiently explore the molecular space.
- GPBO (Tripp et al., 2021) employs the Bayesian optimization (BO) framework to tackle the multi-objective molecular optimization problem. It utilizes GraphGA as its backbone model and aims to enhance sample efficiency by incorporating BO within its method.
- LaMBO (Stanton et al., 2022) leverages the BO framework on top of denoising autoencoders to address multi-objective sequence design problems. It employs a discriminative multi-task Gaussian process to improve sample efficiency by predicting objective values.
- HN-GFN (Zhu et al., 2023) is one of the most recent methods that tackle the multi-objective molecular optimization problem. It introduces a multi-objective BO algorithm with GFlowNets as its core model for sampling diverse molecule candidates. Additionally, it employs a hindsight-like off-policy strategy with the main purpose of sharing the memory of high-performing molecules, thereby accelerating the learning process.
- MOEA/D (Zhang & Li, 2007) is one of the most popular algorithms for handling dynamic many-objective optimization problems. It decomposes a multi-objective problem into simpler single-objective subproblems using scalarization functions. Each subproblem is then optimized concurrently using evolutionary algorithms. Our proposed method aligns with decomposition-based algorithms in its approach. However, unlike MOEA/D-based algorithms, our method is specifically designed for molecular optimization and does not solely rely on the population-based nature of evolutionary algorithms. Additionally, our method proposes a unique incremental objective addition strategy, starting with a single objective and systematically introducing additional objectives over time. Furthermore, we have developed an objective adaptation technique to aid our model in adapting to newly introduced objectives.
- NSGA-III (Deb & Jain, 2013) is another widely popular algorithm for addressing dynamic many-objective optimization problems. It categorizes solutions into trade-off fronts based on their dominance relationships. Molecular NSGA-III (Verhellen, 2022) provides comprehensive results for small molecule drug generation by utilizing NSGA-based algorithms.
- REINVENT (Olivecrona et al., 2017) is a reinforcement learning (RL)-based method that utilizes an agent interacting with an environment to generate molecules. It utilizes an autoregressive approach to sequentially generate molecules represented as SMILES strings, with each step in the generation process building upon the previously generated elements. The generation process is further guided by a pre-trained prior model that enforces chemical grammar constraints, ensuring the chemical validity of the generated molecular structures. REINVENT has been recognized as the best-performing algorithm for the molecular optimization problem, as evidenced by the PMO benchmark (Gao et al., 2022). Due to its superior performance, many other methods have adopted REINVENT as their backbone model. In alignment with this trend, we have also employed REINVENT as our backbone model to leverage its proven capabilities in generating chemically valid molecules.
- REINVENT BO (Tripp et al., 2021) is the RL-based method that incorporates the BO framework. In essence, REINVENT BO shares similarities with GPBO, but instead of using GraphGA as its backbone model, it employs REINVENT. This method can demonstrate the potential level of performance that can be achieved from integrating the RL-based method and the BO framework when addressing dynamic many-objective molecular optimization problems.
- AugMem (Guo & Schwaller, 2023) is another RL-based method that builds upon the REINVENT method. It enhances the performance of REINVENT by incorporating a data

augmentation technique and experience replay. The authors claim that their method has achieved a new state-of-the-art performance in the PMO benchmark. Hence, in our comparative analysis, we have primarily compared our method against AugMem. The results indicate that our method has successfully achieved better performance than AugMem, thereby demonstrating the effectiveness of our progressive optimization via the divide-and-conquer approach in addressing dynamic many-objective molecular optimization problems.

We reproduced SMILES-VAE, GFlowNet, MIMOSA, GraphGA, GPBO, REINVENT, and AugMem within PMO benchmark repository settings (Gao et al., 2022). We closely followed the recommended hyperparameter tuning strategy from the PMO benchmark repository, and we disabled the early stop strategy for the fair comparison of hypervolume. However, we observed that the default hyperparameter setting consistently yielded comparable to or similar to those obtained through hyperparameter tuning. In the case of REINVENT BO, the Bayesian optimization algorithm used in GPBO was additionally applied to the REINVENT model. For LaMBO and HN-GFN, we conducted experiments by replacing only the objective function of these papers with the objective function used in the PMO benchmark (Huang et al., 2021). For NSGA-III and MOEA/D, we implemented these based on the repository by Jonas Verhellen (Verhellen, 2022).

C.2 Implementation Details for DyMol

We implemented the proposed DyMol using PyTorch (Paszke et al., 2019) and integrated it within the PMO benchmark. We did not change any hyperparameter settings of the baseline generative model REINVENT (Olivecrona et al., 2017) from the default PMO benchmark setting. For the decomposition module, we determined the ordering using N_{order} oracle calls during stage t_0. In the progressive optimization stage, we advanced to the next stage when either the average objective score in the generated batch surpassed the predefined threshold s_{thre} or the patience threshold of N_{thre} oracle calls in that stage was reached. Throughout each stage, we calculated the relative weights of the objectives using a weighted sum approach, with their averages representing the cumulative importance of each objective. However, when dealing with newly introduced objectives in each stage, we multiplied their weights by a factor of 1.5 within a predetermined time period. This adjustment was made to facilitate a rapid adaptation to the newly introduced objectives. During each iteration, the generative model produced B samples, from which we computed their objective scores using a dedicated objective function. Subsequently, we calculated the reward scores for these samples and stored both the generated molecules and their corresponding reward scores in the experience replay buffer $\mathcal{B}$. For Pareto Sampling (PS), we split sample acquisition equally between convergence sampling and Pareto sampling. During the transition from stage t to $t+1$, Objective Adaptation (OA) recalibrated the reward scores for the next stage's objectives using the top-k high-reward molecules from the current stage. This recalibration was then used to update the parameters of the generative model, providing learning feedback in the updated objective space.

Table 2: Detailed overview of the specific hyperparameter settings employed in our DyMol method.

Generative model (REINVENT from PMO benchmark)	
Batch size B	64
Embedding dimension	128
Hidden dimension	512
Number of layer	3
Sigma	500
Experience replay size	24
Learning rate	5e-04
Optimizer	Adam
Decomposition and Progressive Optimization	
Number of calls in Decomposition Module N_{order}	500
Score threshold per stage s_{thre}	0.35
Patience threshold per stage N_{thre}	2500
Convergence sampling	12
Pareto sampling	12
TopK in high reward Buffer $\mathcal{B}$	100

C.3 EMPIRICAL RUNNING TIME

We performed all experiments using an RTX 3090 GPU. In the case of Random ZINC, the running time solely reflects the time taken for objective function computation and hypervolume calculation, as it involves random sampling from the ZINC dataset. The computation times for Random ZINC with Four, Five, and Six objectives were 0.3, 0.6, and 1.0 hours, respectively. For further details on the empirical running times of each method with Four objectives, please refer to Table 3.

We empirically confirmed that methods in the REINVENT family, such as REINVENT, AugMem, and DyMol, as well as those in the genetic algorithm family like GraphGA and NSGA, not only performed better but also had significantly faster running times. For MOEA/D, although it can be considered as a genetic algorithm, we observed that it consumed a considerable amount of time due to neighborhood calculation. In addition, BO methods such as GPBO, LaMBO, and HN-GFN exhibited high computational costs. Despite employing a batch size of 20 in LaMBO, we encountered out-of-memory issues, and the training of the surrogate model, such as the Gaussian Process (GP), required extensive time and GPU resources. Interestingly, although HN-GFN displayed considerable performance improvements compared to the GFlowNet, the experience replay mechanism appeared to yield greater benefits than the BO-based proxy oracle.

Table 3: Average empirical running times for each method under Four objectives (GSK3β+JNK3+QED+SA) optimization scenario.

Method	Running time (hr)
RandomZINC	0.344
SMILES-VAE	2.086
GFlowNet	0.856
MIMOSA	0.589
LaMBO	66.490
HN-GFN	87.904
MOEA/D	14.38
NSGA-III	0.413
GraphGA	0.430
GPBO	0.908
REINVENT BO	17.434
REINVENT	0.471
AugMem	0.765
DyMol (Ours)	0.494

D ANALYSIS OF MANY-OBJECTIVE SCENARIOS

In this section, we present the ablation study supporting the main results of our main manuscript, as detailed in Table 4. Then, we analyze the relationship between score convergence and Pareto diversity, as demonstrated in Table 5 and Table 6. We also present results where objective orders were assigned arbitrarily in Table 7, as opposed to using the ordering scores derived from the decomposition module. Furthermore, we provide the results of many-objective optimization with various combinations, including the additional objective of Fexofenadine MPO, in Table 8 and Table 9.

D.1 MAIN ABLATION STUDY

As shown in Table 4, we have conducted an ablation study to investigate the impact of key techniques on the performance of our method: Pareto Sampling (PS), Divide-and-Conquer (DC), and Objective Adaptation (OA). We observed that each of these techniques significantly contributes to improved performance. DC primarily focuses on improving convergence, bringing solutions closer to the optimal Pareto front values. PS enhances performance through emphasis on Pareto diversity. Remarkably, OA leads to substantial performance gains, especially in Five-objective scenarios, highlighting its capability to adapt effectively to newly introduced objectives.

Table 4: Ablation study for each technique: Pareto Sampling (PS), Divide-and-Conquer (DC), and Objective Adaptation (OA).

Ablation			Four objectives		Five objectives		Six objectives	
PS	DC	OA	HV(↑)	R2(↓)	HV(↑)	R2(↓)	HV(↑)	R2(↓)
-	-	-	0.338	2.770	0.099	7.578	0.062	10.032
✔	-	-	0.379	2.501	0.103	7.018	0.073	10.033
-	✔	-	0.363	2.692	0.150	6.276	0.101	10.408
-	✔	✔	0.373	2.621	0.209	5.480	0.105	10.269
✔	✔	-	0.412	2.321	0.182	5.488	0.122	9.439
✔	✔	✔	**0.422**	**2.297**	**0.247**	**4.943**	**0.143**	**8.842**

D.2 Score Convergence and Pareto Diversity

In the main ablation study presented in Table 4, we observed a minimal performance gain when applying Pareto Sampling (PS), especially in the Five objectives scenario. We hypothesized that this resulted from focusing solely on Pareto diversity without ensuring a score convergence. Therefore, to test this hypothesis, we conducted an experiment presented in Table 5, where we implemented Pareto Sampling alone without considering score convergence. Notably, as the number of objectives increased, the extent of performance decline became more pronounced. The results showed a significant performance decrease for scenarios involving Five and Six objectives, with a slight decrease for Four objectives. This suggests that achieving score convergence in complex problems is challenging, and the absence of score convergence sampling has a more pronounced effect in such cases. The importance of prioritizing score convergence before Pareto diversity aligns with our divide-and-conquer (DC) strategy. Given that DC aims to improve the score convergence of joint objectives, PS is likely to be more effective when combined with DC. As indicated in Table 6, a notable performance improvement is evident when both DC and PS are employed together, rather than applying PS alone.

Table 5: The performance of Pareto sampling (PS) with, and without score convergence sampling (SC). This result indicates that Pareto diversity without score convergence leads to inferior performance.

Ablation		Four objectives		Five objectives		Six objectives	
SC	PS	HV(↑)	R2(↓)	HV(↑)	R2(↓)	HV(↑)	R2(↓)
✔	-	0.338	2.770	0.099	7.578	0.062	10.032
-	✔	0.335	2.660	0.053	7.912	0.021	11.669
✔	✔	0.379	2.501	0.103	7.018	0.073	10.033

Table 6: Ablation study on the combined use of Pareto Sampling (PS) with Divide-and-Conquer (DC). When PS was applied alongside DC, a significant improvement in performance was observed.

Ablation		Four objectives		Five objectives		Six objectives	
DC	PS	HV(↑)	R2(↓)	HV(↑)	R2(↓)	HV(↑)	R2(↓)
-	-	0.338	2.770	0.099	7.578	0.062	10.032
-	✔	0.379	2.501	0.103	7.018	0.073	10.033
✔	-	0.363	2.692	0.150	6.276	0.101	10.408
✔	✔	0.412	2.321	0.182	5.488	0.122	9.439

D.3 Ablation Study on the Ordering of Objectives

In our DC approach, the ordering of objectives is very crucial. As there is no pre-defined order for training, the sequence heavily depends on domain-specific knowledge. This becomes increasingly challenging in scenarios with many objectives, where the number of objectives reaches five or six, resulting in an exponential increase in the number of potential ordering combinations. As a result, the manual determination of objective order becomes even more challenging. To address and resolve these challenges, we have established a criterion for determining the order of objectives through our

Table 7: Ablation study of DyMol based on the ordering of each objective. The highest performing results are in **bold**, and the lowest performing ones are underlined. The experimental results showed that the optimization order of DRD2 is a crucial element for the overall performance. Note that 'Osi' denotes Osimertinib MPO.

Ordering of objectives						Metrics	
QED	SA	JNK3	GSK3β	DRD2	Osi	HV(↑)	R2(↓)
3	2	0	1	-	-	**0.416**	**2.300**
2	3	1	0	-	-	0.364	2.596
4	3	0	1	**2**	-	**0.242**	4.893
3	4	0	2	1	-	0.235	4.921
4	3	1	0	**2**	-	0.242	**4.870**
4	3	1	2	0	-	0.103	7.327
3	4	2	0	1	-	0.229	5.145
4	3	2	1	0	-	0.097	7.361
4	5	0	1	2	3	0.133	8.419
5	4	0	1	**3**	2	**0.150**	**8.116**
5	4	0	2	1	3	0.143	8.544
5	4	0	2	3	1	0.120	8.708
5	4	0	3	1	2	0.137	8.622
4	5	0	3	2	1	0.128	8.668
4	5	1	0	2	3	0.121	8.498
4	5	1	0	3	2	0.131	8.410
5	4	1	2	0	3	0.065	11.553
4	5	1	2	3	0	0.114	8.848
5	4	1	3	0	2	0.050	12.051
5	4	1	3	2	0	0.123	9.001
5	4	2	0	1	3	0.124	8.721
5	4	2	0	3	1	0.144	8.300
4	5	2	1	0	3	0.055	12.153
4	5	2	1	3	0	0.122	8.747
5	4	2	3	0	1	0.049	12.257
4	5	2	3	1	0	0.105	9.852
4	5	3	0	1	2	0.141	8.660
5	4	3	0	2	1	0.142	8.476
4	5	3	1	0	2	0.054	12.283
4	5	3	1	2	0	0.133	8.738
4	5	3	2	0	1	0.052	12.447
5	4	3	2	1	0	0.109	9.634

decomposition module. This enables the model to autonomously evaluate their significance and establish the appropriate order.

In this section, we present the results of experiments in which we replaced the ordering mechanism of the decomposition module with manual ordering, as outlined in Table 7. These experiments consider various combinations in many-objective optimization scenarios. To reduce the number of combinations, QED and SA are positioned last. This is attributed to the fact that QED and SA are known to be more manageable tasks, as they typically start training with high objective scores that surpass the score threshold. Please note that there is no universally applicable rule for assessing the importance of objectives. Therefore, while we cannot assert that our decomposition module outperforms all possible combinations in scenarios like Six objectives scenarios with a large number of potential orderings, it has demonstrated strong performance in many cases. This illustrates that our decomposition module can serve as a valuable tool for guiding the ordering of objectives strategically.

A notable observation from the ordering results is the performance variation associated with the position of DRD2 in the ordering sequence. In scenarios with both Five and Six objectives, orderings that placed DRD2 last yielded the best performance, whereas those training DRD2 first were least effective. This consistent trend of diminished performance when the DRD2 objective is prioritized first highlights a potential avenue for future research. This could involve a more in-depth exploration into the complexities and specific challenges associated with the DRD2 receptor.

D.4 Results of DyMol across Various Objective Combinations

In the main paper, DRD2 was used as the fifth objective and Osimertinib MPO as the sixth. In this section, we expand our study by including Fexofenadine MPO as an additional objective. We present the results of experiments involving various combinations of molecular objectives in many-objective scenarios. In Table 8 and Table 9, 'Osi' represents Osimertinib MPO, and 'Fexo' signifies Fexofenadine MPO. The base objectives set in each scenario consists of Four objectives: QED, SA, GSK3β, and JNK3. In scenarios with Five objectives, we add one additional objective to this base set, and in scenarios with Six objectives, two additional objectives are included. Across all scenarios, DyMol consistently outperforms the baseline REINVENT backbone model in terms of both HV and R2 metrics. These results demonstrate the adaptability and effectiveness of our method, showing that its performance is not confined to a specific set of molecular objectives. Instead, DyMol exhibits robustness and efficacy across a diverse range of many-objective scenarios, effectively handling various combinations of molecular objectives. Furthermore, the inclusion of an additional objective, 'Fexo' (Fexofenadine MPO), in our experiments further validates the versatility and robustness of our proposed method.

Table 8: Performance comparison of Five objectives many-objective optimization scenarios using 10 different seeds. In Five objectives scenario, one additional molecular objective was included.

Five Objectives			Method	Metrics	
DRD2	Osi	Fexo		HV(↑)	R2(↓)
✔	-	-	REINVENT	0.083±0.041	7.912±0.844
			DyMol (Ours)	0.247±0.087	4.943±0.990
-	✔	-	REINVENT	0.170±0.074	6.092±1.232
			DyMol (Ours)	0.248±0.050	5.137±0.621
-	-	✔	REINVENT	0.178±0.052	5.731±0.666
			DyMol (Ours)	0.231±0.059	5.137±0.621

Table 9: Performance comparison of Six objectives many-objective optimization scenarios using 10 different seeds. In Six objectives scenario, two additional molecular objectives were included.

Six Objectives			Method	Metrics	
DRD2	Osi	Fexo		HV(↑)	R2(↓)
✔	✔	-	REINVENT	0.062±0.028	9.880±0.798
			DyMol (Ours)	0.143±0.056	8.842±1.632
✔	-	✔	REINVENT	0.069±0.042	10.637±1.675
			DyMol (Ours)	0.125±0.060	9.545±1.338
-	✔	✔	REINVENT	0.118±0.034	9.686±1.134
			DyMol (Ours)	0.181±0.021	8.119±0.455

E Analysis of Hypervolume Improvement Curves

In this section, we present the hypervolume improvement curves for many-objective scenarios and further analyze the early stages of hypervolume improvement curves to investigate the optimization mechanism for each method.

E.1 Results of Additional Hypervolume Improvement Curves

As shown in Figure 4, we observed that our method consistently outperforms all other competing methods across various many-objective scenarios, including those with Four objectives, Five objectives, and Six objectives. Note that for the sake of simplicity and clarity in our comparative analysis, we chose to focus on the top 8 performing methods – MOEA/D, NSGA-III, GraphGA, GPBO, REINVENT BO, REINVENT, AugMem, and our method. Intriguingly, our method exhibits

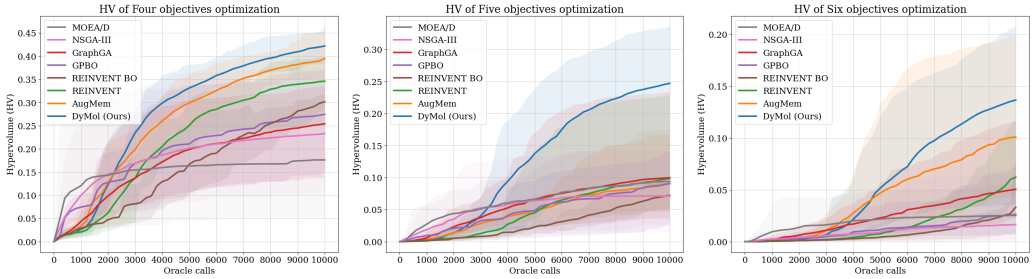


Figure 4: Average HV improvement curves for various many-objective scenarios.

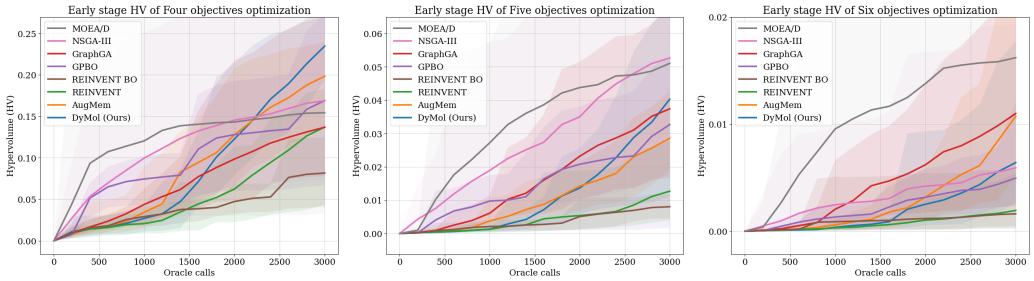


Figure 5: Early stage HV improvement curves for various many-objective scenarios.

a larger performance gap compared to other competing methods, particularly in scenarios with Five and Six objectives. This observation highlights the effectiveness of our divide-and-conquer approach in managing the complexities inherent in many-objective optimization problems, effectively dealing with exponential increases in complexity.

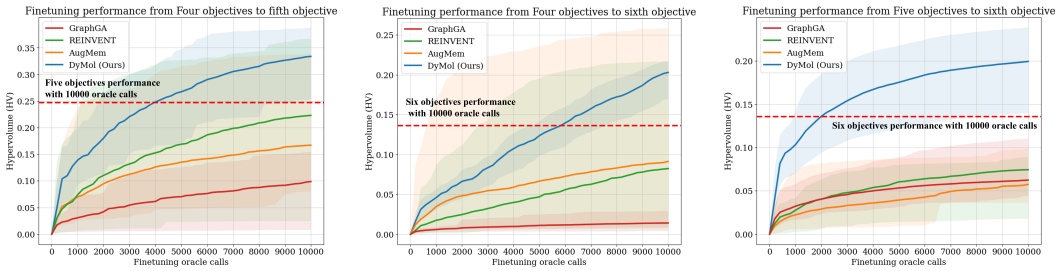
E.2 Early Stage Hypervolume Improvement Curves

Shifting our focus to the early stages of the hypervolume improvement curves, defined in this study as the initial oracle calls up to 3000, we conducted an in-depth analysis to gain deeper insights into the optimization mechanisms employed by each method. As shown in Figure 5, we observed that genetic algorithm-based methods such as MOEA/D, NSGA-III, GraphGA, and GPBO exhibit rapid improvement in the beginning, however, their performance plateau after. We think that this is because the population-based nature of these methods allows for a broad exploration of the solution space at the beginning. This wide exploration is effective in quickly identifying high-potential areas, leading to rapid improvements in performance. However, as the algorithm progresses, the population may start to converge, reducing the Pareto diversity. When this happens, it can limit the algorithm's capacity to explore new and promising regions of the search space, often resulting in a plateau in performance.

In contrast, RL-based algorithms like REINVENT, REINVENT BO, AugMem, and DyMol consistently improve hypervolume performance through continuous learning and adaptation via trial-and-error interactions with the environment. Furthermore, RL-based algorithms are inherently effective for the exploration process as they are designed to explore and learn from the environment.

F Analysis of Dynamic-Objective Scenarios

In this section, we explain more detailed information regarding the dynamic-objective scenarios and the primary motivation behind the design of this novel experimental setup. Furthermore, we provide additional experiments that explore dynamic-objective scenarios with different types of molecular objectives and various numbers of objectives.



(a) Initial optimization of Four objectives (GSK3β + JNK3 + QED + SA) with 10,000 oracle calls, subsequently followed by the introduction and fine-tuning of a fifth objective (DRD2).

(b) Initial optimization of Four objectives (GSK3β + JNK3 + QED + SA) with 10,000 oracle calls, subsequently followed by the introduction and fine-tuning of a fifth (DRD2) and a sixth objective (Osimertinib MPO).

(c) Initial optimization of Five objectives (GSK3β + JNK3 + QED + SA + DRD2) with 10,000 oracle calls, subsequently followed by the introduction and fine-tuning of a sixth objective (Osimertinib MPO).

Figure 6: Fine-tuning performance of the top 4 methods in various dynamic-objective scenarios, where new objectives are introduced.

F.1 Significance and Motivation

The concept of dynamic-objective scenarios can be significant in fields like drug discovery, where the optimization landscape continually evolves in response to various factors such as regulatory changes, scientific advancements, and emerging public health needs. In practice, molecular objectives in projects such as drug development are not static; they change and evolve as new scientific information and pharmaceutical requirements emerge. Hence, the main motivation for developing our novel experimental setup is its significance in real-world applicability and relevance in such scenarios. Despite the evident importance of dynamic-objective scenarios, to the best of our knowledge, prior studies have not approached or tackled these challenges.

F.2 Experimental Setups for Dynamic-Objective Scenarios

In our main paper, we proposed a novel experimental setup to evaluate the adaptability and efficiency of the optimization model in dynamic many-objective scenarios. The experiment began with the model already fully optimized for a specific set of molecular objectives, achieved through 10,000 oracle calls. After this initial optimization, we introduced a new objective to the optimization process. In our main paper, this new objective was selected as Osimertinib MPO. The introduction of this new objective simulates a common scenario in drug development, where additional criteria or requirements emerge during the optimization process. Instead of restarting the optimization process from scratch with the initial set of objectives and an additional objective, we opted for a fine-tuning approach, utilizing fine-tuning oracle calls. This approach involved adjusting the already optimized model (for the initial set of objectives) to accommodate the new objective. The rationale behind this strategy is to leverage the existing strengths and solutions of the model while efficiently integrating the new objective. We think that this approach is more resource-efficient and time-effective compared to re-optimizing all objectives from the beginning. The key focus of this experimental setup is to assess how well and how quickly the model can adapt to the introduction of a new objective. We measured the performance of the model in terms of its ability to maintain or improve the optimization levels of the initial objectives while effectively optimizing for the new objective. The results from these experiments can provide vital insights into the adaptability of each method in dynamic-objective scenarios. The model's ability to efficiently integrate and optimize new objectives without compromising existing optimization levels can serve as a key indicator of its robustness and practical applicability in real-world scenarios.

F.3 Additional Results for Dynamic-Objective Scenarios

Here, we present additional experimental results for various dynamic-objective scenarios. Figure 6-(a) illustrates one specific scenario where the initial set of Four objectives (GSK3β + JNK3 + QED

+ SA) is optimized with 10,000 oracle calls. Following this, a fifth objective (DRD2) is introduced and integrated through a fine-tuning approach by employing additional fine-tuning oracle calls. In the sub-figure, the red dashed line represents the baseline performance level established by jointly optimizing all Five objectives (GSK3β + JNK3 + QED + SA + DRD2) with 10,000 oracle calls. This baseline performance serves as a reference point to evaluate how efficiently and rapidly each method adapts to the addition of new molecular objectives.

It is important to note that in our comparative analysis, for the sake of simplicity and to present the performance of the top-performing methods succinctly in a single figure, we selectively compared and plotted the performance results of the top 4 performing methods – AugMem, REINVENT, GraphGA, and our method. As depicted in Figure 6-(a), our method demonstrates the capability to reach the baseline performance of Five objectives within just 4000 fine-tuning oracle calls and continues to improve thereafter. Conversely, the other methods do not achieve comparable performance within the same number of oracle calls.

In addition to the scenario presented in Figure 6-(a), we further explore different dynamic-objective scenarios as illustrated in Figures 6-(b) and 6-(c). Figure 6-(b) presents a scenario where the initial set of Four objectives (GSK3β + JNK3 + QED + SA) is optimized with 10,000 oracle calls. Subsequently, this is followed by the integration of a fifth objective (DRD2) and a sixth objective (Osimertinib MPO), each through a fine-tuning approach with additional fine-tuning oracle calls. Hence, in this sub-figure, the red dashed line indicates the baseline performance for optimizing all Six objectives (GSK3β + JNK3 + QED + SA + DRD2 + Osimertinib MPO) jointly with 10,000 oracle calls, accounting for the inclusion of the two additional objectives. Similarly, Figure 6-(c) depicts yet another dynamic-objective scenario. Here, the initial set of Five objectives (GSK3β + JNK3 + QED + SA + DRD2) is first optimized with 10,000 oracle calls, followed by the introduction and integration of the sixth objective (Osimertinib MPO) using the fine-tuning approach with additional oracle calls.

Consistent with the results from Figure 6-(a), both Figures 6-(b) and 6-(c) demonstrate that our method demonstrates the capability to efficiently reach and exceed the baseline performance within a limited number of fine-tuning oracle calls. This is in contrast to the other competing methods, which do not exhibit comparable performance within the same constraint of oracle calls. These results collectively highlight the adaptability and efficiency of our method in various dynamic-objective scenarios.

G MOLECULE EXAMPLES

In this section, we present visual examples of molecules generated by our proposed method that achieved high reward scores across various many-objective optimization problems, including those with Four, Five, and Six objectives. The corresponding objective scores for these molecules are indicated numerically beneath each molecule graph. For a more detailed visual presentation of these molecule examples, please refer to the following page.

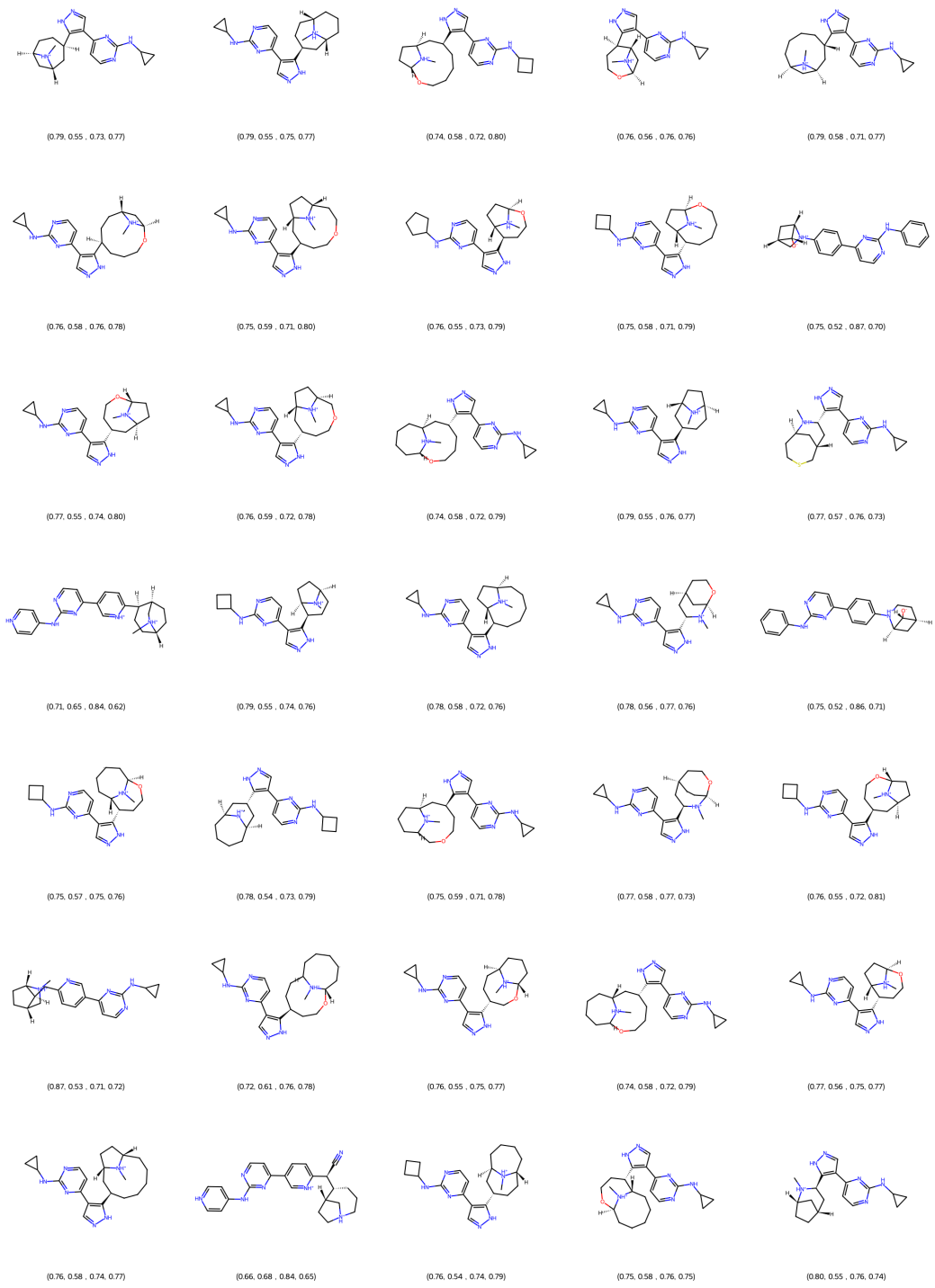


Figure 7: Sampled molecules with high reward scores in Four objectives (QED + SA + GSK3β + JNK3) optimization scenario.

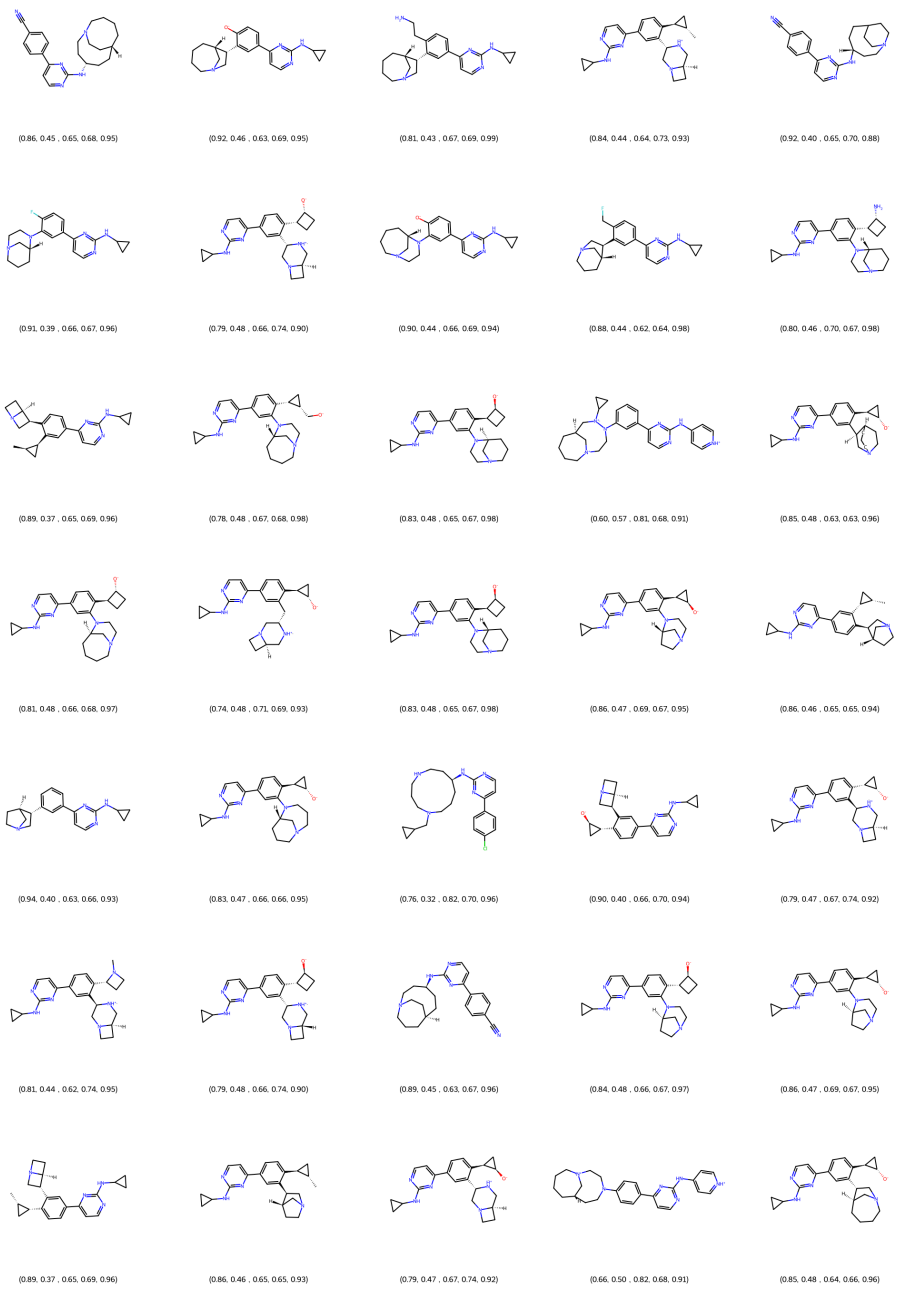


Figure 8: Sampled molecules with high reward scores in Five objectives (QED + SA + GSK3β + JNK3 + DRD2) optimization scenario.

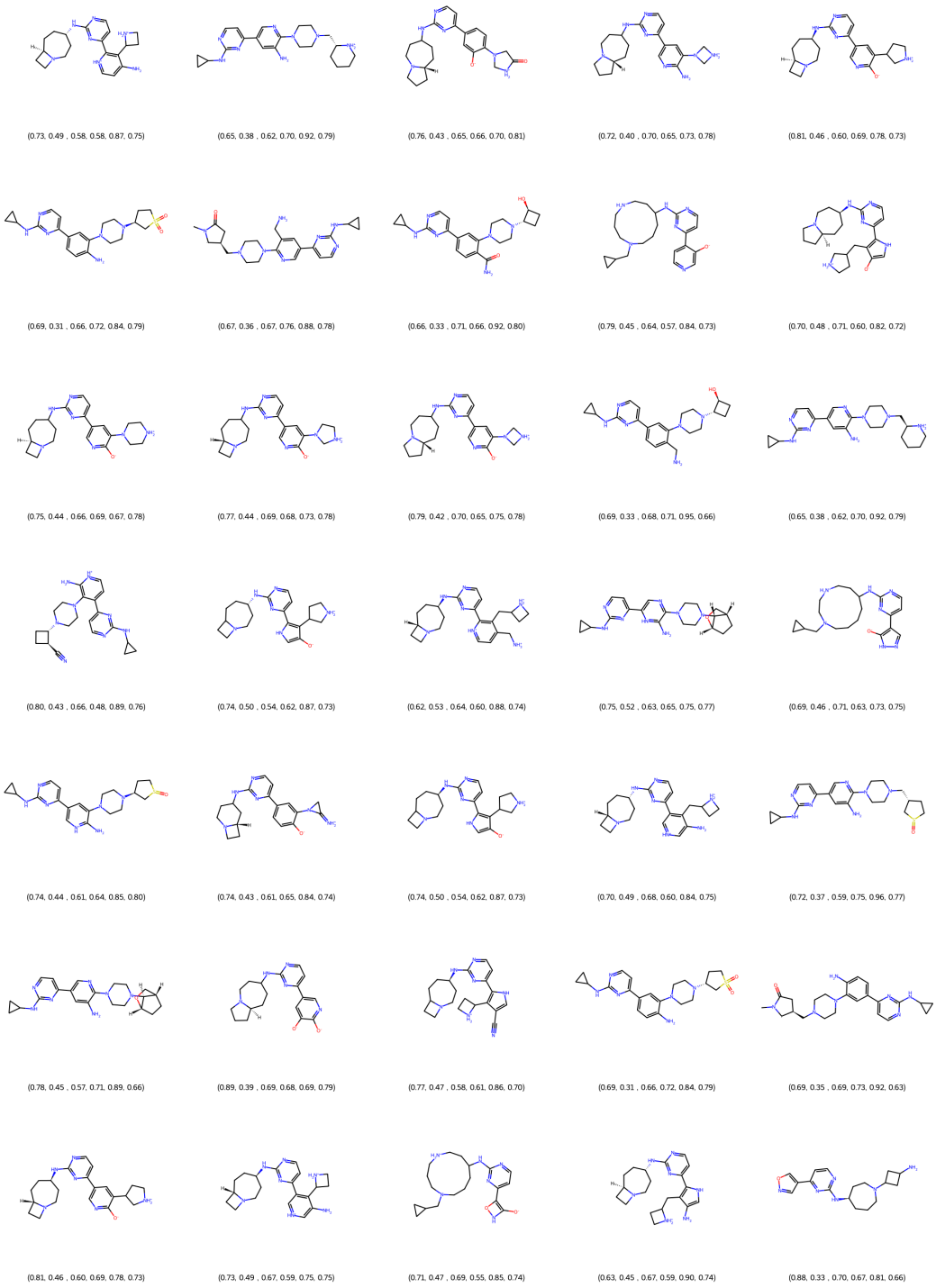


Figure 9: Sampled molecules with high reward scores in Six objectives (QED + SA + GSK3β + JNK3 + DRD2 + Osimertinib MPO) optimization scenario.