7 Supplementary Material

7.1 Additional Proofs and Proof Sketches

Initial Convergence Assurances

To establish the initial convergence result, we first utilize the Bolzano-Weierstrass theorem for compact sets. Given that the feasible set $\mathcal{X}_1$ is compact, the theorem guarantees the existence of a convergent sequence $\{x_k\} \subseteq \mathcal{X}_1$ converging to an arbitrary limit point. We denote this limit point as x_1^*, which is the maximizer of f_1, such as follows:

$$x_k \to x_1^*, \quad x_1^* = \arg\max_{x \in \mathcal{X}_1} f_1(x). \tag{14}$$

Let us assume that f_1 is differentiable at least in a neighborhood of x_1^* and consider x_1^* as a local maximum. The directional derivative of f_1 at x_1^* in any direction u (where u is a unit vector in $\mathbb{R}^n$) satisfies:

$$\nabla f_1(x_1^*)^T u \leq 0, \quad \forall u \in \mathbb{R}^n : \|u\| = 1. \tag{15}$$

As $\{x_k\}$ converges to x_1^*, we can analyze the behavior of f_1 by considering the limit of the gradient inner product:

$$\lim_{k\to\infty} \frac{f_1(x_1^*) - f_1(x_k)}{\|x_1^* - x_k\|} \geq \lim_{k\to\infty} \nabla f_1(x_1^*)^T \frac{x_1^* - x_k}{\|x_1^* - x_k\|} = 0. \tag{16}$$

The vanishing gradient at x_1^* establishes its status as a stationary point. This finding suggests that x_1^* satisfies the necessary condition for a local maximum, affirming it as a viable solution for the sub-problem focused on maximizing f_1.

Divide-and-Conquer and Coordinate Ascent

Here, we draw parallels between our divide-and-conquer approach and the coordinate ascent method. Initially, we obtain a near-optimal solution $\tilde{x}_1$ of the problem $\mathcal{P}_1$, by maximizing f_1. In next stage, $\mathcal{P}_2$ is formulated by incorporating objective f_2. We then define the restricted Pareto set $\mathcal{S}$ by maximizing f_2 while keeping f_1 fixed based on the prior solution $\tilde{x}_1$. This aligns with our divide-and-conquer concept of leveraging the existing solution for f_1 as a starting point, akin to the coordinate ascent method, which optimizes objectives sequentially.

In many-objective optimization problems approached with a coordinate ascent perspective, each 'coordinate' metaphorically represents an 'objective' since we sequentially optimize each objective one at a time. As a result, we interchangeably use the terms 'coordinate' and 'objective' due to their inherent connection. Our underlying hypothesis is that jointly optimizing all objectives in many-objective settings can be highly complex and susceptible to issues like being trapped in poor local optima. Therefore, we embrace the coordinated stagewise approach, where each stage focuses on a subset of objectives, progressively building a comprehensive solution that addresses all objectives.

Theorem 2 (Convergence Properties of Coordinate Ascent)**.** *Consider the optimization problem $\mathcal{P}$ with n objective functions $f_i : \mathcal{X} \to \mathbb{R}$ where $\mathcal{X} \subseteq \mathbb{R}^d$ is a compact set:*

$$\mathcal{P} : \max_{x \in \mathcal{X}} \{f_1(x), f_2(x), \ldots, f_n(x)\}. \tag{17}$$

The goal is to find a Pareto optimal point $x^* \in \mathcal{X}$, where no objective function can be improved simultaneously without worsening at least one other objective.

Given an initial guess $x^{(0)}$, the coordinate ascent method iteratively updates each x_i for $k = 1, 2, 3, \ldots$, to improve the objective functions as follows:

$$\begin{aligned} x_1^{(k)} &\in \arg\max_{x_1} f_1(x_1, x_2^{(k-1)}, \ldots, x_n^{(k-1)}), \\ x_2^{(k)} &\in \arg\max_{x_2} f_2(x_1^{(k)}, x_2, \ldots, x_n^{(k-1)}), \\ &\vdots \\ x_n^{(k)} &\in \arg\max_{x_n} f_n(x_1^{(k)}, \ldots, x_{n-1}^{(k)}, x_n). \end{aligned} \tag{18}$$

Proof Sketch: Given the compact set $\mathcal{X} \subseteq \mathbb{R}^d$ and the continuity of the objective functions f_i on $\mathcal{X}$ for all $i = 1, \ldots, n$, we consider the sequence of iterates $\{x^{(k)}\}$ produced by the coordinate ascent method. For each coordinate update and for every objective function f_i, the update rule ensures a non-decreasing sequence of function values:

$$f_i(x^{(k)}) \geq f_i(x^{(k-1)}), \quad \forall i = 1, \ldots, n \text{ and } \forall k \geq 1. \tag{19}$$

This is due to the fact that each $x_i^{(k)}$ is chosen to maximize f_i, given the fixed values of the other coordinates. Since $\mathcal{X}$ is compact, the sequence $\{x^{(k)}\}$ generated by the coordinate ascent method is contained within this bounded set. By applying the Bolzano-Weierstrass Theorem, we infer the existence of a convergent subsequence $\{x^{(k_j)}\}$ within the bounded sequence $\{x^{(k)}\}$. Hence, this subsequence can converge to a limit point $x^* \in \mathcal{X}$ such as:

$$\exists \{x^{(k_j)}\} \subset \{x^{(k)}\} : x^{(k_j)} \to x^* \text{ as } j \to \infty. \tag{20}$$

Due to the continuity of f_i, the convergence of $\{x^{(k_j)}\}$ to x^* implies the convergence of $f_i(x^{(k_j)})$ to $f_i(x^*)$ for each i:

$$f_i(x^{(k_j)}) \to f_i(x^*) \text{ as } j \to \infty, \quad \forall i = 1, \ldots, n. \tag{21}$$

Suppose that there exists a point $\hat{x} \in \mathcal{X}$ distinct from x^*. We consider the possibility that $\hat{x}$ improves upon x^* in all f_i. Formally, this can be expressed as:

$$f_i(\hat{x}) \geq f_i(x^*), \quad \forall i = 1, \ldots, n, \tag{22}$$

with the strict inequality for at least one objective f_h:

$$\exists h \in \{1, \ldots, n\} : f_h(\hat{x}) > f_h(x^*). \tag{23}$$

However, this would contradict the fact that x^* is a limit of the sequence $\{x^{(k)}\}$ designed to maximize each f_i individually, and due to the non-decreasing nature of f_i along the sequence $\{x^{(k)}\}$. Consequently, no $\hat{x} \in \mathcal{X}$ can satisfy the above inequalities without violating the convergence to x^*. As a result, we can conclude that x^* can be a Pareto optimal point and the sequence $\{x^{(k)}\}$ produced by the coordinate ascent method can progressively converge towards x^*.

Lemma 3 (Fast Convergence Rates of Coordinate Ascent)**.** Let $\{x^{(k)}\}$ be the sequence of iterates produced by coordinate ascent on an L-smooth, μ-strongly concave objective function f_i. The coordinate ascent updates can be defined as:

$$x_i^{(k+1)} = \begin{cases} x_i^{(k)} + \gamma \nabla_i f_i(x^{(k)}), & \text{with probability } p_i \\ x_i^{(k)}, & \text{with probability } 1 - p_i \end{cases} \tag{24}$$

where f_i is assumed to be L-smooth and μ-strongly concave. For an L-smooth function, we have:

$$f_i(x') \geq f_i(x) + \langle \nabla f_i(x), x' - x \rangle - \frac{L}{2}||x' - x||,^2 \quad (25)$$

and the gradients of f_i are Lipschitz continuous with constant L:

$$||\nabla f_i(x) - \nabla f_i(x')|| \leq L||x - x'||. \quad (26)$$

Defining $\epsilon_k = \mathbb{E}[f_i(x^*) - f_i(x^{(k)})]$, where x^* is the optimal solution, the rate under smoothness is:

$$\epsilon_k \leq \epsilon_{k-1} \left(1 - \frac{\eta\mu}{2n}\right), \quad (27)$$

where $\eta > 0$ is a parameter dependent on L and γ. Recursively, this yields:

$$\epsilon_k = O\left(\left(1 - \frac{\eta\mu}{2n}\right)^k\right). \quad (28)$$

Under assumptions of concavity and appropriate step size, the coordinate ascent method can achieve a convergence rate of $O(\log(1/\epsilon_k))$ towards Pareto optimal solutions x^*.

7.2 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 $\mathcal{L}(\theta, 0) = \left[-\log P_\theta(\mathbf{x}) + \log P_{\text{prior}}(\mathbf{x}) + R(\mathbf{x}, 0)\right]^2$
  $N = N + N_{batch}$
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 $\mathcal{L}(\theta, t) = \left[-\log P_\theta(\mathbf{x}) + \log P_{\text{prior}}(\mathbf{x}) + R(\mathbf{x}, t)\right]^2$
  end
  Objective adaptation with entire TopK buffer $x_b = TopK(\mathcal{B})$
  $\mathcal{L}_{\text{OA}}(\theta, t) = \left[-\log P_\theta(x_b) + \log P_{\text{prior}}(x_b) + R(x_b, t+1)\right]^2$
end

7.3 Experimental Settings

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 and 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 utilizing MCMC (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 head 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 and 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 and 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 and Schwaller, 2024] 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].

Implementation Details for DyMol

We implemented the proposed DyMol using PyTorch framework and integrated it within the PMO benchmark. We did not change any hyperparameter settings of the baseline gen-

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

Table 4: The hyperparameter settings used in DyMol.

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

Table 5: Average empirical running times for each method under Four objectives (GSK3β+JNK3+QED+SA) optimization scenario.

erative 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.

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 5.

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.

7.4 Evaulation Metrics

In this section, we provide detailed explanations of the evaluation metrics employed in analyzing our results. Specifically, we utilize two key evaluation metrics: the hypervolume indicator (HV) [Zitzler *et al.*, 2003] and the R2 indicator (R2) [Brockhoff *et al.*, 2012]. Each of these metrics serves a distinct purpose in evaluating the quality of solutions generated by competing methods and our proposed method.

Hypervolume Indicator

The hypervolume indicator, denoted as I_H, is utilized to quantify the volume within the objective space that is covered by the Pareto front generated by each optimization algorithm. For multi- and many-objective optimization problems, this metric is crucial in assessing the degree to which the solutions dominate the objective space. It effectively measures how closely the Pareto front approaches the ideal objectives.

Formally, given a set of non-dominated solutions $\mathcal{X}$ in an n-dimensional objective space and a reference point z^r, where z^r is chosen as the worst acceptable value for each of the objectives, the hypervolume $I_H(\mathcal{X}, z^r)$ is defined as the Lebesgue measure of the region within the objective space that is dominated by the solutions in $\mathcal{X}$ and bounded by z^r. The mathematical representation of the hypervolume (HV) using the Lebesgue integral is as follows:

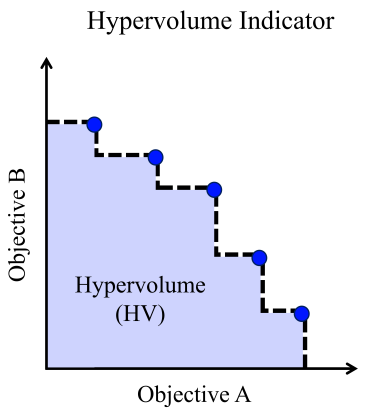


Figure 7: Visualization of the hypervolume indicator in 2D space, where the hypervolume corresponds to the area of the shaded region.

$$I_H(\mathcal{X}, z^r) = \int_{\mathbb{R}^n} \chi_{\bigcup_{x\in\mathcal{X}} H(x,z^r)}(z)\, dz, \tag{29}$$

where $H(x, z^r)$ represents the hyperrectangle that is bounded by the non-dominated solution x and the reference point z^r. The indicator function χ denotes set membership and is defined for an arbitrary set Ψ as:

$$\chi_A(z) = \begin{cases} 1 & \text{if } z \in \Psi, \\ 0 & \text{if } z \notin \Psi. \end{cases} \tag{30}$$

Thus, the integrand $\chi_{\bigcup_{x\in\mathcal{X}} H(x,z^r)}(z)$ is 1 if the point z lies within the hyperrectangle defined by x and z^r, indicating that the point z is within the region covered by the Pareto front and bounded by the reference point. This results in a measure of the volume of the region dominated by $\mathcal{X}$. The hyperrectangle $H(x, z^r)$ for a solution x is defined as:

$$H(x, z^r) = \prod_{i=1}^{n} [z_i^r, x_i], \tag{31}$$

where x_i are the i-th objective values of the solution x, and z_i^r are the components of the reference point z^r. Note that each x_i is at least as great as the corresponding z_i^r since we are dealing with a maximization problem, and the reference point is considered the worst acceptable value for each objective.

As illustrated in Figure 7, the blue points represent a Pareto front, which is a set of non-dominated solutions. Then the HV is simply defined as a measure of the region in the objective space that is dominated by the Pareto front and bounded by a reference point. This reference point is commonly chosen as the worst acceptable value for the objectives. In this study, we set the reference point as the origin (e.g., $(0, 0)$ for two-dimensional space, $(0, 0, 0)$ for three-dimensional space, and so on) in each respective dimensional space, since we have normalized all objective values between 0 and 1. The greater the HV area, the better the Pareto front is considered in terms of its coverage of the objective space. Essentially, the greater HV area indicates a diverse range of trade-offs among the objectives, providing decision-makers with a wider selection of solutions to align with their specific needs or preferences.

R2 Indicator

The R2 Indicator [Brockhoff *et al.*, 2012] is another evaluation metric that assesses the quality of solutions generated by the optimization algorithm. Distinct from the hypervolume indicator, which measures the volume covered by the Pareto front, the R2 indicator calculates an aggregate distance. This distance measures how closely the solutions approach a set of utopian points, each representing the ideal value for the respective objectives. The R2 indicator is part of the R indicator family that employs utility functions to map vectors in multi-dimensional objective space into scalar utility values [Brockhoff *et al.*, 2012]. These utility values serve as a criterion for evaluating the relative quality of Pareto front approximations.

To formally define the R2 indicator, consider a set Ω of general utility functions along with a probability distribution ρ over Ω. Under these conditions, the R2 indicator for a solution set Λ is defined as the expected utility difference between a set of user-defined reference points Ξ and the solution set Λ. Contrary to the hypervolume indicator, which typically uses the worst acceptable values as its reference, the R2 indicator employs a user-defined reference set Ξ. This set is conceptually aligned with the utopian point, which represents ideal but often unattainable objective values. In this work, we set the utopian point as the vector of ones corresponding to the dimensionality of the space, such as $(1, 1)$ for two-dimensional space, $(1, 1, 1)$ for three-dimensional space, and so on. The R2 indicator, $R2(\cdot)$, is formally defined as follows:

$$\begin{aligned} R2(\Xi, \Lambda, \Omega, \rho) = &\int_{\omega\in\Omega} \max_{\xi\in\Xi}\{\omega(\xi)\}\rho(\omega)d\omega \\ &- \int_{\omega\in\Omega} \max_{a\in\Lambda}\{\omega(a)\}\rho(\omega)d\omega. \end{aligned} \tag{32}$$

When Ω represents a discrete and finite set, and we assume a uniform probability distribution p over Ω, the original R2 indicator formula simplifies to the following expression:

$$R2(\Xi, \Lambda, \Omega) = \frac{1}{|\Omega|} \sum_{\omega\in\Omega} \left(\max_{\xi\in\Xi}\{\omega(\xi)\} - \max_{a\in\Lambda}\{\omega(a)\} \right). \tag{33}$$

Given that the first summand in the equation becomes a constant when we assume the reference set Ξ to be constant, we can omit this summand for simplicity and continue to refer to the resulting unary indicator as R2. Consequently, under the assumption of a constant reference set, the R2 indicator can be redefined as a unary indicator, which is expressed as follows [Brockhoff *et al.*, 2012]:

$$R2(\Lambda, \Omega) = -\frac{1}{|\Omega|} \sum_{\omega\in\Omega} \max_{a\in\Lambda}\{\omega(a)\}. \tag{34}$$

The unary R2 indicator equation takes the average of the maximum utility values obtained by applying each utility function in Ω to the solution set Λ. Since these utility values are maximized, the negative sign in front of the summation flips the measure so that higher utility values result in lower R2 values (which are desirable). Therefore, when comparing two solution sets, the one with the lower R2 value is considered better as it indicates the solutions are yielding higher utility values.

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: 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	
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

Table 7: 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.

7.5 Analysis of Many-Objective Scenarios

In this section, we analyze the relationship between score convergence and Pareto diversity, as demonstrated in Table 6 and Table 7. We also present results where objective orders were assigned arbitrarily in Table 8, 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 9 and Table 10.

Score Convergence and Pareto Diversity

In the ablation study presented in Table 2 of our main paper, 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 6, 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 7, a notable performance improvement is evident when both DC and PS are employed together, rather than applying PS alone.

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 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 8. 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

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

Table 8: 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.

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 Five objectives many-objective optimization scenarios using 10 different seeds. In Five objectives scenario, one additional molecular objective was 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

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

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.

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 9 and Table 10, '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.

7.6 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.

Results of Additional Hypervolume Improvement Curves

As shown in Figure 8, 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 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.

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 9, 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 ef-

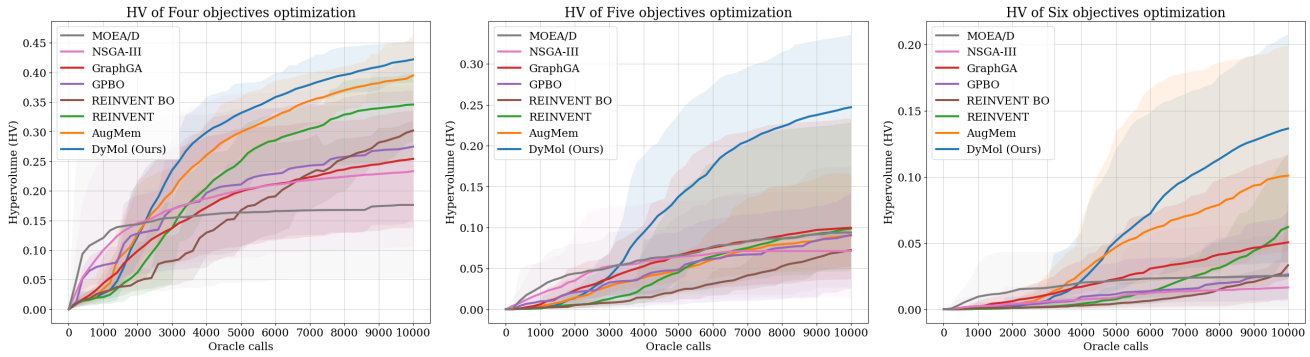


Figure 8: Average HV improvement curves for various many-objective scenarios.

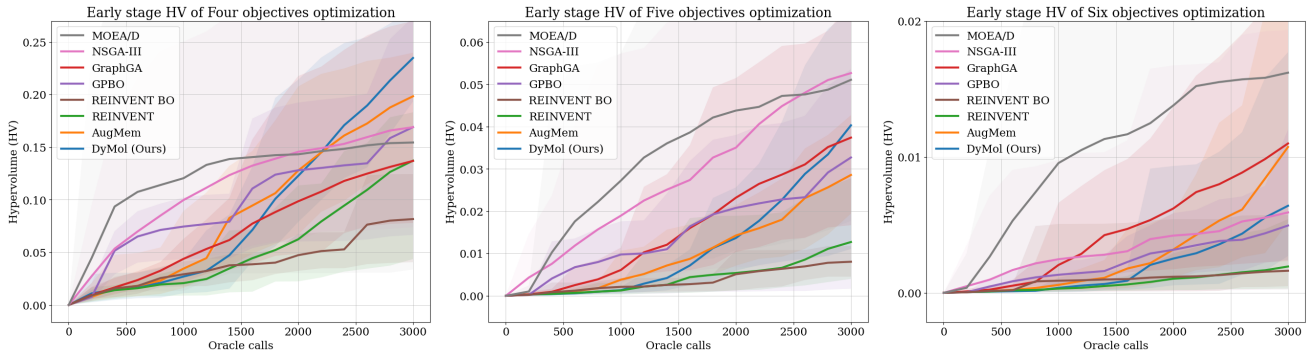


Figure 9: Early stage HV improvement curves for various many-objective scenarios.

fective for the exploration process as they are designed to continually explore and learn from the environment.

7.7 Analysis of Dynamic-Objective Scenarios

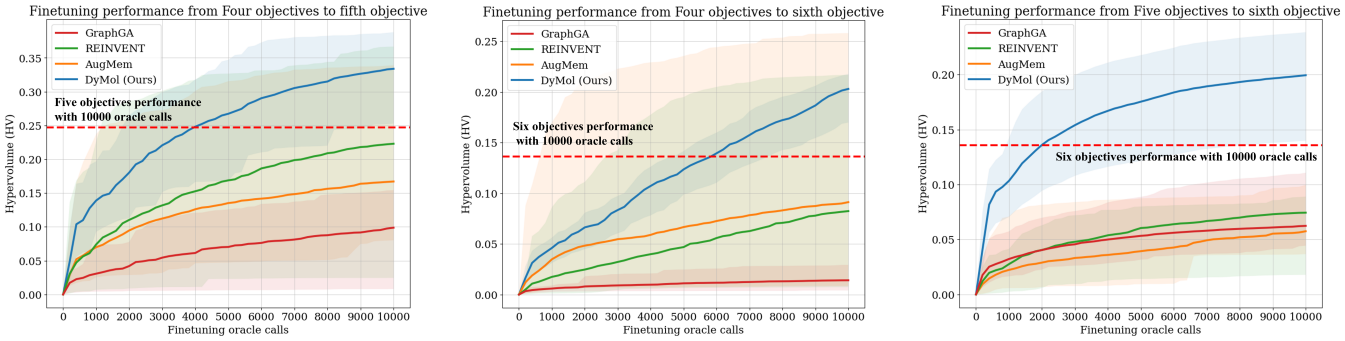
In this section, we explain more detailed information regarding the dynamic-objective scenarios and the primary motivation behind the design of this novel experimental setup. Furthermore, we provide additional experiments that explore dynamic-objective scenarios with different types of molecular objectives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

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.

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.



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

Additional Results for Dynamic-Objective Scenarios

Here, we present additional experimental results for various dynamic-objective scenarios. Figure 10-(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 10-(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 10-(a), we further explore different dynamic-objective scenarios as illustrated in Figures 10-(b) and 10-(c). Figure 10-(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 10-(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 10-(a), both Figures 10-(b) and 10-(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.

7.8 Bemis-Murcko Scaffolds & Carbon Skeletons

In this section, we delve into the analysis of molecular structure diversity by investigating Bemis-Murcko (BM) Scaffolds and Carbon Skeletons (CS) [Bemis and Murcko, 1996]. We provide detailed explanations for these concepts to gain insights into the structural diversity of the generated molecules.

BM scaffolds are a valuable approach for deconstructing organic molecules to discover their fundamental chemical structures. This approach plays a crucial role in medicinal chemistry, particularly in the realm of drug discovery, by enabling the classification and analysis of the core structural elements within chemical compounds. As illustrated in Figure 11, BM scaffolds dissect molecules into simpler constituents by removing side chains while preserving the core structure, which includes ring systems and their connecting linkers. In essence, these scaffolds represent the molecular backbone. This characteristic allows BM scaffolds to serve as an effective quantitative tool in evaluating the structural diversity of chemical compounds. More specifically, they enable the evaluation of structural diversity by facilitating comparisons between a molecule's backbone and its divergence from existing compounds. This is precisely why we have adopted BM scaffolds in our study to comprehensively evaluate the structural diversity of the molecules generated by each method.

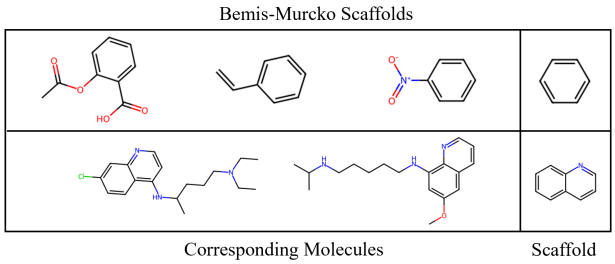
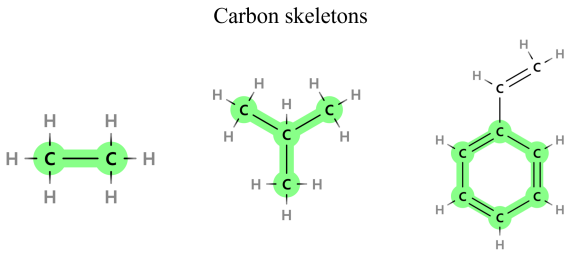


Figure 11: Visual representation of the Bemis-Murcko scaffolds.



(a) Straightline (b) Branching (c) Ring

Figure 12: Visual representation of the carbon skeletons.

Another approach for assessing the structural diversity of molecules is through the examination of carbon skeletons (CS). As depicted in Figure 12, CS exhibits various configurations of carbon atoms in molecules, including straightline skeletons, branching skeletons, and ring structures. In particular, straightline skeletons represent the simplest form of carbon skeletons where carbon atoms are connected in a linear sequence. On the other hand, branching skeletons occur when carbon chains have side chains or branches stemming from the main chain. Therefore, this branching may alter the physical and chemical properties of a molecule such as its reactivity and interaction with biological targets. Additionally, ring structures are closed loops of carbon atoms that are commonly found in many biologically active compounds.

BM scaffolds and CS both aim to simplify and categorize molecular structures to better understand their properties and interactions. While BM scaffolds focus on the core structures by removing side chains and functional groups, CS emphasizes the basic carbon-based structure of a molecule. Consequently, we have incorporated both BM scaffolds and CS in our analysis because this combined approach provides a more comprehensive understanding of structural diversity.

7.9 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.

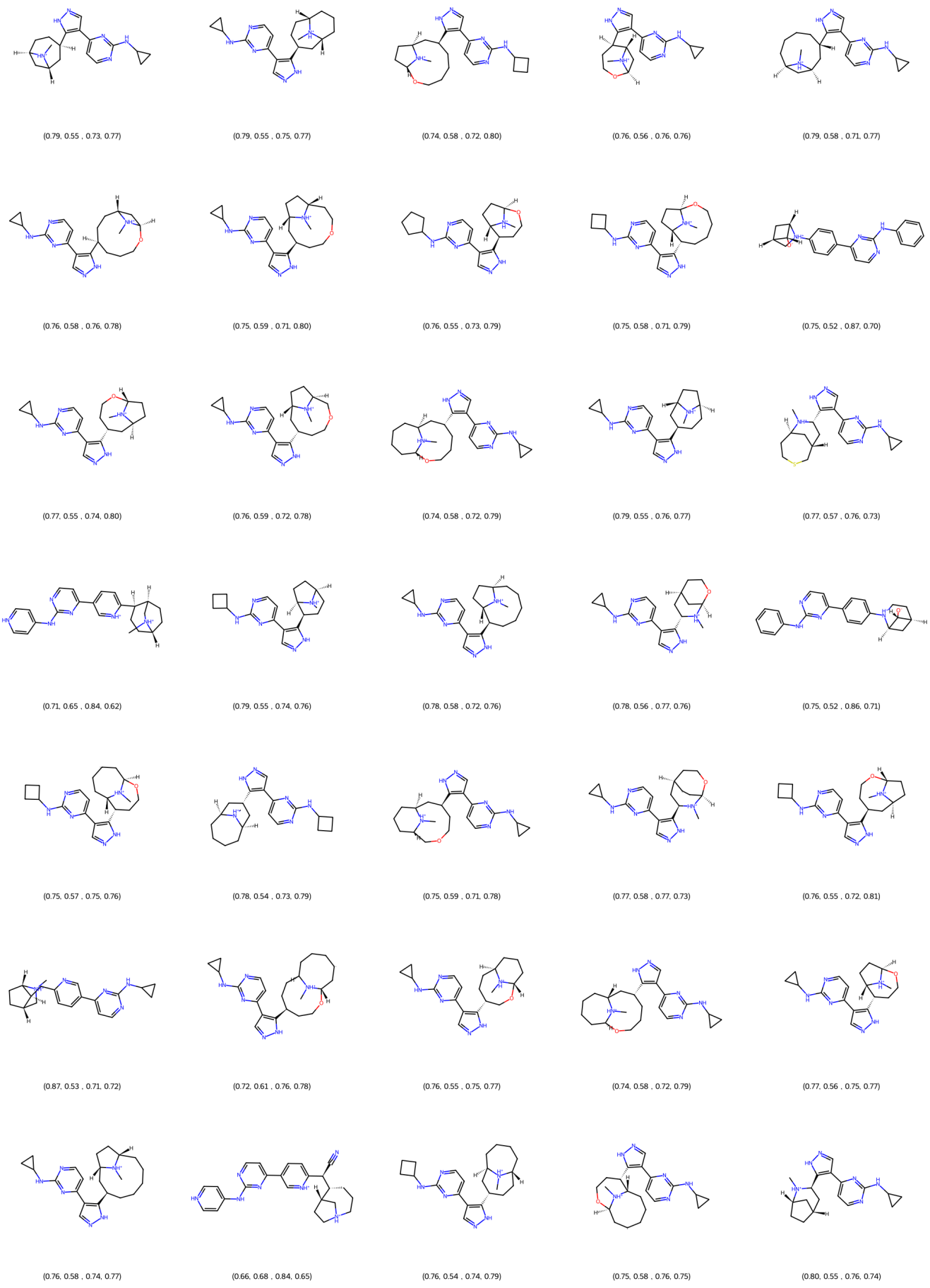


Figure 13: Sampled molecules with high reward scores in Four objectives (QED+SA+GSK3β+JNK3) optimization.

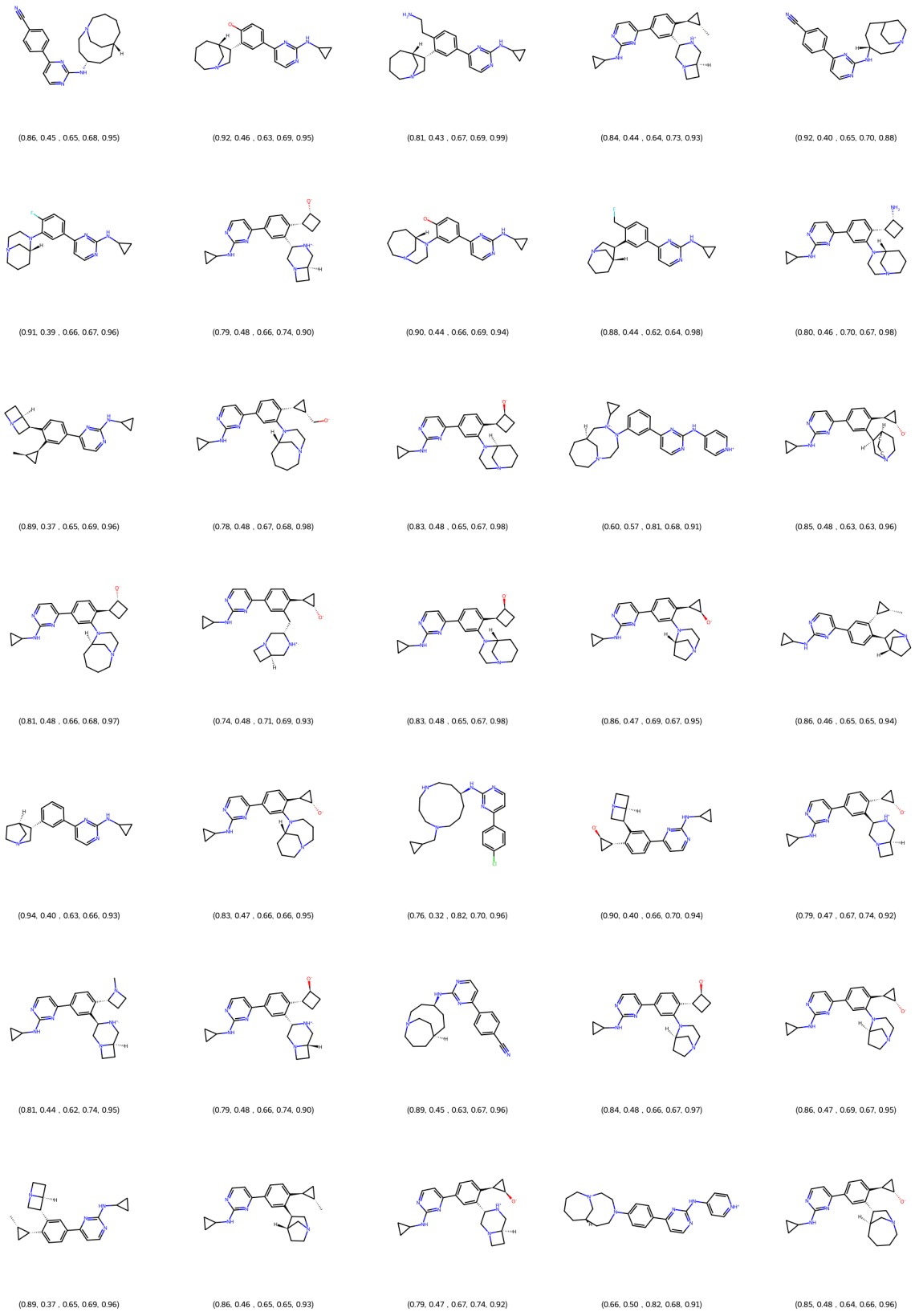


Figure 14: Sampled molecules with high reward scores in Five objectives (QED+SA+GSK3β+JNK3+DRD2) optimization.

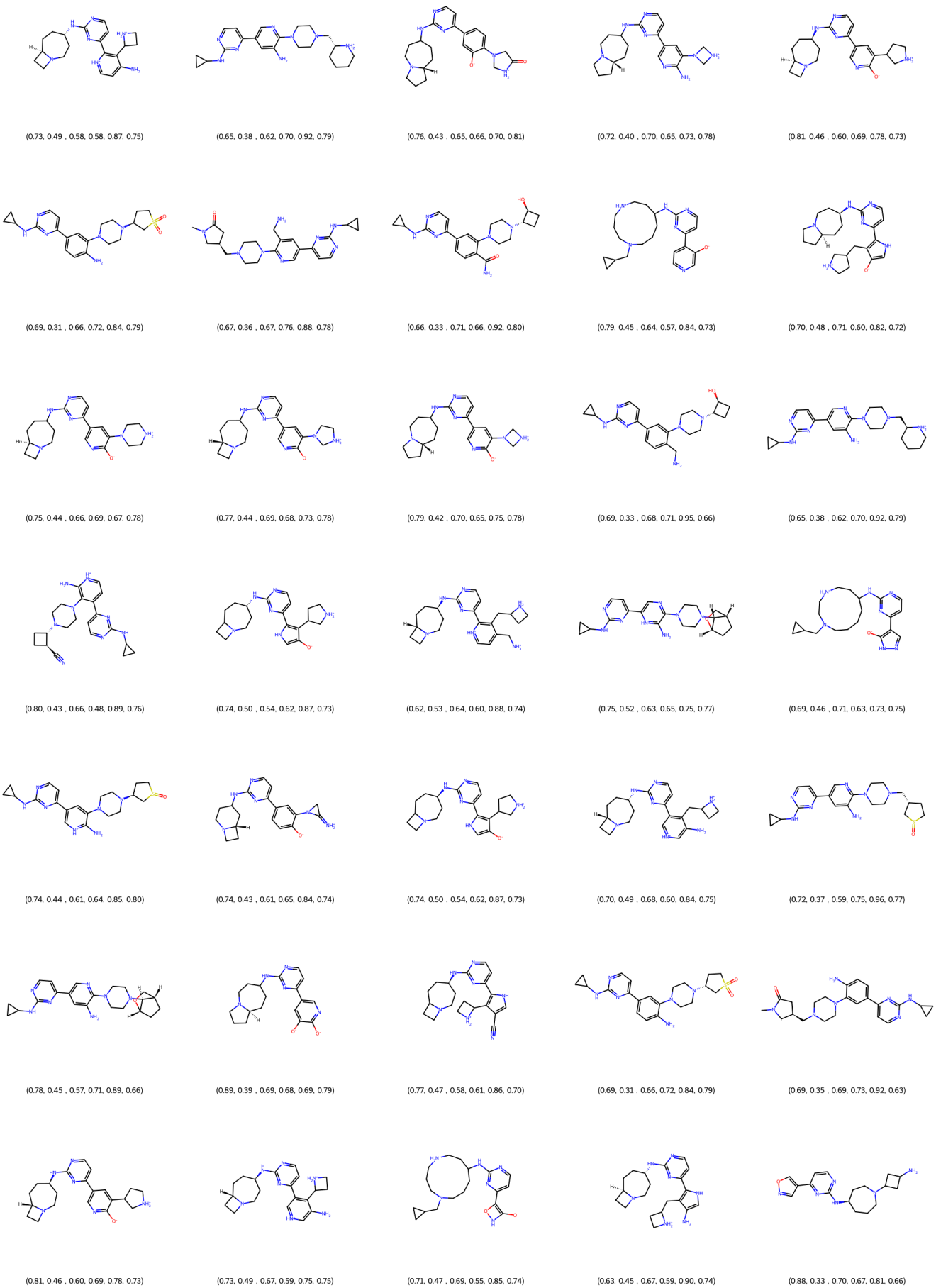


Figure 15: Sampled molecules with high reward scores in Six objectives (QED+SA+GSK3β+JNK3+DRD2+Osimertinib MPO) optimization.