MOLSTITCH: OFFLINE MULTI-OBJECTIVE MOLECULAR OPTIMIZATION WITH MOLECULAR STITCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Molecular discovery is essential for advancing various scientific fields by generating novel molecules with desirable properties. This process is naturally a multi-objective optimization problem, as it must balance multiple molecular properties simultaneously. Although numerous methods have been developed to address this problem, most rely on online settings that repeatedly evaluate candidate molecules through oracle queries. However, in practical applications, online settings may not be feasible due to the extensive time and resources required for each oracle query. To fill this gap, we propose the Molecular Stitching (MolStitch) framework, which utilizes a fixed offline dataset to explore and optimize molecules without the need for repeated oracle queries. Specifically, MolStitch leverages existing molecules from the offline dataset to generate novel 'stitched molecules' that combine their desirable properties. These stitched molecules are then used as training samples to fine-tune the generative model, enhancing its ability to produce superior molecules beyond those in the offline dataset. Experimental results on various offline multi-objective molecular optimization problems validate the effectiveness of MolStitch. MolStitch has been thoroughly analyzed, and its source code is available online.[1]

1 INTRODUCTION

In recent years, diverse *in silico* generative models have been developed to tackle molecular discovery, which is inherently a multi-objective optimization (MOMO) problem (Fromer & Coley, 2023). These computational approaches have demonstrated impressive success across various benchmarks, leading to a growing interest in integrating them into real-world applications such as drug discovery. Despite this success, most existing *in silico* models operate under an online optimization setting, where numerous candidate molecules are generated iteratively and those molecules are evaluated immediately using an oracle function. However, in real-world molecular discovery, the oracle function is typically represented by wet-lab experiments, which are resource-intensive and can take weeks or even months for evaluation (Payton et al., 2023). This creates a significant bottleneck, as *in silico* models cannot receive online evaluation feedback from wet-lab. Instead, these models must wait for the wet-lab experiments to finish, resulting in prolonged delays before they can be optimized.

To address these challenges, a promising research direction is to enable the optimization and refinement of *in silico* models without relying on online evaluation feedback from the wet-lab experiments. To achieve this, we propose to explore offline optimization settings for real-world molecular discovery. Specifically, offline optimization aims to fully leverage the information contained within a static offline dataset, utilizing this information to improve and optimize the model, even in the absence of online evaluation feedback. Detailed explanations for offline settings are provided in Appendix A.

One of the most promising approaches for solving the offline optimization problem is offline model-based optimization (MBO) (Trabucco et al., 2022). In this approach, a proxy model, typically parameterized as a deep neural network $\hat{f}_\theta(\cdot)$, is trained to approximate the oracle function by fitting it to an offline dataset. The trained proxy model then guides the optimization of a generative model, which is designed to generate candidate molecules. In particular, a gradient ascent (Zinkevich, 2003) can be applied to the generative model's parameters with respect to the proxy model's predictions, aiming to optimize the generated candidate molecules toward increasingly desirable properties.

[1] https://tinyurl.com/ycbts7j2

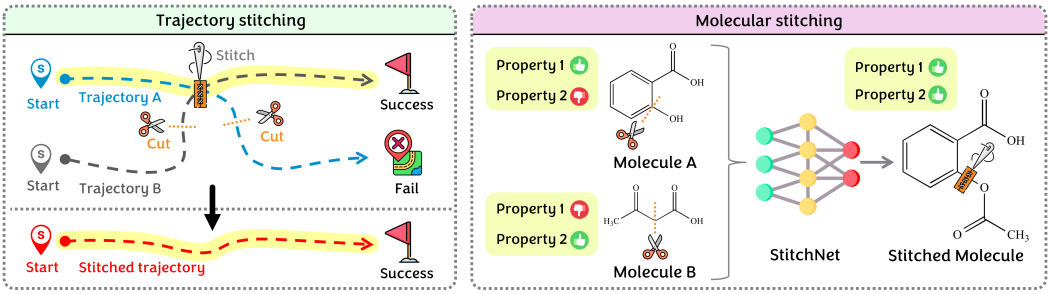


Figure 1: An illustration of trajectory stitching (left) and molecular stitching (right), demonstrating how fragments from distinct trajectories or molecules can be combined to achieve better outcomes.

This offline MBO approach demonstrates a strong performance by generating synthetic data guided by the proxy model. However, there are issues to consider: the vanilla proxy model is trained using a supervised regression loss, which may struggle to accurately approximate the true values from the oracle function as the problem becomes more complex (Fu & Levine, 2021). This issue is further exacerbated when the proxy model encounters out-of-distribution (OOD) data, leading to significant discrepancies between the true values and the proxy model's predictions (Qi et al., 2022). To tackle these issues, recent studies have proposed various strategies to enhance the robustness and accuracy of proxy models, including providing conservative estimates (Trabucco et al., 2021), introducing regularizers (Yu et al., 2021), and adopting multiple proxy models (Chen et al., 2023a).

While these advanced methods significantly enhance the proxy model, they may not fully leverage the valuable information inherent in the offline dataset, as this data is typically used exclusively for training the proxy. In the domain of offline reinforcement learning (RL), researchers have introduced trajectory stitching techniques (Li et al., 2024; Kim et al., 2024b) to directly leverage the existing offline data by creating synthetic trajectories through segment combination. As depicted in Figure 1, consider two distinct trajectories in the offline dataset: trajectory A has a strong start but ends at the wrong destination, whereas trajectory B starts poorly yet successfully reaches the goal destination. By applying trajectory stitching, these trajectories can be combined to form a new stitched trajectory that incorporates the strong start from trajectory A with the goal destination from trajectory B.

In this paper, we propose the Molecular Stitching (MolStitch) framework that effectively tackles the offline MOMO problem. Drawing inspiration from trajectory stitching in offline RL, our framework involves stitching molecules from the offline dataset. For instance, if molecule A possesses desirable property 1 but lacks property 2, while molecule B has the opposite characteristics, we aim to 'stitch' these molecules together to produce a new stitched molecule that exhibits both desirable properties. In other words, our framework utilizes the molecules in the offline dataset to generate novel stitched molecules, allowing the generative model to learn from these newly synthesized data samples.

To effectively utilize stitched molecules as augmented synthetic data, it is essential to evaluate them and provide constructive feedback to the generative model. However, this evaluation process poses a challenge, as these molecules are unfamiliar to the proxy model. This challenge becomes even more pronounced in the MOMO problem due to its increased complexity. To mitigate this, we reformulate the proxy model's task from regression to classification. Instead of directly predicting property scores for stitched molecules, our proxy model is designed to compare pairs of stitched molecules to determine which one is superior based on the desired properties. This transformation simplifies the task for the proxy, thereby enabling it to provide more reliable feedback for the generative model.

In the MOMO problem, it is necessary to optimize multiple molecular objectives (properties) simultaneously. Hence, these objectives are often combined into a single objective through scalarization, where the weights determine the relative importance or priority of each objective (Gunantara, 2018). However, in offline settings, the exact importance of each objective is often unknown, and adjusting weights based on immediate feedback is limited (Xue et al., 2024). To address this, we incorporate priority sampling using a Dirichlet distribution (Minka, 2000) into our framework. Specifically, instead of manually selecting weights, we employ priority sampling to generate a variety of weight configurations during the molecular stitching process, resulting in a diverse set of stitched molecules.

The main contributions of our proposed framework can be summarized as follows:

- We propose the Molecular Stitching (MolStitch) framework, which is the first offline multi-objective optimization approach specifically designed for molecular discovery. In particular, MolStitch includes StitchNet for leveraging existing molecules from an offline dataset to generate novel stitched molecules, a proxy model for evaluating these stitched molecules, and preference optimization technique to fine-tune the generative model without oracle queries.
- We reformulate the proxy model's task from property score regression to pairwise classification. Specifically, we construct a rank-based proxy that learns the ranking relationship between two molecules based on desired properties and classifies which molecule is more favorable.
- We introduce priority sampling using a Dirichlet distribution to efficiently generate diverse weight configurations. This allows for effective exploration of trade-offs among objectives in offline multi-objective optimization, where the importance of each objective is often unknown.

2 Related Work

Multi-Objective Molecular Optimization (MOMO). In recent years, various generative models have been developed to address the MOMO problem, including genetic algorithms (Jensen, 2019; Tripp et al., 2021), sampling-based methods (Xie et al., 2021a; Fu et al., 2021), RL-based methods (Olivecrona et al., 2017; Jin et al., 2020), and GFlowNets (Kim et al., 2024a). To manage multiple objectives, these generative models often employ scalarization techniques, such as weighted sums or Tchebycheff methods, which aggregate multiple objectives into a single objective function. For example, REINVENT (Olivecrona et al., 2017) applies RL algorithms that interact with a chemical environment to generate optimized molecules and can incorporate scalarization techniques to handle multiple objectives. Similarly, GeneticGFN (Kim et al., 2024a) integrates GFlowNets with genetic algorithms to generate molecules and uses scalarization to balance multiple objectives effectively.

Offline Model-based Optimization (MBO). In offline settings, optimization relies solely on a pre-collected dataset and prohibits any real-time oracle queries. A prominent approach for this setting is offline MBO, which performs data augmentation, evaluates synthetic data through a proxy model, and fine-tune the generative model based on the proxy feedback. The most straightforward approach in offline MBO is to use a vanilla proxy that directly approximates objective scores. However, recent studies have proposed various methods to improve the robustness and accuracy of this vanilla proxy. For instance, COMs (Trabucco et al., 2021) employs adversarial learning to encourage conservative estimates on data, while IOM (Qi et al., 2022) leverages invariant representation learning through domain adaptation to reduce distributional shifts. Further details on related work are in Appendix B.

3 Preliminaries

Problem formulation. Let $\mathcal{M}$ denote the space of all possible molecules m, and let $f_1, f_2, \ldots, f_k : \mathcal{M} \to \mathbb{R}$ be k real-valued molecular objective functions, each representing a molecular property to be optimized. The multi-objective molecular optimization (MOMO) problem can be stated as:

$$\underset{m \in \mathcal{M}}{\text{Maximize}} \quad \mathbf{F}(m) = \{f_1(m), f_2(m), \ldots, f_k(m)\}. \tag{1}$$

In this problem, it is often challenging to identify a single molecule that simultaneously maximizes all objective functions. This challenge arises because each objective function reflects a distinct molecular property, and improving one molecular property may lead to the deterioration of other properties due to inherent trade-offs between them (Fromer & Coley, 2023). Therefore, the goal of this problem is to identify a diverse set of Pareto optimal molecules on the Pareto front.

Definition 1 (**Pareto optimal**)**.** A molecule $m^* \in \mathcal{M}$ is considered to be Pareto optimal if and only if there does not exist any other molecule $m \in \mathcal{M}$ such that:

$$\nexists m \in \mathcal{M} : (\forall i \in \{1, \ldots, k\}, f_i(m) \geq f_i(m^*)) \wedge (\exists j \in \{1, \ldots, k\}, f_j(m) > f_j(m^*)). \tag{2}$$

Definition 2 (**Pareto front**)**.** The Pareto front, denoted as $\mathbf{PF}$, is the set of all Pareto optimal solutions in the objective space. Mathematically, it can be expressed as:

$$\mathbf{PF} = \{\mathbf{F}(m^*) \mid m^* \in \mathcal{PS}\}, \tag{3}$$

where $\mathcal{PS}$ is the Pareto set, defined as:

$$\mathcal{PS} = \{m^* \in \mathcal{M} \mid \nexists m \in \mathcal{M} : \mathbf{F}(m) \succeq \mathbf{F}(m^*) \wedge \mathbf{F}(m) \neq \mathbf{F}(m^*)\}. \tag{4}$$

Offline setting. Let $\mathcal{D} = \{(m_n, \mathbf{F}(m_n))\}_{n=1}^{N}$ be the offline dataset, where $m_n \in \mathcal{M}$ represents a pre-collected molecule and $\mathbf{F}(m_n)$ represents the corresponding true molecular objective scores. The goal of this offline molecular optimization is to identify Pareto optimal molecules within $\mathcal{D}$ and to generate new molecules that potentially outperform the best-known molecules in $\mathcal{D}$. To explore molecular space beyond the dataset $\mathcal{D}$, a common strategy involves constructing a proxy model, $\hat{f}_\theta(\cdot) : \mathcal{M} \to \mathbb{R}$, to evaluate molecules. The most direct approach is the vanilla proxy, which approximates the scores of true objective functions $\mathbf{F}(\cdot)$ by training on the mean squared error loss.

Generative model. Let G_ϕ be a generative model that generates molecules in an auto-regressive manner. The generation process for a molecule m of total length T can be stated as:

$$G_\phi(m) = \prod_{t=1}^{T} G_\phi(m^t | m^{t-1}, m^{t-2}, \ldots, m^1), \tag{5}$$

where m^t represents the t-th component (or token) in the sequence that constitutes the molecule m. To optimize the generative model such that it produces molecules with improved objective scores, the vanilla proxy can be employed. Specifically, the generative model can be updated by maximizing the expected performance of the generated molecules based on the vanilla proxy's predictions:

$$\phi^* = \arg\max_{\phi} \mathbb{E}_{m \sim G(\phi)} \left[\hat{f}_\theta(m) \right]. \tag{6}$$

However, this approach may face challenges as the problem's complexity increases. The vanilla proxy might produce unreliable predictions when encountering molecules outside its training data distribution, leading to potentially misguided optimization. Moreover, this approach may not fully leverage the valuable information inherent in the existing offline dataset.

4 METHOD

In this section, we present our MolStitch framework for tackling the offline MOMO problem. There are three distinct neural networks in our framework: the generative model, StitchNet, and the proxy model. The generative model is designed to generate molecules in textual formats, such as SMILES (Weininger, 1988). StitchNet takes two parent molecules as input and outputs a novel stitched molecule that combines desirable properties from both inputs. The proxy model serves as a surrogate for evaluating molecules by classifying which molecule in a given pair has more desirable properties.

4.1 UNSUPERVISED PRE-TRAINING FOR STITCHNET AND THE GENERATIVE MODEL

In the pre-training stage of our framework, we conduct unsupervised training for both StitchNet and the generative model using the public ZINC dataset (Sterling & Irwin, 2015). For pre-training StitchNet, we randomly sample two parent molecules from the dataset and employ a rule-based crossover operator (Jensen, 2019) to generate an offspring molecule, as shown in Figure 2. This operator ensures that the offspring molecules are chemically valid and potentially possess desirable properties (Kamphausen et al., 2002). We then train StitchNet using a maximum likelihood approach (Myung, 2003) to produce a stitched molecule that closely resembles the offspring molecule. This pre-training encourages StitchNet to internalize chemical grammar, thereby enabling it to generate stitched molecules that are chemically valid.

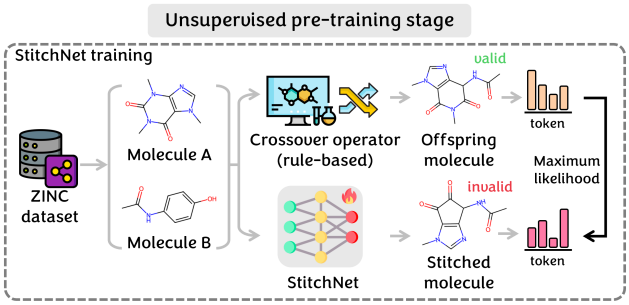


Figure 2: Unsupervised pre-training for our StitchNet.

For pre-training the generative model, we also randomly sample molecules from the ZINC dataset and use them as ground truth labels. The model is then trained using a maximum likelihood, wherein it learns to predict the next component of each molecule based on the preceding sequence, as outlined in Equation 5. Since all molecules within the ZINC dataset are chemically valid, this pre-training process naturally guides the generative model to generate chemically valid molecules on its own. The visualization of this pre-training process for the generative model is provided in Appendix C.

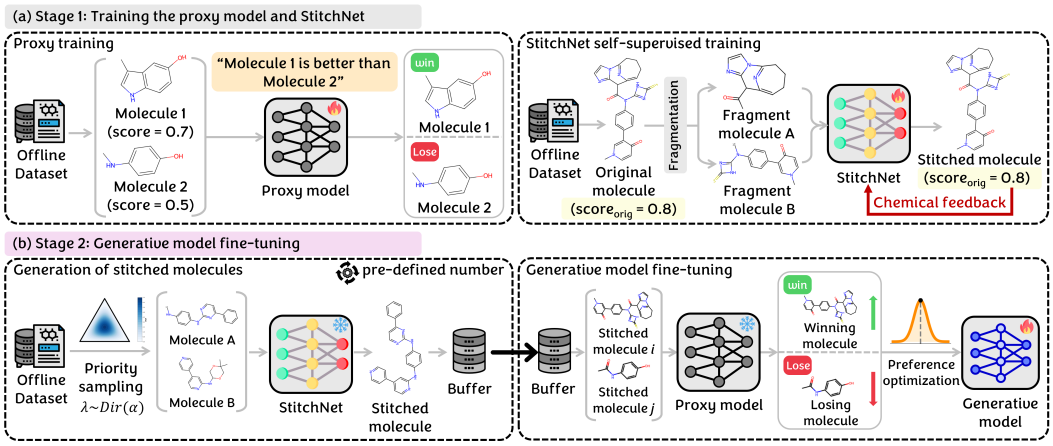


Figure 3: Overview of the MolStitch framework. (a) Stage 1: The proxy model is trained to classify which molecule in a given pair has desirable properties, while StitchNet undergoes self-supervised training with chemical feedback. (b) Stage 2: StitchNet generates stitched molecules, which are stored in a buffer. Once the buffer is full, the proxy model evaluates the pairs and selects the superior molecule. The generative model is then fine-tuned using preference optimization techniques.

4.2 Training the Proxy and StitchNet for Offline Molecular Optimization

In the first stage, we obtain the offline dataset that consists of pre-collected molecules along with their true molecular objective scores. To facilitate the process of offline MBO, we require a proxy model capable of evaluating each molecule effectively. In this work, rather than training the proxy model to approximate the exact objective scores, we train it to classify which molecule in a given pair has better objective scores. Specifically, as shown in Figure 3, we sample pairs of molecules from the offline dataset. Since we have access to the ground truth objective scores for each molecule in this dataset, we can establish a ranking between the molecules in each pair. We then train our rank-based proxy model $\hat{f}_\theta$ to learn this ranking relationship using a pairwise ranking loss as follows:

$$\mathcal{L}_{\text{proxy}}(\theta) = \frac{1}{|\mathcal{P}|} \sum_{(m_w, m_l) \in \mathcal{P}} \ell\left(\hat{f}_\theta(m_w) - \hat{f}_\theta(m_l), \mathbf{F}(m_w) - \mathbf{F}(m_l)\right), \tag{7}$$

where $\mathcal{P}$ is the set of all valid molecule pairs (m_w, m_l) within the offline dataset $\mathcal{D}$, defined as:

$$\mathcal{P} = \{(m_w, m_l) \mid m_w, m_l \in \mathcal{D},\ \mathbf{F}(m_w) > \mathbf{F}(m_l)\}. \tag{8}$$

The loss function ℓ penalizes the proxy model when the predicted ranking does not match the true ranking. A common choice for ℓ is the binary cross-entropy loss, which can be re-written as:

$$\mathcal{L}_{\text{proxy}}(\theta) = -\frac{1}{|\mathcal{P}|} \sum_{(m_w, m_l) \in \mathcal{P}} \left[\log \sigma\left(\hat{f}_\theta(m_w) - \hat{f}_\theta(m_l)\right)\right], \tag{9}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. Once the proxy model has been effectively trained to rank pairs of molecules, we proceed to the self-supervised training process for StitchNet. While the pre-training stage focused on training StitchNet to learn chemical grammar and crossover operation, the focus in this stage is to integrate chemical feedback into StitchNet. In particular, we leverage the true objective scores from the offline dataset as chemical feedback to inform StitchNet about the potential efficacy of the resulting stitched molecules. This feedback helps StitchNet to understand how the stitched molecules are likely to exhibit objective scores when two molecules are combined.

To achieve this, we first sample the original molecule m_{orig} from the offline dataset $\mathcal{D}$. Subsequently, we use the fragmentation function within the rule-based crossover operator to decompose this original molecule into two smaller fragment molecules. StitchNet is then employed to recombine these fragment molecules into a new stitched molecule $\bar{m}_{\text{stit}}$. If the molecular similarity (Bender & Glen, 2004) between the original molecule and stitched molecules is above a certain threshold δ, $\texttt{sim}(m_{\text{orig}}, \bar{m}_{\text{stit}}) \geq \delta$, we then train StitchNet $\mathcal{S}_\psi$ using the following loss function:

$$\mathcal{L}_{\text{stitch}}(\psi) = \frac{1}{|\mathcal{D}|} \sum_{m_{\text{orig}} \in \mathcal{D}} \mathbb{E}_{\bar{m}_{\text{stit}} \sim \mathcal{S}_\psi} \left[\left(-\log \mathcal{S}_\psi(\bar{m}_{\text{stit}}) + \log \mathcal{S}_{\text{ref}}(\bar{m}_{\text{stit}}) + \mathcal{R}(m_{\text{orig}})\right)^2\right], \tag{10}$$

where $\mathcal{S}_{\text{ref}}$ refers to the pre-trained StitchNet that acts as a reference model for maintaining chemical validity throughout the molecular stitching process. The $\mathcal{R}(m_{\text{orig}})$ represents the reward score, serving as chemical feedback derived from the given objective scores of the original molecule m_{orig}. The $\mathcal{L}_{\text{stitch}}(\psi)$ guides StitchNet $\mathcal{S}_\psi$ to generate stitched molecules $\bar{m}_{\text{stit}}$ with desirable objective scores, while not deviating too far from $\mathcal{S}_{\text{ref}}$. Note that since we are addressing the offline MOMO problem, we cannot query the oracle to directly measure the objective scores of $\bar{m}_{\text{stit}}$ for computing $\mathcal{R}(\bar{m}_{\text{stit}})$. Instead, we utilize the given objective scores of m_{orig} as a form of chemical feedback to approximate the objective scores of $\bar{m}_{\text{stit}}$. This approximation is reasonable because StitchNet generates $\bar{m}_{\text{stit}}$ by recombining fragment molecules that are derived directly from m_{orig}. Moreover, we ensure that $\bar{m}_{\text{stit}}$ is sufficiently similar to m_{orig} through the similarity threshold δ. This allows us to assume that the objective scores of $\bar{m}_{\text{stit}}$ are also similar to the objective scores of m_{orig}, as it is widely acknowledged that structurally similar molecules often exhibit similar properties and biological activities (Barbosa & Horvath, 2004; Alvesalo et al., 2006). Detailed visualization of this process is in Appendix E.

4.3 Offline Molecular Optimization via Molecular Stitching

In the second stage of our framework, we address the offline MOMO problem by utilizing the trained proxy model and StitchNet. The main goal of this stage is to train the generative model to generate novel molecules that potentially surpass the best-known molecule in $\mathcal{D}$. In the context of the MOMO problem, the scalarization approach is widely adopted, where a weighted sum of multiple objectives is combined into a single scalar objective, expressed as $F(m) = \sum_{i=1}^{k} \lambda_i f_i(m)$. Here, k denotes the number of objectives, and λ_i represents the weight assigned to each objective, reflecting its relative importance or priority. However, in offline settings, the exact importance is often unknown, making it challenging to select appropriate weights. In addition, the goal of StitchNet is to combine molecules with different characteristics to generate novel stitched molecules that integrate desirable properties from both inputs. Hence, it is essential to provide StitchNet with diverse molecule pairs.

To address these challenges, we introduce priority sampling using the Dirichlet distribution. This sampling approach generates a diverse set of weight configurations, allowing StitchNet to work with a wide variety of molecule pairs, each focusing on a different balance among multiple objectives. Our choice of the Dirichlet distribution is due to its capability to sample directly from the simplex, naturally providing valid weight combinations that are non-negative and sum to 1. The probability density function of the Dirichlet distribution can be expressed by:

$$p(\lambda_1, \lambda_2, \ldots, \lambda_k \mid \lambda \sim \text{Dir}(\alpha_1, \alpha_2, \ldots, \alpha_k)), \tag{11}$$

where $\text{Dir}(\cdot)$ refers to the Dirichlet distribution, and α denotes the concentration parameters. As illustrated in Figure 3, we use priority sampling $\lambda \sim \text{Dir}(\alpha_1, \alpha_2, \ldots, \alpha_k)$ to sample molecule pairs from the offline dataset. These sampled molecules are then fed into StitchNet, which outputs a novel stitched molecule $\bar{m}$. This newly generated stitched molecule is subsequently stored in a buffer $\mathcal{B}$ and utilized as a training sample for the fine-tuning training process of the generative model. Please refer to Appendix F for a detailed visualization and the rationale behind priority sampling.

Once the buffer $\mathcal{B}$ is populated with a pre-defined number of stitched molecules, we can proceed to train the generative model. Specifically, we sample pairs of stitched molecules $(\bar{m}_i, \bar{m}_j)$ from $\mathcal{B}$ and use our trained proxy model to determine which molecule in each pair is more favorable such as:

$$(\bar{m}_w, \bar{m}_l) = \Big\{ (\bar{m}_i, \bar{m}_j), \quad \text{if } \hat{f}_\theta(\bar{m}_i) > \hat{f}_\theta(\bar{m}_j) \tag{12}$$

where the more favorable molecule is denoted as $\bar{m}_w$ (the winning molecule) and the less favorable molecule is $\bar{m}_l$ (the losing molecule). Then, we can update the generative model G_ϕ by increasing the log-likelihood of generating the winning molecule and decreasing the log-likelihood of generating the losing molecule. The loss function for the generative model can be formulated as:

$$\mathcal{L}_{\text{gen}}(\phi) = -\mathbb{E}_{(\bar{m}_w, \bar{m}_l) \sim \mathcal{B}} \left[\log G_\phi(\bar{m}_w) - \log G_\phi(\bar{m}_l) \right] + \beta \cdot \mathbb{D}_{\text{KL}}(G_\phi \| G_{\text{ref}}), \tag{13}$$

where G_{ref} represents the pre-trained generative model serving as a reference model. The KL divergence term $\mathbb{D}_{\text{KL}}$ encourages G_ϕ not to deviate significantly from G_{ref}, ensuring that it maintains adherence to chemical validity. After formulating the initial loss function for the generative model, we can draw an intriguing parallel to preference optimization for language models (Rafailov et al., 2023; Tang et al., 2024). In this analogy, our generative model G_ϕ can be thought of as the language

model and the favorable molecule $\bar{m}_w$ as the preferred response. This conceptual alignment allows us to incorporate various preference optimization techniques into our training process. Inspired by Direct Preference Optimization (DPO) (Rafailov et al., 2023), we can reformulate the Equation 13 into a DPO-like loss by employing the Bradley-Terry model (Bradley & Terry, 1952) such as follow:

$$\mathcal{L}_{\text{gen-dpo}}(\phi) = -\mathbb{E}_{(\bar{m}_w,\bar{m}_l)\sim\mathcal{B}}\left[\log\sigma\left(\beta\log\frac{G_\phi(\bar{m}_w)}{G_{\text{ref}}(\bar{m}_w)} - \beta\log\frac{G_\phi(\bar{m}_l)}{G_{\text{ref}}(\bar{m}_l)}\right)\right]. \tag{14}$$

This DPO-like loss integrates the separate KL divergence into a single term by utilizing the sigmoid of log odds ratios, simplifying the optimization process. In addition, the sigmoid function mitigates extreme values and provides more stable gradients during training. However, despite its effectiveness, DPO is known to be prone to overfitting the preference dataset, particularly in scenarios where there is a deterministic preference between two samples (Hu et al., 2024a). To address this, Identity Preference Optimization (IPO) (Azar et al., 2024) introduces a regularization term that penalizes the model when its confidence in the preference margin becomes excessively high. Building upon the concepts of IPO, we can modify the Equation 14 to adopt an IPO-like loss formulation as follows:

$$\mathcal{L}_{\text{gen-ipo}}(\phi) = -\mathbb{E}_{(\bar{m}_w,\bar{m}_l)\sim\mathcal{B}}\left[\left(\log\left(\frac{G_\phi(\bar{m}_w)}{G_\phi(\bar{m}_l)}\cdot\frac{G_{\text{ref}}(\bar{m}_l)}{G_{\text{ref}}(\bar{m}_w)}\right) - \frac{1}{2\beta}\right)^2\right]. \tag{15}$$

Using the Equation 15, we fine-tune the generative model, which is REINVENT (Olivecrona et al., 2017), chosen for its widespread use and robust performance. Details of the generative model's loss function are in Appendix D, and the pseudo-code for our MolStitch framework is in Appendix G.

5 Experiments

5.1 Experimental Design and Results

Experimental setup. We conducted two main offline MOMO experiments to evaluate the efficacy of our MolStitch framework. The first benchmark focused on the Molecular Property Optimization (MPO) task (Gao et al., 2022), while the second addressed the docking score optimization task (Lee et al., 2023). Both experiments were designed to simulate real-world constraints by restricting the number of oracle calls. In the first experiment, we closely followed prior studies (Xie et al., 2021b; Shin et al., 2024) and adopted four widely used molecular objectives. The objectives include JNK3 and GSK3β, which evaluate inhibition against target proteins associated with Alzheimer's disease, along with QED and SA, which measure drug-likeness and synthesizability. For the second experiment, we also closely followed recent work (Guo & Schwaller, 2024b) and targeted the docking score optimization of five proteins—parp1, fa7, jak2, braf, and 5ht1b—alongside QED and SA. Note that all experiments were conducted under offline settings, and each experiment was repeated with 10 different seeds to ensure reliability. Further experimental details are in Appendix H.

Competing methods. We compared our framework against two main categories of methods: molecular optimization and offline optimization. For molecular optimization, we included REINVENT (Olivecrona et al., 2017), REINVENT-BO (Tripp et al., 2021), AugMem (Guo & Schwaller, 2024a), GraphGA (Jensen, 2019), DST (Fu et al., 2022), GeneticGFN (Kim et al., 2024a), and Saturn (Guo & Schwaller, 2024b). For offline optimization, we considered various offline MBO methods, including Gradient ascent (Grad) (Zinkevich, 2003), COMs (Trabucco et al., 2021), IOM (Qi et al., 2022), RoMA (Yu et al., 2021), Ensemble Proxy (Trabucco et al., 2022), ICT (Yuan et al., 2023), and Tri-Mentoring (Chen et al., 2023a). We included BIB (Chen et al., 2023b) and BootGen (Kim et al., 2023), which are current state-of-the-art models for offline optimization in biological sequence design. Note that we used REINVENT as the backbone generative model for all offline optimization methods, not only because it is one of the most robust models for diverse molecular optimization tasks, but also to ensure fairness and consistency, as REINVENT serves as the main backbone model in our framework. Detailed descriptions of each competing method are in Appendix I.

Evaluation metrics. The performance of each method was evaluated using two evaluation metrics: the hypervolume indicator (HV) (Zitzler et al., 2003) and the R2 indicator (Brockhoff et al., 2012). The HV quantifies the volume of the space dominated by a set of solutions on the Pareto front, where higher values reflect better performance. On the other hand, the R2 assesses the quality of a solution set by measuring the projection onto pre-defined reference points, with lower values indicating better performance. A more detailed explanation of these evaluation metrics is presented in Appendix J.

Table 1: Experimental results on molecular property optimization tasks under the full-offline setting. The evaluation metrics are the hypervolume (HV) and R2 indicators, with the best values in bold.

Molecular objectives	GSK3β+JNK3		GSK3β+JNK3+QED		GSK3β+JNK3+QED+SA	
Method	HV(↑)	R2(↓)	HV(↑)	R2(↓)	HV(↑)	R2(↓)
REINVENT	0.462±0.133	0.921±0.259	0.196±0.083	2.646±0.327	0.168±0.046	3.969±0.664
AugMem	0.489±0.077	0.845±0.148	0.272±0.083	2.118±0.280	0.185±0.043	4.101±0.346
GraphGA	0.367±0.090	1.116±0.189	0.212±0.063	2.482±0.240	0.200±0.070	3.973±0.504
DST	0.327±0.070	1.211±0.141	0.244±0.072	2.341±0.321	0.228±0.065	3.748±0.473
Saturn	0.531±0.087	0.785±0.159	0.293±0.058	1.977±0.280	0.281±0.058	3.339±0.280
GeneticGFN	0.482±0.073	0.869±0.117	0.309±0.087	1.990±0.365	0.237±0.066	3.630±0.453
REINVENT-BO	0.472±0.107	0.909±0.216	0.232±0.086	2.385±0.393	0.205±0.105	3.974±0.895
Grad	0.494±0.058	0.857±0.126	0.205±0.045	2.502±0.231	0.171±0.026	4.176±0.319
COMs	0.479±0.063	0.877±0.109	0.205±0.072	2.496±0.288	0.171±0.062	4.219±0.628
IOM	0.506±0.070	0.807±0.138	0.215±0.060	2.380±0.336	0.195±0.065	4.042±0.529
RoMA	0.492±0.091	0.843±0.177	0.198±0.052	2.537±0.269	0.169±0.071	4.207±0.617
Ensemble Proxy	0.500±0.033	0.835±0.055	0.218±0.039	2.462±0.160	0.213±0.057	3.888±0.529
BIB	0.486±0.070	0.874±0.120	0.203±0.049	2.503±0.245	0.172±0.027	4.080±0.387
BootGen	0.540±0.113	0.741±0.167	0.225±0.067	2.452±0.319	0.201±0.074	4.092±0.560
ICT	0.514±0.049	0.827±0.104	0.213±0.080	2.429±0.385	0.180±0.060	4.197±0.593
Tri-Mentoring	0.510±0.042	0.824±0.079	0.216±0.071	2.458±0.363	0.195±0.057	4.067±0.467
MolStitch (Ours)	**0.579±0.070**	**0.698±0.128**	**0.403±0.065**	**1.649±0.259**	**0.352±0.080**	**2.953±0.571**

Table 2: Experimental results on docking score optimization tasks under the full-offline setting.

Target protein	parp1	jak2	braf	fa7	5ht1b
Method	HV(↑)	HV(↑)	HV(↑)	HV(↑)	HV(↑)
REINVENT	0.515±0.016	0.477±0.009	0.500±0.008	0.414±0.006	0.509±0.011
AugMem	0.532±0.039	0.499±0.053	0.511±0.008	0.430±0.038	0.521±0.014
Saturn	0.528±0.009	0.498±0.030	0.523±0.046	0.431±0.034	0.537±0.033
GeneticGFN	0.539±0.033	0.476±0.008	0.508±0.005	0.441±0.054	0.523±0.011
REINVENT-BO	0.518±0.009	0.480±0.007	0.505±0.012	0.421±0.067	0.518±0.012
Grad	0.513±0.007	0.481±0.014	0.510±0.007	0.445±0.053	0.525±0.033
COMs	0.510±0.010	0.478±0.014	0.505±0.022	0.411±0.007	0.509±0.008
IOM	0.520±0.009	0.474±0.008	0.500±0.013	0.411±0.005	0.519±0.042
RoMA	0.512±0.010	0.470±0.009	0.512±0.032	0.429±0.053	0.512±0.013
Ensemble Proxy	0.517±0.008	0.479±0.010	0.501±0.010	0.414±0.006	0.507±0.008
BIB	0.514±0.010	0.476±0.007	0.497±0.006	0.414±0.006	0.505±0.009
BootGen	0.544±0.032	0.496±0.007	0.524±0.007	0.436±0.030	0.545±0.063
ICT	0.516±0.005	0.476±0.006	0.504±0.021	0.410±0.005	0.506±0.010
Tri-Mentoring	0.529±0.038	0.482±0.017	0.511±0.019	0.416±0.008	0.513±0.009
MolStitch (Ours)	**0.560±0.037**	**0.515±0.041**	**0.554±0.042**	**0.451±0.061**	**0.575±0.051**

Main results. As shown in Table 1, we present the mean HV and R2 performance along with their standard deviations for the MPO task under the full-offline setting. We observed that our MolStitch framework consistently demonstrated superior performance across all scenarios with varying numbers of molecular objectives. This underscores the efficacy of our StitchNet in addressing the offline MOMO problem, as it leverages existing molecules to create novel stitched molecules, which serve as valuable training samples for fine-tuning the generative model. Among the competing methods, Saturn and GeneticGFN exhibited strong performance, both of which are recent methods that employ genetic algorithms, while BootGen demonstrated its effectiveness by utilizing a bootstrapping technique for iterative self-training. Furthermore, we validated the effectiveness of our framework on an additional protein docking score optimization task. As presented in Table 2, MolStitch consistently outperformed all competing methods across all five proteins in terms of the HV performance, highlighting the robustness and generalizability of our framework in tackling diverse offline MOMO.

Additional results. In recent years, semi-offline optimization, also known as batch hybrid learning, has gained significant attention in the field of large language models (Xiong et al., 2024). Specifically, this semi-offline setting allows for a limited number of online human feedback cycles and enables the model to be fine-tuned on new data through large batches. Inspired by this, we conducted additional experiments for the semi-offline setting, starting with an offline dataset and periodically querying oracle functions to evaluate molecules in large batches. Due to page constraints, the results are in Appendix K.3, where our framework maintained its superior performance. We also explored the impact of different backbone generative models in our framework, as detailed in Appendix K.4.

Table 3: An ablation study for Rank-based Proxy (RP), StitchNet (SN), and Priority Sampling (PS).

Ablation			GSK3β+JNK3		GSK3β+JNK3+QED		GSK3β+JNK3+QED+SA	
RP	SN	PS	HV(↑)	R2(↓)	HV(↑)	R2(↓)	HV(↑)	R2(↓)
-	-	-	0.494±0.058	0.857±0.126	0.205±0.045	2.502±0.231	0.171±0.026	4.176±0.319
-	✔	-	0.513±0.073	0.780±0.106	0.269±0.081	2.183±0.318	0.193±0.053	4.134±0.502
-	✔	✔	0.505±0.049	0.824±0.084	0.277±0.083	2.195±0.357	0.220±0.054	3.835±0.483
✔	-	-	0.545±0.063	0.773±0.120	0.319±0.059	1.928±0.314	0.251±0.084	3.504±0.634
✔	✔	-	0.573±0.078	**0.688±0.138**	0.337±0.068	1.967±0.311	0.289±0.096	3.317±0.713
✔	✔	✔	**0.579±0.070**	0.698±0.128	**0.403±0.065**	**1.649±0.259**	**0.352±0.080**	**2.953±0.571**

Table 4: Performance comparison of different data augmentation techniques in offline MOMO.

Molecular objectives	GSK3β+JNK3		GSK3β+JNK3+QED		GSK3β+JNK3+QED+SA	
Augmentation	HV(↑)	R2(↓)	HV(↑)	R2(↓)	HV(↑)	R2(↓)
Baseline (REINVENT)	0.462±0.133	0.921±0.259	0.196±0.083	2.646±0.327	0.168±0.046	3.969±0.664
+ Stochastic sampling	0.545±0.063	0.773±0.120	0.319±0.059	1.928±0.314	0.251±0.084	3.504±0.634
+ Crossover operator	0.540±0.088	0.790±0.181	0.367±0.062	1.793±0.245	0.302±0.072	3.110±0.479
+ StitchNet (Ours)	**0.579±0.070**	**0.698±0.128**	**0.403±0.065**	**1.649±0.259**	**0.352±0.080**	**2.953±0.571**

5.2 Ablation Study

To investigate the impact of each key component in our framework—Rank-based Proxy (RP), StitchNet (SN), and Priority Sampling (PS)—we conducted an ablation study, as presented in Table 3.

Effects of rank-based proxy. When RP was ablated and replaced with a score-based proxy, which is similar to the vanilla proxy that directly approximates objective scores, we observed a noticeable drop in performance. This performance drop became more pronounced as the number of objectives increased. Detailed investigations of score- and rank-based proxies are provided in Appendix L. In addition, we extended RP by employing multiple proxies, with the results presented in Appendix M.

Benefits of StitchNet. The ablation study highlighted the significant impact of SN in the offline optimization process. By generating novel stitched molecules, SN provides valuable training samples for fine-tuning the generative model. Importantly, SN incorporates a crossover mechanism similar to that in genetic algorithms but with the added capability of receiving chemical feedback. The efficacy of this crossover operation was validated in our main results, where genetic algorithm-based methods like GeneticGFN and Saturn also demonstrated strong performance. These findings suggest that incorporating the crossover operation, as SN does, is beneficial for offline MOMO because it naturally promotes diversity by exploring novel combinations derived from existing molecules.

Benefits of priority sampling. PS played a crucial role in generating diverse weight configurations, which enabled SN to operate with a wide variety of molecule pairs. In the ablation study, PS had a minimal impact on performance in the two-objective scenario. This is likely because, with only two objectives, the trade-offs are simpler, and the Pareto front can be adequately explored using basic weight configurations. However, as the number of objectives increased to three and four, the benefits of PS became more pronounced. PS significantly improved performance by enabling our framework to efficiently navigate more complex Pareto front through diverse weight configurations.

5.3 Experimental Analysis and Discussion

Data augmentation. In our main results, we observed that employing StitchNet as a data augmentation technique significantly enhanced performance in offline MOMO. To investigate its effectiveness, we compared StitchNet with other data augmentation techniques. One technique is stochastic sampling, where new molecules are stochastically drawn from the generative model's learned distribution. To put it simply, this process can be represented in code-level terms as `model.sample()`. Another technique is the crossover operator, used in GeneticGFN and Saturn, which generates new offspring molecules by combining features from parent molecules in a rule-based manner. As shown in Table 4, all data augmentation techniques outperformed the baseline, underscoring their effectiveness in offline MOMO. Notably, the crossover operator generally demonstrated comparable or better performance than stochastic sampling due to its ability to combine existing high-quality molecules to create diverse and unique offspring. Importantly, StitchNet achieved the best performance across all scenarios, showing its effectiveness by leveraging a neural network to integrate chemical feedback.

Table 5: Performance comparison of various preference optimization techniques in offline MOMO.

Molecular objectives	GSK3β+JNK3		GSK3β+JNK3+QED		GSK3β+JNK3+QED+SA	
Method	HV(↑)	R2(↓)	HV(↑)	R2(↓)	HV(↑)	R2(↓)
Baseline (REINVENT)	0.462±0.133	0.921±0.259	0.196±0.083	2.646±0.327	0.168±0.046	3.969±0.664
+ StitchNet & RLHF	0.561±0.055	0.742±0.098	0.303±0.087	2.012±0.318	0.232±0.071	3.715±0.611
+ StitchNet & DPO	0.557±0.094	0.747±0.174	0.363±0.069	1.843±0.271	0.327±0.081	3.015±0.493
+ StitchNet & IPO	0.552±0.056	0.746±0.106	0.385±0.062	1.755±0.232	0.344±0.082	2.955±0.533
+ MolStitch (Ours)	**0.579**±0.070	**0.698**±0.128	**0.403**±0.065	**1.649**±0.259	**0.352**±0.080	**2.953**±0.571

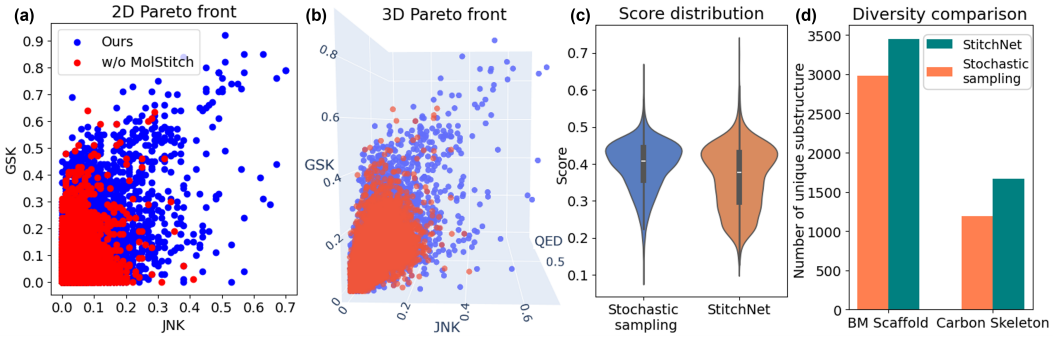


Figure 4: Visualizations of (a-b) the Pareto front, and (c-d) the diversity analysis for StitchNet.

Preference optimization. In our MolStitch framework, we fine-tuned the generative model using a process analogous to the preference optimization techniques employed in large language models. To evaluate different preference optimization techniques in offline MOMO, we explored alternatives such as RLHF (Ouyang et al., 2022), where the proxy model serves as a reward model to generate rewards that are directly optimized. Other approaches involved removing the proxy by allowing the generative model to act as a judge to directly classify winning and losing molecules and update itself using DPO or IPO loss functions. As illustrated in Table 5, our MolStitch consistently outperformed other techniques in all scenarios by constructing the separate proxy model for molecule evaluation and updating the generative model separately based on proxy feedback. This separation has shown to be effective, as supported by recent studies (Singhal et al., 2024; Liu et al., 2024), where maintaining a separate reward-ranking model helps to mitigate distributional shifts and enhance performance.

Pareto front visualization. To evaluate the impact of MolStitch on solution quality, we visualized the Pareto front in both 2D and 3D objective spaces. As depicted in Figure 4 (a-b), the Pareto front obtained from MolStitch dominated the baseline without MolStitch, indicating superior performance across all objectives. Notably, the solutions generated by MolStitch were concentrated in the upper right region of the Pareto front, signifying the effectiveness of molecular stitching in offline MOMO.

Diversity analysis. In offline MOMO, promoting molecular diversity is crucial for identifying candidates with desirable properties while avoiding over-exploration of similar structures. To assess the diversity of augmented molecules generated by StitchNet in comparison to stochastic sampling, we visualized their objective score distributions using violin plots. As shown in Figure 4 (c), StitchNet exhibited a broader and more varied score distribution, demonstrating its capacity to provide a diverse range of augmented molecules for the generative model. We also evaluated the final molecules produced by the generative model fine-tuned with StitchNet against those from stochastic sampling, using diversity metrics that measure the number of unique substructures, specifically Bemis-Murcko (BM) scaffolds and carbon skeletons (Bemis & Murcko, 1996). As depicted in 4 (d), the generative model fine-tuned with StitchNet exhibited greater diversity compared to stochastic sampling across both BM scaffolds and carbon skeletons. Additional diversity analysis is provided in Appendix N.

6 Conclusion

In this study, we propose the Molecular Stitching (MolStitch) framework to tackle the offline multi-objective molecular optimization (MOMO) problem. MolStitch generates novel stitched molecules by combining the desirable properties of both parent molecules in the offline dataset. These stitched molecules serve as valuable training samples for fine-tuning the generative model, thereby enhancing its ability to produce superior molecules beyond the offline dataset. Through extensive experiments, we validate the efficacy of MolStitch in offline MOMO. Future work can be found in Appendix O.

ETHICS STATEMENT

In this study, we address the offline multi-objective molecular optimization problem, which has potential applications in drug discovery. We emphasize the responsible application of our methodologies, with a strong focus on safety considerations. Although our framework enhances the efficiency of molecular optimization, it is crucial that all identified molecules must undergo experimental validation, safety assessments, and regulatory approval before being considered for real-world deployment. We caution against relying solely on computationally generated molecules without proper testing, as it could lead to unintended health or environmental risks. Furthermore, all datasets used in this study are publicly available, and meet ethical standards, ensuring transparency and integrity.

REPRODUCIBILITY STATEMENT

We provide comprehensive information on the experimental settings, workflow, hyperparameters, and implementation details in Appendix H. Additionally, the source code for our proposed framework is available online at `https://tinyurl.com/ycbts7j2`.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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

Joni Alvesalo, Antti Siiskonen, Mikko J Vainio, Päivi SM Tammela, and Pia M Vuorela. Similarity based virtual screening: a tool for targeted library design. *Journal of medicinal chemistry*, 49(7): 2353–2356, 2006.

Dylan M Anstine and Olexandr Isayev. Generative models as an emerging paradigm in the chemical sciences. *Journal of the American Chemical Society*, 145(16):8736–8750, 2023.

Md Atiqur Rahman, Ali Salajegheh, Robert Anthony Smith, King-yin Lam, et al. Braf inhibitor therapy for melanoma, thyroid and colorectal cancers: development of resistance and future prospects. *Current cancer drug targets*, 14(2):128–143, 2014.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455. PMLR, 2024.

V Bagal, R Aggarwal, P Vinod, and UD Priyakumar. Liggpt: Molecular generation using a transformer-decoder model, 2021.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Frédérique Barbosa and Dragos Horvath. Molecular similarity and property similarity. *Current topics in medicinal chemistry*, 4(6):589–600, 2004.

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

Andreas Bender and Robert C Glen. Molecular similarity: a key technique in molecular informatics. *Organic & biomolecular chemistry*, 2(22):3204–3218, 2004.

Eleonore Beurel, Steven F Grieco, and Richard S Jope. Glycogen synthase kinase-3 (gsk3): regulation, actions, and diseases. *Pharmacology & therapeutics*, 148:114–131, 2015.

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

Marie A Bogoyevitch and Bostjan Kobe. Uses for jnk: the many and varied substrates of the c-jun n-terminal kinases. *Microbiology and Molecular Biology Reviews*, 70(4):1061–1095, 2006.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

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.

Daniele Calandriello, Zhaohan Daniel Guo, Remi Munos, Mark Rowland, Yunhao Tang, Bernardo Avila Pires, Pierre Harvey Richemond, Charline Le Lan, Michal Valko, Tianqi Liu, et al. Human alignment of large language models through online preference optimisation. In *International conference on machine learning*. PMLR, 2024.

Souradip Chakraborty, Jiahao Qiu, Hui Yuan, Alec Koppel, Dinesh Manocha, Furong Huang, Amrit Bedi, and Mengdi Wang. Maxmin-rlhf: Alignment with diverse human preferences. In *Forty-first International Conference on Machine Learning*, 2024.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45, 2024.

Can Chen, Yingxue Zhang, Jie Fu, Xue Liu, and Mark Coates. Bidirectional learning for offline infinite-width model-based optimization. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 29454–29467, 2022.

Can Chen, Christopher Beckham, Zixuan Liu, Xue Liu, and Christopher Pal. Parallel-mentoring for offline model-based optimization. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 76619–76636, 2023a.

Can Chen, Yingxue Zhang, Xue Liu, and Mark Coates. Bidirectional learning for offline model-based biological sequence design. In *International Conference on Machine Learning*, pp. 5351–5366. PMLR, 2023b.

Xiwei Cheng, Xiangxin Zhou, Yuwei Yang, Yu Bao, and Quanquan Gu. Decomposed direct preference optimization for structure-based drug design. *arXiv preprint arXiv:2407.13981*, 2024.

Jin-Hee Cho, Yating Wang, Ray Chen, Kevin S Chan, and Ananthram Swami. A survey on modeling and optimizing multi-objective systems. *IEEE Communications Surveys & Tutorials*, 19(3):1867–1901, 2017.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9): 10850–10869, 2023.

Kalyanmoy Deb, Karthik Sindhya, and Jussi Hakanen. Multi-objective optimization. In *Decision sciences*, pp. 161–200. CRC Press, 2016.

Soham Deshmukh, Benjamin Elizalde, Rita Singh, and Huaming Wang. Pengi: an audio language model for audio tasks. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 18090–18108, 2023.

Charles Eigenbrot. Structure, function, and activation of coagulation factor vii. *Current protein and peptide science*, 3(3):287–299, 2002.

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

Klaus B Fink and Manfred Göthert. 5-ht receptor regulation of neurotransmitter release. *Pharmacological reviews*, 59(4):360–417, 2007.

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

Justin Fu and Sergey Levine. Offline model-based optimization via normalized maximum likelihood estimation. In *The Ninth International Conference on Learning Representations*, 2021.

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.

Tianfan Fu, Wenhao Gao, Cao Xiao, Jacob Yasonik, Connor W Coley, and Jimeng Sun. Differentiable scaffolding tree for molecule optimization. *International Conference on Learning Representation (ICLR)*, 2022.

Wenhao Gao, Tianfan Fu, Jimeng Sun, and Connor W Coley. Sample efficiency matters: a benchmark for practical molecular optimization. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 21342–21357, 2022.

Rashid Giniatullin. 5-hydroxytryptamine in migraine: The puzzling role of ionotropic 5-ht3 receptor in the context of established therapeutic effect of metabotropic 5-ht1 subtypes. *British Journal of Pharmacology*, 179(3):400–415, 2022.

Rogelio González-González, Sandra López-Verdín, Jesús Lavalle-Carrasco, Nelly Molina-Frechero, Mario Isiordia-Espinoza, Ramón G Carreón-Burciaga, and Ronell Bologna-Molina. Current concepts in ameloblastoma-targeted therapies in b-raf proto-oncogene serine/threonine kinase v600e mutation: Systematic review. *World journal of clinical oncology*, 11(1):31, 2020.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Siyi Gu, Minkai Xu, Alexander Powers, Weili Nie, Tomas Geffner, Karsten Kreis, Jure Leskovec, Arash Vahdat, and Stefano Ermon. Aligning target-aware molecule diffusion models with exact energy optimization. *arXiv preprint arXiv:2407.01648*, 2024.

Longfei Guan, Hongbin Yang, Yingchun Cai, Lixia Sun, Peiwen Di, Weihua Li, Guixia Liu, and Yun Tang. Admet-score–a comprehensive scoring function for evaluation of chemical drug-likeness. *Medchemcomm*, 10(1):148–157, 2019.

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: Sample-efficient generative molecular design with reinforcement learning. *JACS Au*, 2024a.

Jeff Guo and Philippe Schwaller. Saturn: Sample-efficient generative molecular design using memory manipulation. *arXiv preprint arXiv:2405.17066*, 2024b.

Charles A Hall, Samuel I Rapaport, Sara B Ames, Jean A DeGroot, Edward S Allen, and Marc A Ralston. A clinical and family study of hereditary proconvertin (factor vii) deficiency. *The American Journal of Medicine*, 37(2):172–181, 1964.

Jiazhen He, Eva Nittinger, Christian Tyrchan, Werngard Czechtizky, Atanas Patronov, Esben Jannik Bjerrum, and Ola Engkvist. Transformer-based molecular optimization beyond matched molecular pairs. *Journal of cheminformatics*, 14(1):18, 2022.

Jian Hu, Xibin Wu, Weixun Wang, Dehao Zhang, Yu Cao, et al. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*, 2024a.

Xiangkun Hu, Tong He, and David Wipf. New desiderata for direct preference optimization. *arXiv preprint arXiv:2407.09072*, 2024b.

Moksh Jain, Emmanuel Bengio, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Bonaventure FP Dossou, Chanakya Ajit Ekbote, Jie Fu, Tianyu Zhang, Michael Kilgour, Dinghuai Zhang, et al. Biological sequence design with gflownets. In *International Conference on Machine Learning*, pp. 9786–9801. PMLR, 2022.

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.

José Jiménez-Luna, Francesca Grisoni, Nils Weskamp, and Gisbert Schneider. Artificial intelligence in drug discovery: recent advances and future perspectives. *Expert opinion on drug discovery*, 16 (9):949–959, 2021.

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.

Richard S Jope, Christopher J Yuskaitis, and Eléonore Beurel. Glycogen synthase kinase-3 (gsk3): inflammation, diseases, and therapeutics. *Neurochemical research*, 32:577–595, 2007.

Stefan Kamphausen, Nils Höltge, Frank Wirsching, Corinna Morys-Wortmann, Daniel Riester, Ruediger Goetz, Marcel Thürk, and Andreas Schwienhorst. Genetic algorithm for the design of molecules with desired properties. *Journal of Computer-Aided Molecular Design*, 16:551–567, 2002.

Hyeonah Kim, Minsu Kim, Sanghyeok Choi, and Jinkyoo Park. Genetic-guided gflownets: Advancing in practical molecular optimization benchmark. *arXiv preprint arXiv:2402.05961*, 2024a.

Minsu Kim, Federico Berto, Sungsoo Ahn, and Jinkyoo Park. Bootstrapped training of score-conditioned generator for offline design of biological sequences. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 67643–67661, 2023.

Sungyoon Kim, Yunseon Choi, Daiki E Matsunaga, and Kee-Eung Kim. Stitching sub-trajectories with conditional diffusion model for goal-conditioned offline rl. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13160–13167, 2024b.

J-M Launay, P Herve, K Peoc'h, C Tournois, J Callebert, Canan G Nebigil, N Etienne, L Drouet, M Humbert, G Simonneau, et al. Function of the serotonin 5-hydroxytryptamine 2b receptor in pulmonary hypertension. *Nature medicine*, 8(10):1129–1135, 2002.

Seul Lee, Jaehyeong Jo, and Sung Ju Hwang. Exploring chemical space with score-based out-of-distribution generation. In *International Conference on Machine Learning*, pp. 18872–18892. PMLR, 2023.

Guanghe Li, Yixiang Shan, Zhengbang Zhu, Ting Long, and Weinan Zhang. Diffstitch: Boosting offline reinforcement learning with diffusion-based trajectory stitching. In *International Conference on Machine Learning*, 2024.

Shuiqiao Liu, Weibo Luo, and Yingfei Wang. Emerging role of parp-1 and parthanatos in ischemic stroke. *Journal of neurochemistry*, 160(1):74–87, 2022.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. Statistical rejection sampling improves preference optimization. In *The Twelfth International Conference on Learning Representations*, 2024.

Simon Loiodice, Andre Nogueira da Costa, and Franck Atienzar. Current trends in in silico, in vitro toxicology, and safety biomarkers in early drug development. *Drug and chemical toxicology*, 42 (2):113–121, 2019.

Gerald Maggiora, Martin Vogt, Dagmar Stumpfe, and Jurgen Bajorath. Molecular similarity in medicinal chemistry: miniperspective. *Journal of medicinal chemistry*, 57(8):3186–3204, 2014.

Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in gflownets. *Advances in Neural Information Processing Systems*, 35:5955–5967, 2022.

R Timothy Marler and Jasbir S Arora. The weighted sum method for multi-objective optimization: new insights. *Structural and multidisciplinary optimization*, 41:853–862, 2010.

Thomas Minka. Estimating a dirichlet distribution, 2000.

In Jae Myung. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*, 47(1):90–100, 2003.

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

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 27730–27744, 2022.

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *arXiv preprint arXiv:2402.13228*, 2024.

Ryan Park, Ryan Theisen, Navriti Sahni, Marcel Patek, Anna Cichońska, and Rayees Rahman. Preference optimization for molecular language models. *arXiv preprint arXiv:2310.12304*, 2023.

Alexis Payton, Kyle R Roell, Meghan E Rebuli, William Valdar, Ilona Jaspers, and Julia E Rager. Navigating the bridge between wet and dry lab toxicology research to address current challenges with high-dimensional data. *Frontiers in Toxicology*, 5:1171175, 2023.

Han Qi, Yi Su, Aviral Kumar, and Sergey Levine. Data-driven offline decision-making via invariant representation learning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 13226–13237, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: your language model is secretly a reward model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 53728–53741, 2023.

Arnab Ray Chaudhuri and André Nussenzweig. The multifaceted roles of parp1 in dna repair and chromatin remodelling. *Nature reviews Molecular cell biology*, 18(10):610–621, 2017.

Lynn Resnick and Myles Fennell. Targeting jnk3 for the treatment of neurodegenerative disorders. *Drug discovery today*, 9(21):932–939, 2004.

Maisha T Robinson, Alejandro A Rabinstein, James F Meschia, and William D Freeman. Safety of recombinant activated factor vii in patients with warfarin-associated hemorrhages of the central nervous system. *Stroke*, 41(7):1459–1463, 2010.

Lorenzo Rosasco, Ernesto De Vito, Andrea Caponnetto, Michele Piana, and Alessandro Verri. Are loss functions all the same? *Neural computation*, 16(5):1063–1076, 2004.

Michèle Rouleau, Anand Patel, Michael J Hendzel, Scott H Kaufmann, and Guy G Poirier. Parp inhibition: Parp1 and beyond. *Nature reviews cancer*, 10(4):293–301, 2010.

Amélie Royer, Tijmen Blankevoort, and Babak Ehteshami Bejnordi. Scalarization for multi-task and multi-domain learning at scale. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 16917–16941, 2023.

E Sanz-Garcia, G Argiles, E Elez, and J Tabernero. Braf mutant colorectal cancer: prognosis, treatment, and new perspectives. *Annals of Oncology*, 28(11):2648–2657, 2017.

Matthew M Seavey and Pawel Dobrzanski. The many faces of janus kinase. *Biochemical pharmacology*, 83(9):1136–1145, 2012.

Emilee Senkevitch and Scott Durum. The promise of janus kinase inhibitors in the treatment of hematological malignancies. *Cytokine*, 98:33–41, 2017.

Dong-Hee Shin, Young-Han Son, Deok-Joong Lee, Ji-Wung Han, and Tae-Eui Kam. Dynamic many-objective molecular optimization: Unfolding complexity with objective decomposition and progressive optimization. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 6026–6034, 2024. URL https://doi.org/10.24963/ijcai.2024/666.

Prasann Singhal, Nathan Lambert, Scott Niekum, Tanya Goyal, and Greg Durrett. D2po: Discriminator-guided dpo with response evaluation models. *arXiv preprint arXiv:2405.01511*, 2024.

Young-Han Son, Dong-Hee Shin, and Tae-Eui Kam. FTMMR: Fusion transformer for integrating multiple molecular representations. *IEEE Journal of Biomedical and Health Informatics*, 2024.

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.

Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Rémi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Ávila Pires, and Bilal Piot. Generalized preference optimization: A unified approach to offline alignment. In *International conference on machine learning*. PMLR, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Brandon Trabucco, Aviral Kumar, Xinyang Geng, and Sergey Levine. Conservative objective models for effective offline model-based optimization. In *International Conference on Machine Learning*, pp. 10358–10368. PMLR, 2021.

Brandon Trabucco, Xinyang Geng, Aviral Kumar, and Sergey Levine. Design-bench: Benchmarks for data-driven offline model-based optimization. In *International Conference on Machine Learning*, pp. 21658–21676. PMLR, 2022.

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.

David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

Yutong Xie, Chence Shi, Hao Zhou, Yuwei Yang, Weinan Zhang, Yong Yu, and Lei Li. MARS: Markov molecular sampling for multi-objective drug discovery. In *The Ninth International Conference on Learning Representations*, 2021a.

Yutong Xie, Chence Shi, Hao Zhou, Yuwei Yang, Weinan Zhang, Yong Yu, and Lei Li. Mars: Markov molecular sampling for multi-objective drug discovery. In *International Conference on Learning Representations*, 2021b.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint. In *International Conference on Machine Learning*. PMLR, 2024.

Ke Xue, Rongxi Tan, Xiaobin Huang, and Chao Qian. Offline multi-objective optimization. In *International conference on machine learning*. PMLR, 2024.

Kunihiro Yamaoka, Pipsa Saharinen, Marko Pesu, Vance ET Holt, Olli Silvennoinen, and John J O'Shea. The janus kinases (jaks). *Genome biology*, 5:1–6, 2004.

Sihyun Yu, Sungsoo Ahn, Le Song, and Jinwoo Shin. Roma: robust model adaptation for offline model-based optimization. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, pp. 4619–4631, 2021.

Ye Yuan, Can Chen, Zixuan Liu, Willie Neiswanger, and Xue Liu. Importance-aware co-teaching for offline model-based optimization. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 55718–55733, 2023.

Mohd Yusuf. Insights into the in-silico research: current scenario, advantages, limits, and future perspectives. *Life in Silico*, 1(1):13–25, 2023.

Jiahui Zhang, Jin Zhang, Hua Li, Lixia Chen, and Dahong Yao. Dual-target inhibitors of parp1 in cancer therapy: A drug discovery perspective. *Drug Discovery Today*, 28(7):103607, 2023.

Richard Zhang and Daniel Golovin. Random hypervolume scalarizations for provable multi-objective black box optimization. In *International conference on machine learning*, pp. 11096–11105. PMLR, 2020.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning*, pp. 928–936, 2003.

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.

Contents

A Online and Offline settings for Molecular Discovery **19**

B Related Work **20**
B.1 Generative Models for Molecular Discovery 20
B.2 Multi-Objective Molecular Optimization 20
B.3 Offline Model-based Optimization 21
B.4 Preference Optimization in Generative Models 22

C Pre-Training Process for the Generative Model **23**

D Preference Optimization Techniques for the Generative Model **24**
D.1 From Initial Loss Formulation to DPO-like Loss Formulation 24
D.2 IPO-like Loss Formulation 24

E Self-Supervised Training Process for StitchNet **26**

F Priority Sampling Process for StitchNet **27**

G Pseudo-Code **28**

H Experimental Details **28**
H.1 Experimental Settings and Configurations 29
H.2 Experimental Workflow for Offline Molecular Optimization 29
H.3 Descriptions of the Molecular Objectives 31
H.4 Hyperparameters and Implementation Details 33

I Competing Methods Details **34**

J Details on Evaluation Metrics **36**
J.1 Hypervolume Indicator 37
J.2 R2 Indicator 37

K Additional Results **38**
K.1 Evaluating Molecular Optimization Methods Using Average Property Score (APS) of Top 10 and Top 100 Molecules 38
K.2 R2 Performance for the Docking Score Optimization Task 39
K.3 Semi-offline Optimization 39
K.4 Evaluating Mamba and GFlowNets as Additional Backbone Models 40

L Detailed Analysis of Rank-based Proxy **41**

M Additional Experiments on Multiple Proxies **43**

N Additional Analysis on Molecular Diversity of StitchNet **44**

O Future Work and Limitations **46**

P Quantitative Assessment of StitchNet's Ability to Learn Crossover Operations **46**

Q Effectiveness and contribution of StitchNet within our framework **46**

R Reward hacking problem in Multi-objective Optimization **47**

S Molecule Examples **48**

APPENDIX

A ONLINE AND OFFLINE SETTINGS FOR MOLECULAR DISCOVERY

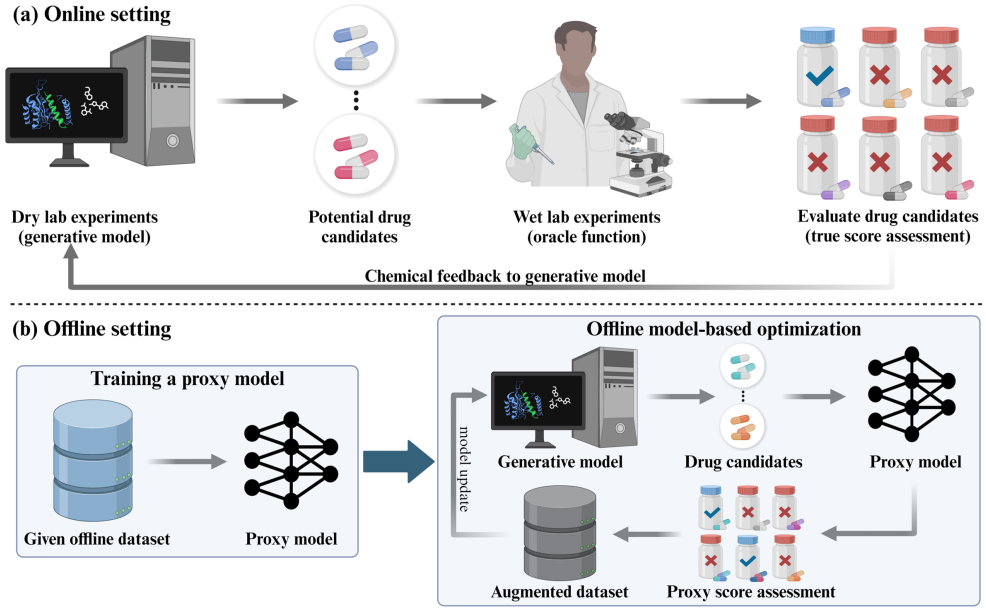


Figure 5: An illustration of online and offline settings for drug discovery. (a) Online setting: potential drug candidates from the generative model are evaluated directly through wet lab experiments (oracle function). (b) Offline setting: a proxy model approximates the oracle function using offline dataset, guiding the generative model to produce better drug candidates through iterative updates.

In this section, we delve into the detailed pipeline for online and offline settings in molecular discovery, using the specific case of *in silico* drug discovery combined with real-world wet lab experiments as an illustrative example. We further highlight the advantages of implementing the offline setting.

Online setting. Traditionally, many *in silico* drug discovery methods have been based on the assumptions of the online setting (Jiménez-Luna et al., 2021). As depicted in Figure 5 (a), the online setting begins in the computational or 'dry' lab, where a generative model produces potential drug candidates that are predicted to be potent. Researchers then select the top K drug candidates or apply specific filters to choose which drug candidates to advance. These selected drugs are sent to the wet lab, where they undergo physical biological experiments to validate their efficacy. Note that these wet lab experiments serve as a true oracle function, which provides accurate assessments of drug potency and properties based on real-world testing. Once the wet lab experiments are complete, the results—true score assessments—are sent back to the dry lab as chemical feedback. This feedback is then used to update the generative model, enabling it to produce more desirable and potential drug candidates in the next iteration. This iterative process continues until successful drug candidates are identified or predefined criteria, such as reaching a certain optimization score, are met.

Why offline setting? The main advantage of the online setting is its ability to continuously refine the generative model using feedback from the true oracle function. However, this feedback relies on real-world wet lab experiments, which are typically time-consuming and costly. Therefore, querying the true oracle function for every drug candidate is often impractical, and safety concerns can further limit its use (Loiodice et al., 2019; Yusuf, 2023). Even if we assume these challenges are mitigated and resources are available to query the true oracle function as needed, there still remains the challenge of a significant time mismatch between the dry lab and the wet lab. The dry lab can generate new drug candidates within hours, but the wet lab evaluation—including chemical synthesis, purification, and biological testing—can take weeks or even months (Payton et al., 2023). This significant lag means that while the wet lab is engaged in lengthy experiments, the dry lab may be left idle, which is inefficient. To address these limitations, the offline setting has gained considerable attention in recent years (Xue et al., 2024). In the offline setting, the generative model can be trained using existing offline datasets without relying on continuous feedback from wet lab experiments.

Offline setting. One of the most prevalent and widely adopted approaches for handling the offline setting is offline model-based optimization (MBO). As illustrated in Figure 5 (b), the process begins by training a proxy model on the given offline dataset. This proxy model serves as a surrogate for evaluating drug candidates, as access to the true oracle function is not available in the offline setting. Once the proxy model is trained, the offline MBO process is initiated to enable the training of the generative model without relying on real-world wet lab feedback. Specifically, the generative model produces new drug candidates, which are evaluated by the proxy model instead of being sent to the wet lab. The proxy model provides estimated proxy scores for these candidates, and these pairs of drug candidates and their proxy scores are stored in a buffer to create an augmented dataset. This augmented dataset is then utilized to update the generative model via gradient ascent, leveraging the proxy model's predictions. This iterative cycle continues until predefined criteria are met.

B Related Work

B.1 Generative Models for Molecular Discovery

The rapid advancement of generative models has profoundly impacted various fields, including computer vision (Croitoru et al., 2023), natural language processing (Chang et al., 2024), and audio signal processing (Deshmukh et al., 2023). This progress has extended to molecular discovery (Anstine & Isayev, 2023; Son et al., 2024), where generative models have demonstrated their capacity to generate and optimize molecules towards promising regions of the chemical search space. Several types of generative models have been employed in molecular discovery.

Genetic algorithms (GAs). Inspired by natural evolution, GAs maintain a population of candidate solutions and iteratively improve them based on a predefined fitness function. In particular, GAs employ selection, crossover, mutation, and replacement operations to improve the overall quality of the population. In the context of molecular discovery, GraphGA (Jensen, 2019) has demonstrated notable success in generating promising molecules by navigating the chemical space effectively.

Sampling-based methods. These methods leverage advanced sampling techniques to draw samples from distributions that are likely to yield desirable molecular properties. MARS (Xie et al., 2021a) is a notable example that employs Markov Chain Monte Carlo (MCMC) sampling to efficiently search for high-quality molecules. By focusing on probabilistic sampling, these methods can explore the chemical space more efficiently than deterministic approaches.

Reinforcement learning (RL). RL-based methods formulate the molecule generation process as a Markov decision process, allowing an RL agent to interact with a chemical environment to construct molecular structures in an autoregressive manner. A prominent example is REINVENT (Olivecrona et al., 2017), which utilizes a GRU model (Chung et al., 2014) as its RL agent to generate molecules in SMILES format. REINVENT has been acknowledged as one of the best models for the molecular property optimization task, showcasing the effectiveness of RL-based methods. Following its success, several variants have been proposed to enhance its capabilities. One line of research focuses on improving the underlying neural architecture by replacing the GRU with either a transformer (He et al., 2022) or Mamba (Gu & Dao, 2023). Another approach incorporates data augmentation techniques to boost sample efficiency, leading to methods like Augmented Memory (AugMem) (Guo & Schwaller, 2024a), which achieved new state-of-the-art performance.

GFlowNets. While RL-based methods have shown effectiveness, they often struggle with maintaining diversity in the generated molecules due to a tendency to exploit a single promising direction. GFlowNets (Jain et al., 2022) aims to address this limitation by emphasizing probabilistic sampling over reward maximization, inherently promoting diversity in the generated molecules. As a result, GFlowNets have gained popularity in multi-objective molecular optimization tasks, where generating a diverse set of high-quality molecules across multiple objectives is crucial.

B.2 Multi-Objective Molecular Optimization

The multi-objective molecular optimization (MOMO) problem differs from single-objective optimization by requiring the simultaneous optimization of multiple molecular properties, which often conflict with one another. In the context of the MOMO problem, identifying a single solution that optimally satisfies all objectives is generally infeasible. Instead, the goal shifts to discovering a

diverse set of Pareto optimal molecules, where improving one objective may lead to trade-offs in others. To tackle multiple objectives, several studies have integrated Bayesian optimization (BO) within their molecular optimization frameworks. For instance, GPBO and REINVENT-BO (Tripp et al., 2021) incorporate BO into GraphGA and REINVENT, respectively, resulting in enhanced sample efficiency. In a similar approach, LamBO (Stanton et al., 2022) applies BO alongside denoising autoencoders to address the multi-objective biological sequence design problem. Other studies have employed scalarization, which simplifies the multi-objective problem by converting multiple objectives into a single scalar objective function (Gunantara, 2018). This scalarization is typically achieved by combining the objectives using a weighted sum or other aggregation techniques (Marler & Arora, 2010; Deb et al., 2016). Scalarization offers simplicity and ease of implementation, making it a popular choice for its scalability and computational efficiency (Cho et al., 2017). In the context of the MOMO problem, MIMOSA (Fu et al., 2021) utilizes linear scalarization to efficiently manage the complexity of multiple objectives, while demonstrating strong performance and scalability. Similarly, MARS (Xie et al., 2021a) applies scalarization to effectively handle up to four molecular objectives, further showcasing the potential of scalarization in the MOMO problem. However, scalarization presents challenges in selecting appropriate weights. Users must assign weights to each objective to reflect its relative importance, a process that is often sensitive and subjective (Royer et al., 2023). Incorrect or biased weight selection may fail to accurately represent true preferences, potentially resulting in suboptimal solutions (Zhang & Golovin, 2020). In our study, we also employ the scalarization approach due to its widespread adoption and practical advantages (Fromer & Coley, 2023). However, to mitigate the limitations associated with subjective weight selection, we introduce priority sampling using the Dirichlet distribution to generate a diverse set of weight configurations. This enables our StitchNet to operate on a wide variety of molecular pairs, each representing a different balance of multiple objectives.

B.3 Offline Model-based Optimization

As mentioned earlier in the main manuscript, one of the most promising approaches for addressing the offline MOMO problem is offline model-based optimization (MBO) (Trabucco et al., 2022). The goal of offline MBO is to optimize the objective function using a pre-collected offline dataset, without the ability to acquire new data during the optimization process. In this approach, the proxy (surrogate) model—such as Gaussian processes, random forests, or neural networks—is trained on the offline dataset to approximate the objective function. This proxy model is then used to predict objective scores for new inputs, guiding the optimization algorithm in finding inputs that maximize the predicted objective scores. The most straightforward approach in offline MBO is to use a differentiable vanilla proxy model and apply gradient ascent to find optimal inputs. However, this approach may face limitations, such as increased inaccuracies as problem complexity grows and a higher risk of overfitting. To address these limitations, various recent studies have been proposed.

Improving the proxy model. One line of research focuses on enhancing the accuracy and robustness of the proxy model to better handle high-dimensional and complex objective functions. Some studies (Trabucco et al., 2021; Qi et al., 2022) enforce constraints to mitigate overfitting and address distributional shifts caused by out-of-distribution (OOD) inputs, while another study (Yu et al., 2021) enhances the generalization capabilities of the proxy model by employing a local smoothness prior.

Improving optimization algorithms. Another line of research concentrates on improving the optimization algorithms used within the offline MBO framework. For example, the bidirectional learning technique (Chen et al., 2022; 2023b) has been introduced to utilize both forward and backward mappings to generate input configurations that are likely to produce optimal outputs while adhering to the data distribution of the offline dataset. Additionally, the bootstrapping technique (Kim et al., 2023) has been developed to enhance the optimization process by iteratively augmenting the offline dataset with self-generated data, using the proxy model as a pseudo-labeler.

Ensemble learning. To leverage the benefits of ensemble learning, several studies (Trabucco et al., 2022; Yuan et al., 2023; Chen et al., 2023a) have proposed to utilize multiple proxy models to combine their predictions, thereby enhancing the robustness and reliability of the optimization process. Notably, Tri-Mentoring (Chen et al., 2023a) not only employs ensemble learning but also shifts its focus to generating pairwise comparison labels rather than directly approximating objective scores. Our proposed proxy model is similar to Tri-Mentoring, as it reformulates the task from direct property score regression to pairwise classification.

B.4 PREFERENCE OPTIMIZATION IN GENERATIVE MODELS

In recent years, preference optimization has gained significant attention, particularly with the rise of large language models and generative models (Tang et al., 2024). As these models grow more powerful and are deployed into real-world applications, the need to align their outputs with human expectations becomes increasingly important. Preference optimization enables models to better align with human standards in subjective areas such as sentiment, creativity, and ethical considerations.

Reinforcement Learning from Human Feedback (RLHF). A leading and widely adopted method for incorporating human preferences into model training is RLHF (Ouyang et al., 2022). By embedding human feedback within an RL framework, RLHF allows models to generate higher-quality content that aligns more closely with human judgments. Notable implementations like OpenAI's ChatGPT (Achiam et al., 2023) have demonstrated significant performance improvements through RLHF, highlighting its potential in fine-tuning models. This success has driven further research into more streamlined approaches that aim to simplify the incorporation of human preferences.

Direct Preference Optimization (DPO). DPO (Rafailov et al., 2023) is a recent method that moves away from RL and focuses directly on optimizing for human preferences without the need for reward modeling. It operates by directly training on human preference pairs, enabling the model to generate outputs that are consistently favored over less preferred alternatives. This approach is considered more straightforward and potentially more stable than RLHF, as it bypasses the complexities associated with RL training. However, DPO has exhibited limitations, particularly in scenarios involving deterministic preferences, due to its relatively weak regularization mechanisms.

Identity Preference Optimization (IPO). IPO (Azar et al., 2024) is a more recent method that builds on DPO by introducing enhancements to address its limitations and offering a more theoretically sound framework. Specifically, IPO incorporates a stronger regularization term that penalizes models for excessive confidence in preference margins. This is achieved by replacing the log-sigmoid function used in DPO with a squared loss function. The stronger regularization term in IPO aims to balance adaptation to the preference dataset while maintaining generalization capabilities, which is crucial for model performance on out-of-distribution (OOD) data. While IPO offers theoretical improvements over DPO, empirical results have been mixed. Some studies report IPO performing on par with or slightly better than DPO (Pal et al., 2024; Calandriello et al., 2024), while others observe diminished performance in certain settings (Hu et al., 2024b).

Preference optimization in molecular discovery. In large language models, preference typically reflects human sentiments, opinions, or judgments about what constitutes a desirable output. On the other hand, in the field of molecular discovery, preference represents the relative importance of each objective within the optimization process. When the generative model is tasked with optimizing several conflicting objectives, preference guides the optimization process by specifying how much weight or priority each objective should be given. For example, if a researcher wants to prioritize potency over safety, their preferences would assign more importance to optimizing potency. Conversely, if safety is more critical, the preference would shift toward that objective. Recently, preference optimization has been widely adopted in structure-based drug design to align the pretrained generative model with preferred functional properties (Park et al., 2023; Cheng et al., 2024; Gu et al., 2024). Our work also focuses on optimizing molecules with desired properties. However, unlike recent studies (Park et al., 2023; Cheng et al., 2024; Gu et al., 2024) that primarily use DPO and rely on existing preference datasets, our approach differs in several key ways. We explore a variety of preference optimization techniques—including RLHF, DPO, and IPO—and apply them to the offline multi-objective molecular optimization problem. More importantly, we generate a new preference dataset using our StitchNet model, which creates novel stitched molecules with desirable properties from pairs of existing molecules. In other words, rather than depending solely on existing datasets for preferences, we construct a separate proxy model and use StitchNet to build a tailored preference dataset, leveraging existing molecules to further enhance the optimization process. Additionally, we extend our approach to a semi-offline setting—a direction that recent studies have not explored yet. In this setting, we utilize a limited number of online evaluations by periodically querying an oracle function to assess molecules in large batches. This extension allows us to explore ways of further enhancing the optimization process by integrating new evaluation data into our model.

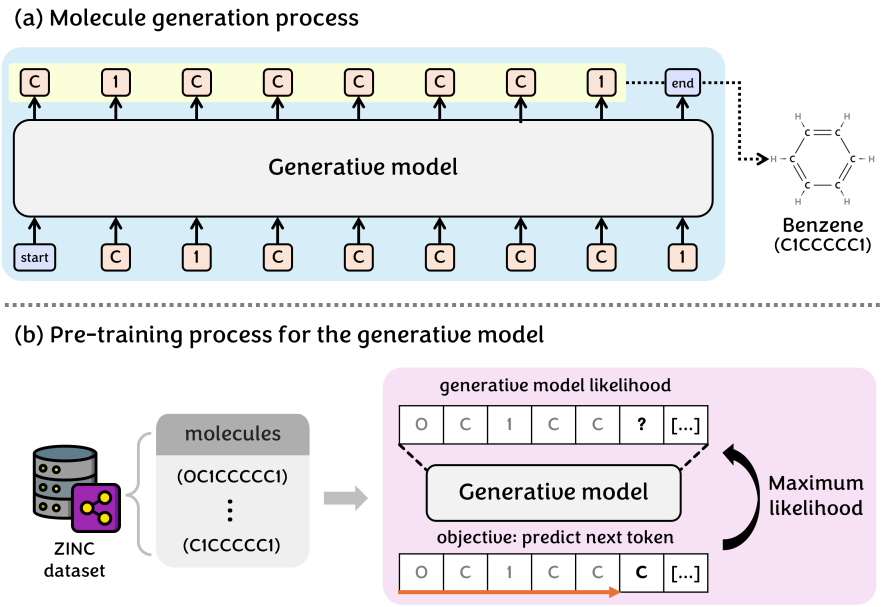


Figure 6: (a) The generative model produces molecules in SMILES format using an auto-regressive approach. (b) During the pre-training stage, molecules from the ZINC dataset are used as ground truth labels. The generative model is then updated through maximum likelihood approach to maximize the probability of the correct next molecular token (component) given the preceding sequence.

C Pre-Training Process for the Generative Model

In this section, we present an illustration of the molecule generation process and describe the pre-training process for the generative model. As depicted in Figure 6 (a), which exemplifies the generation of a benzene molecule, the generative model produces molecules in an auto-regressive manner, similar to how language models generate sentences sequentially. Specifically, the generative model produces molecules in SMILES format, where each token corresponds to an atom or bond. The generation process begins with an initial token, and the model predicts the subsequent token based on the previously generated sequence, continuing this process until the complete molecule is formed.

Moving on to the pre-training process for the generative model, we employ an approach analogous to the next-token prediction loss used in language model training, as shown in Figure 6 (b). Specifically, the model is trained using the maximum likelihood approach, where molecules sampled from the ZINC dataset serve as ground truth labels. The objective of this pre-training process is to maximize the likelihood of accurately predicting the next molecular token (component) based on the preceding sequence. The cross-entropy loss is employed to measure the difference between the predicted probability distribution and the true distribution of the next token, guiding the model to learn the correct sequence of molecular components and generate chemically valid molecules.

Building upon the pre-training of our generative model using the ZINC dataset, we now detail the specific generative model employed in our framework. As mentioned in the main manuscript, REINVENT was selected as our main generative model due to its widespread adoption and proven effectiveness in various molecular optimization tasks. In REINVENT, the molecule optimization process is formulated as a Markov decision process, utilizing the RL algorithm to generate molecules based on a given scoring (reward) function. The training architecture of REINVENT comprises two distinct policy models: the prior model and the agent model. The prior model, denoted as G_{ref}, is a pre-trained reference model that encodes chemical grammar to ensure the chemical validity of the generated molecules, as depicted in Figure 6 (b). The agent model G_ϕ is initialized from the prior model and serves as the main policy that aims to maximize the reward score associated with the desired molecular properties, while not deviating too far from the prior model. The training objective for the agent model can be defined as:

$$\mathcal{L}_{\text{agent}}(\phi) = \mathbb{E}_{m\sim D}\left[\left(-\log G_\phi(m) + \log G_{\text{ref}}(m) + R(m)\right)^2\right],$$

where $R(m)$ represents the reward score for molecule m within the offline dataset $\mathcal{D}$. Note that this work addresses the offline MOMO problem, where the offline dataset comprises pairs of molecules and their corresponding property (objective) scores. Therefore, these property scores can be used as reward scores for training the agent model. To sum up, this loss function $\mathcal{L}_{\text{agent}}(\phi)$ guides $G_\phi(m)$ to maximize the reward $R(m)$ while aligning with $G_{\text{ref}}(m)$. For a detailed derivation and background of this REINVENT loss function, please refer to prior studies (Olivecrona et al., 2017; Guo & Schwaller, 2024a). After completing the initial training phase on the offline dataset, the agent model is fine-tuned to further enhance its performance beyond the constraints of the offline dataset. This fine-tuning process involves optimizing the agent model with stitched molecules using preference optimization techniques, as described in Equation 15 of the main manuscript.

D Preference Optimization Techniques for the Generative Model

D.1 From Initial Loss Formulation to DPO-like Loss Formulation

As mentioned in Subsection 4.3, the initial loss formulation for the generative model is as follows:

$$\mathcal{L}_{\text{gen}}(\phi) = -\mathbb{E}_{(\bar{m}_w,\bar{m}_l)\sim\mathcal{B}}\left[\log G_\phi(\bar{m}_w) - \log G_\phi(\bar{m}_l)\right] + \beta \cdot \mathbb{D}_{\text{KL}}(G_\phi \| G_{\text{ref}}).$$

This loss equation consists of two key components: the first term represents the difference in log-likelihoods between generating the winning molecule $G_\phi(\bar{m}_w)$ and the losing molecule $G_\phi(\bar{m}_l)$, while the second part introduces a KL divergence between the current generative model G_ϕ and the reference model G_{ref}. Following Tang et al. (2024), the KL divergence term can be defined as:

$$\mathbb{D}_{\text{KL}}(G_\phi \| G_{\text{ref}}) \coloneqq \mathbb{E}_{(\bar{m})\sim\mathcal{B}}\left[\log \frac{G_\phi(\bar{m})}{G_{\text{ref}}(\bar{m})}\right].$$

Since we are focusing on a pairwise comparison between winning and losing molecules, $(\bar{m}_w, \bar{m}_l)$, it is possible to apply the KL divergence to each component and simply the loss function as follows:

$$\mathcal{L}_{\text{gen}}(\phi) \coloneqq -\mathbb{E}_{(\bar{m}_w,\bar{m}_l)\sim\mathcal{B}}\left[\beta\left(\log \frac{G_\phi(\bar{m}_w)}{G_{\text{ref}}(\bar{m}_w)} - \log \frac{G_\phi(\bar{m}_l)}{G_{\text{ref}}(\bar{m}_l)}\right)\right].$$

At this point, we can leverage the notion of the Bradley-Terry model that the log odds of one item winning over another (in our case, m_w over m_l) can also be written as:

$$\log \frac{G_\phi(\bar{m}_w)}{G_\phi(\bar{m}_l)},$$

and this log-odds can be converted into a probability using the sigmoid function $\sigma(\cdot)$, defined as:

$$\sigma(x) = \frac{1}{1+e^{-x}}.$$

To incorporate the probabilistic nature of the comparison, we can now apply the sigmoid function to a combination of the two log-odds from G_ϕ and G_{ref} as follows:

$$\sigma\left[\beta\left(\log \frac{G_\phi(\bar{m}_w)}{G_{\text{ref}}(\bar{m}_w)} - \log \frac{G_\phi(\bar{m}_l)}{G_{\text{ref}}(\bar{m}_l)}\right)\right].$$

Finally, the initial formulation can be re-organized into the following compact DPO-like form:

$$\mathcal{L}_{\text{gen-dpo}}(\phi) = -\mathbb{E}_{(\bar{m}_w,\bar{m}_l)\sim\mathcal{B}}\left[\log\sigma\left(\beta\log \frac{G_\phi(\bar{m}_w)}{G_{\text{ref}}(\bar{m}_w)} - \beta\log \frac{G_\phi(\bar{m}_l)}{G_{\text{ref}}(\bar{m}_l)}\right)\right].$$

D.2 IPO-like Loss Formulation

Building on the methodology presented in Tang et al. (2024), we can represent the DPO-like loss formulation in a more generalized form as follows:

$$\mathcal{L}_{\text{gen-dpo}}(\phi) = -\mathbb{E}_{(\bar{m}_w,\bar{m}_l)\sim\mathcal{B}}\left[\mathcal{F}\left(\beta\left(\log \frac{G_\phi(\bar{m}_w)}{G_{\text{ref}}(\bar{m}_w)} - \log \frac{G_\phi(\bar{m}_l)}{G_{\text{ref}}(\bar{m}_l)}\right)\right)\right],$$

where $\mathcal{F}$ is a scalar function $\mathcal{F} : \mathbb{R} \to \mathbb{R}$ that map input values to scalar outputs. In the case of DPO, $\mathcal{F}$ is typically chosen to be the log-sigmoid function. However, DPO can encounter difficulties when preferences are deterministic. For example, if the probability of m_w defeating m_l is exactly 1, indicating deterministic preference, the difference between them becomes unbounded and approaches toward infinity such as follows:

$$\left(\frac{G_\phi(\bar{m}_w)}{G_{\text{ref}}(\bar{m}_w)} \gg \frac{G_\phi(\bar{m}_l)}{G_{\text{ref}}(\bar{m}_l)}\right) \implies \left(\log \frac{G_\phi(\bar{m}_w)}{G_{\text{ref}}(\bar{m}_w)} - \log \frac{G_\phi(\bar{m}_l)}{G_{\text{ref}}(\bar{m}_l)}\right) \to +\infty.$$

Assuming that β is a positive real number, the term inside the log-sigmoid function becomes infinite, leading to:

$$\log \sigma\left(\beta\left(\log \frac{G_\phi(\bar{m}_w)}{G_{\text{ref}}(\bar{m}_w)} - \log \frac{G_\phi(\bar{m}_l)}{G_{\text{ref}}(\bar{m}_l)}\right)\right) \to \log \sigma(+\infty).$$

Since $\sigma(+\infty) = 1$, it follows that:

$$\log \sigma(+\infty) = \log(1) = 0.$$

Therefore, when preferences are deterministic, the loss function converges to 0 for any value of β. In other words, the regularization term β becomes irrelevant and does not play any role in such cases.

To address these challenges, IPO introduces a stronger regularization term that penalizes models for exhibiting excessive confidence in preference margins. Specifically, IPO replaces the log-sigmoid function used in DPO with a squared loss function (Tang et al., 2024). The quadratic nature of the squared loss penalizes large deviations more heavily, discouraging the model from generating extreme outputs (Rosasco et al., 2004). In deterministic preference cases, the squared loss establishes the boundary to prevent the loss function from converging to 0 for any value of β (Azar et al., 2024).

Recall that we can express the IPO-like loss formulation as follows:

$$\mathcal{L}_{\text{gen-ipo}}(\phi) = -\mathbb{E}_{(\bar{m}_w, \bar{m}_l)\sim\mathcal{B}}\left[\left(\log\left(\frac{G_\phi(\bar{m}_w)}{G_\phi(\bar{m}_l)} \cdot \frac{G_{\text{ref}}(\bar{m}_l)}{G_{\text{ref}}(\bar{m}_w)}\right) - \frac{1}{2\beta}\right)^2\right].$$

As shown, the squared loss function is implemented and β is explicitly positioned outside the logarithm term. Let us examine the behavior of this IPO-like loss for different values of β. In the case of $\beta \to \infty$, the term $\frac{1}{2\beta} \to 0$, simplifying the loss function to:

$$\mathcal{L}_{\text{gen-ipo}}(\phi) = -\mathbb{E}\left[\left(\log\left(\frac{G_\phi(\bar{m}_w)}{G_\phi(\bar{m}_l)} \cdot \frac{G_{\text{ref}}(\bar{m}_l)}{G_{\text{ref}}(\bar{m}_w)}\right)\right)^2\right].$$

To minimize this loss, the following conditions should ideally be met:

$$\frac{G_\phi(\bar{m}_w)}{G_\phi(\bar{m}_l)} \approx \frac{G_{\text{ref}}(\bar{m}_w)}{G_{\text{ref}}(\bar{m}_l)}.$$

Thus, as $\beta \to \infty$, our current model G_ϕ converges to the reference model G_{ref}. In contrast, as $\beta \to 0$, the term $\frac{1}{2\beta} \to \infty$ begins to dominate the loss function, causing the IPO-like loss to converge toward the DPO-like formulation. This suggests that the IPO-like loss exhibits distinct behavior depending on the value of β, even in deterministic preference scenarios. In contrast, the DPO-like loss renders β irrelevant in such scenarios, meaning the loss remains unaffected by changes in β.

E SELF-SUPERVISED TRAINING PROCESS FOR STITCHNET

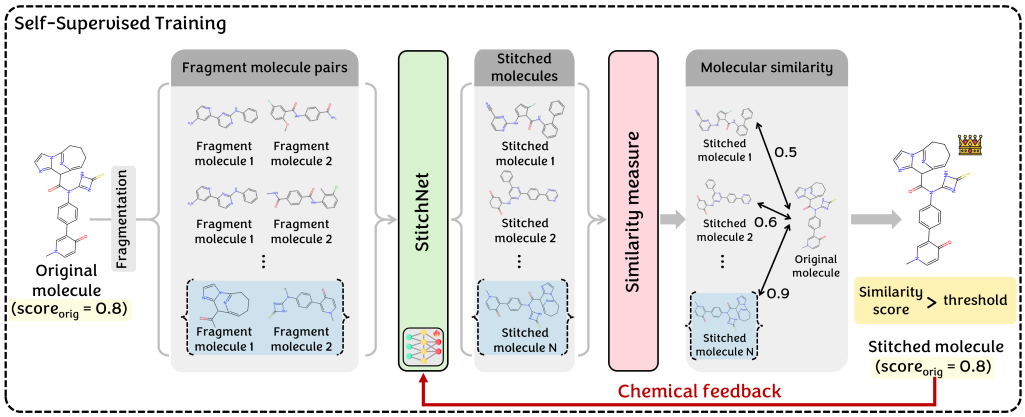


Figure 7: An illustration of the self-supervised training process for StitchNet. An original molecule is sampled from the offline dataset and decomposed into two fragment molecules using a fragmentation function. These pairs of fragment molecules are then fed into StitchNet to generate new stitched molecules. The molecular similarity between the stitched molecule and the original molecule is measured, and if it exceeds a pre-defined threshold, the objective score of the original molecule is leveraged as chemical feedback for StitchNet.

In this section, we provide a detailed explanation of the self-supervised training process for StitchNet, which is a key differentiating factor from traditional rule-based crossover operators. The importance of this process lies in its ability to leverage chemical feedback, allowing StitchNet to better understand how stitched molecules are likely to exhibit objective scores when two molecules are combined. Unlike rule-based crossover operators, StitchNet is built using a neural network architecture that enables it to learn from such chemical feedback.

As shown in Figure 7, we begin by sampling an original molecule from the offline dataset, each with corresponding known objective scores. We then apply a fragmentation function within the crossover operator (Jensen, 2019) to decompose the original molecule into two smaller fragment molecules. There are multiple possible pairings of these fragment molecules, and we consider all viable pairs as inputs to StitchNet. Subsequently, StitchNet takes these pairs of fragment molecules and generates corresponding offspring stitched molecules. We then measure the molecular similarity between each stitched molecule and the original molecule. If the similarity exceeds a certain threshold (e.g., 0.9), we consider the stitched molecule sufficiently similar to the original molecule. This high similarity allows us to leverage the known objective scores of the original molecule as an approximation for the stitched molecule's objective scores, effectively providing chemical feedback to StitchNet. We use this feedback to train StitchNet with the loss function specified in Equation 10. We think that this approach is reasonable based on two key assumptions. First, since the fragment molecules are derived from the original molecule, the stitched molecule is expected to share similar characteristics. Second, because structurally similar molecules often exhibit similar properties (Barbosa & Horvath, 2004; Alvesalo et al., 2006; Maggiora et al., 2014), we assume that the stitched molecule will likely exhibit objective scores comparable to the original molecule. By ensuring that the stitched molecule is sufficiently similar to the original, we can reasonably use the original molecule's objective scores as an approximation for the stitched molecule's scores.

The rationale for this self-supervised training process arises from the inherent nature of the offline MOMO problem. In an online setting, it would be possible to sample two molecules from the offline dataset, input them into StitchNet, generate a stitched molecule, and then query an oracle to obtain its true objective scores for chemical feedback. However, in an offline setting, additional oracle queries are not possible. Therefore, rather than simply using two random molecules from the offline dataset, we decompose a single molecule into two fragment molecules, which are then input into StitchNet. Since the true objective scores of the stitched molecules cannot be obtained due to the unavailability of additional oracle queries, we instead leverage the objective scores of the original molecule as a form of chemical feedback. This allows us to approximate the likely performance of the stitched molecule, ensuring that the training process remains effective even in the offline setting.

F PRIORITY SAMPLING PROCESS FOR STITCHNET

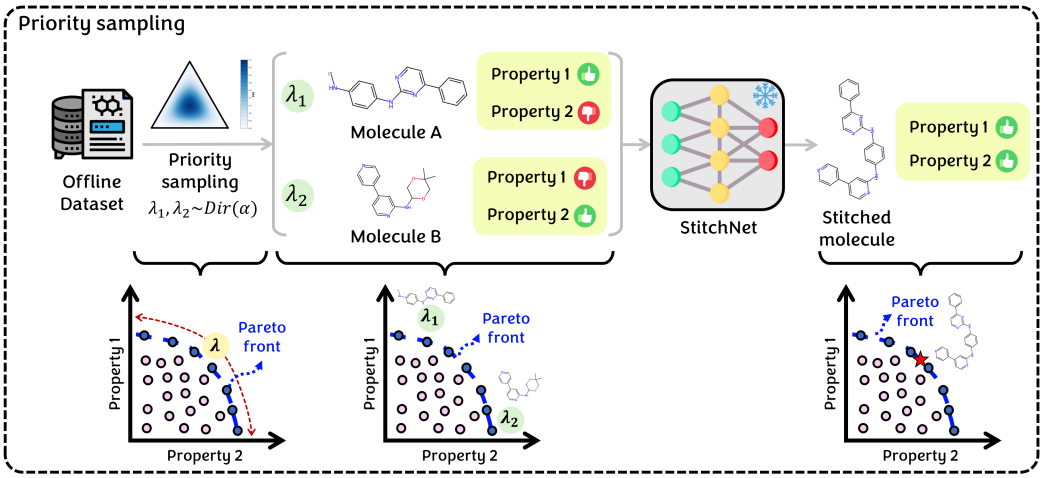


Figure 8: An illustration of the priority sampling process for StitchNet. The figure demonstrates how different weight configurations λ are sampled from the Dirichlet distribution Dir(α), guiding the selection of molecular pairs from the offline dataset. For instance, λ_1 focuses more on property 1, while λ_2 emphasizes property 2, resulting in the selection of molecules A and B, respectively. These molecules are then fed into StitchNet, which generates a novel stitched molecule with the aim of combining the desirable properties of both parent molecules. This priority sampling promotes diversity and balance in the stitched molecules, enhancing the convergence towards the Pareto front.

In this section, we visualize the priority sampling process and explain why it is beneficial for the molecular stitching process in StitchNet. Consider a scenario where we are optimizing two molecular properties: property 1 and property 2, as shown in Figure 8. Our goal is to sample a diverse set of molecular pairs from the offline dataset; for instance, one molecule (Molecule A) exhibits characteristics more aligned with property 1, and the other molecule (Molecule B) emphasizes property 2. This diversity is crucial because StitchNet seeks to combine these molecules to create a novel stitched molecule that inherits the desirable properties of each parent molecule. If we sample pairs of molecules that have similar characteristics or properties, the benefit of the molecular stitching process diminishes due to the lack of diversity. To address this, we propose using priority sampling with a Dirichlet distribution to automatically generate diverse weight configurations, denoted as λ. Different weight configurations indicate varying levels of importance or priority for each property, allowing us to sample molecules from the offline dataset using different perspectives or priorities.

As depicted in Figure 8, λ_1 represents a weight configuration that focuses more on property 1, while λ_2 emphasizes property 2. It is important to note that these weight configurations are sampled from the Dirichlet distribution Dir(α). Based on these configurations, corresponding molecules are sampled from the offline dataset and fed into StitchNet as molecule A and molecule B. StitchNet then generates a novel stitched molecule with the aim of possessing both favorable property 1 and property 2. In terms of the Pareto front, sampling molecules based on λ_1 corresponds to selecting molecules near the y-axis (emphasizing property 1), whereas λ_2 corresponds to molecules near the x-axis (emphasizing property 2). By performing molecular stitching via StitchNet, we aim to generate molecules that balance both properties, thereby improving convergence towards the Pareto front. Please note that weight configurations for focusing on property 1 and property 2 is merely an example for better understanding. In practice, we can sample diverse molecular pairs using priority sampling, as the Dirichlet distribution allows us to automatically generate diverse combinations of weight configurations.

G PSEUDO-CODE

Algorithm 1: StitchNet

Input: StitchNet $\mathcal{S}_\psi$, Unlabelled dataset $\mathcal{D}_u$, Offline dataset $\mathcal{D}$, Crossover operation *Crossover*, Dirichlet distribution *Dir*, Concentration constant α, Fragmentation function *Cut*, Similarity threshold δ, Similarity function `sim`

Output: Generated stitched molecules $\bar{m}$

▷ ***Pre-training for StitchNet***

Sample parent molecules from unlabelled chemical dataset; $m_i \sim \mathcal{D}_u$ and $m_j \sim \mathcal{D}_u$

Generate offspring molecule with crossover operation; $m_o \leftarrow \textit{Crossover}(m_i, m_j)$

Train StitchNet $\mathcal{S}_\psi$ to resemble crossover operation; $\psi \leftarrow \arg\max_\psi \mathbb{P}(m_o \mid S_\psi(m_i, m_j))$

Set pretrained StitchNet as a reference model; $S_{\text{ref}} \leftarrow S_\psi$

▷ ***Self-supervised training for StitchNet***

Sample objective preference; $\lambda \sim \textit{Dir}(\alpha)$

Sample molecule and its score from offline dataset with preference; $(m_s, r_s) \overset{\lambda}{\sim} \mathcal{D}$

Cut m_s into all possible Z fragment molecule sets; $\{(m_{ai}, m_{bi})\}_{i=1}^{Z} \leftarrow \textit{Cut}(m_s)$

Find the most similar offspring and its fragment set with original molecule m_s;

$(m_a, m_b) \leftarrow \underset{(m_{ai}, m_{bi})}{\arg\max}\ \texttt{sim}(m_s, \textit{Crossover}(m_{ai}, m_{bi}))$ subject to $\texttt{sim}(\cdot) \geq \delta$

Provide chemical feedback to StitchNet while maintaining the chemical validity;

$\mathcal{L}_{\text{stitch}}(\psi) \leftarrow (-\log \mathcal{S}_\psi(\bar{m}_{\text{stit}}) + \log \mathcal{S}_{\text{ref}}(\bar{m}_{\text{stit}}) + \mathcal{R}(m_{\text{orig}}))^2$

▷ ***Molecular Stitching***

Sample two objective priorities; $\lambda_1 \sim \textit{Dir}(\alpha)$ and $\lambda_2 \sim \textit{Dir}(\alpha)$

Sample parent molecules of different objective priorities; $m_1 \overset{\lambda_1}{\sim} \mathcal{D}$ and $m_2 \overset{\lambda_2}{\sim} \mathcal{D}$

Generate a novel stitched molecule using fine-tuned StitchNet; $\bar{m} \sim \mathcal{S}_\psi(m_1, m_2)$

Return $\bar{m}$

Algorithm 2: MolStitch

Input: Pretrained Generator G_{ref} Pretrained StitchNet $\mathcal{S}_{\text{ref}}$, Offline dataset $\mathcal{D}$, Proxy model $\hat{f}_\theta$, Dirichlet distribution *Dir*, Concentration constant α,

Output: Final molecules for evaluations m_{final}

Initialize Generative model; $G_\phi \leftarrow G_{\text{ref}}$

Initialize StitchNet; $\mathcal{S}_\psi \leftarrow \mathcal{S}_{\text{ref}}$

Update Generative model G_ϕ with offline dataset $\mathcal{D}$;

Train proxy model $\hat{f}_\theta$ with pairwise ranking loss in eq.9;

Sample objective preference; $\lambda \sim \textit{Dir}(\alpha)$

Finetuning StitchNet $\mathcal{S}_\psi$ with preference λ ;

Sample objective preferences; $\lambda_1, \lambda_2 \sim \textit{Dir}(\alpha)$

Sample stitched molecule $\bar{m}$ by molecular stitching; $\bar{m} \sim \mathcal{S}_\psi(m_1, m_2)$

Determine winning and losing molecules using proxy model $\hat{f}_\theta$ by eq.12; $(m_w, m_l) \leftarrow \hat{f}_\theta(\bar{m})$

Fine-tuning Generative model with IPO-like loss in eq.15;

Sample final molecules for evaluations; $m_{final} \sim G_\phi$

Return m_{final}

H EXPERIMENTAL DETAILS

In this section, we present detailed information on the experimental setups used in our study, including experimental settings, descriptions of the molecular objectives, and implementation details.

H.1 Experimental Settings and Configurations

Oracle calls. In this work, we conducted two main experiments: 1) Molecular Property Optimization (MPO) task (Gao et al., 2022) and 2) docking score optimization task (Lee et al., 2023). Recall that both experiments were designed to simulate real-world constraints by restricting the number of oracle calls, which represent expensive evaluations of molecular properties. For the MPO task, the total number of oracle calls was limited to 10,000 (Gao et al., 2022). Following this guideline, we allocated 5,000 calls to construct the offline dataset and reserved the remaining 5,000 for evaluation. Specifically, we used the initial 5,000 oracle calls to build the offline dataset, which served as the training data for developing and fine-tuning the generative model during the offline optimization process. After completing offline optimization, the performance of the fine-tuned generative model was evaluated using the remaining 5,000 oracle calls on the molecules it newly generated. For the docking score optimization task, the total number of oracle calls was restricted to 3,000 (Lee et al., 2023). This lower allocation might be due to the longer time required for evaluating docking scores. Similar to the MPO task in concept, we allocated 1,500 oracle calls to construct the offline dataset and the remaining 1,500 to evaluate the performance of the fine-tuned generative model.

Offline dataset collection. To construct the offline datasets for both experiments, we utilized the ZINC dataset (Sterling & Irwin, 2015), which is a publicly available chemical database that provides a collection of commercially available compounds. The ZINC dataset offers a wide variety of molecular structures, providing a large chemical space to explore for potential drug candidates. Its compounds are also available in formats suitable for molecular docking, making it a good resource for identifying potential compounds that may bind to biological targets. Therefore, we considered the ZINC dataset to be well-suited for both the MPO task and the docking score optimization task. It is worth noting that we also used the ZINC dataset during the pre-training stage; however, at that stage, we only utilized the molecular structures without any associated objective scores or additional information. When aiming to optimize specific molecular objectives, we needed to query the oracle to obtain the objective scores of molecules within the ZINC dataset. For the MPO task, we randomly sampled 5,000 molecules from the ZINC dataset and executed 5,000 oracle calls to evaluate their corresponding molecular objective scores, such as JNK3, GSK3β, QED, and SA. We collected this data in the form of `(molecule, objective scores)` pairs. Similarly, for the docking score optimization task, we randomly sampled 1,500 molecules from the ZINC dataset and performed 1,500 oracle calls to evaluate their corresponding docking scores for five proteins alongside QED and SA. These constructed offline datasets were subsequently used for offline optimization in our proposed framework as well as across all competing methods to ensure a fair comparison.

H.2 Experimental Workflow for Offline Molecular Optimization

Overall workflow. In this subsection, we aim to conduct an in-depth exploration and comparison of key components in offline MOMO. Specifically, our goal is to outline the critical components that should be considered for solving the offline MOMO problem, discuss the available options for each component, and explain the rationale behind our choices. Figure 9 provides a visual representation of the overall workflow for addressing the offline MOMO problem. The primary objective of offline MOMO is to enhance the generative model's capability to generate molecules that surpass the best-known molecules in the offline dataset. To achieve this, the predominant approach is offline MBO, which involves training a proxy model, performing data augmentation, generating synthetic data, and subsequently training the generative model with this synthetic data under the guidance of the proxy model. Consequently, data augmentation is a pivotal aspect of the offline MOMO problem, and we begin our discussion with this component.

Data augmentation. As highlighted in the main manuscript, we propose StitchNet as a neural network model designed for data augmentation, and demonstrate its effectiveness. However, we acknowledge that StitchNet is not the only viable option. Alternative approaches include stochastic sampling, where new molecules are randomly drawn from the generative model's learned distribution. Additionally, rule-based crossover operators from genetic algorithms can be employed to generate new offspring molecules by combining features from parent molecules.

Proxy training and evaluation. After augmenting the synthetic data, the next step involves training a proxy model to evaluate this augmented dataset. The most straightforward approach is the score-based proxy (vanilla proxy), which directly approximates the scores of the true objective function.

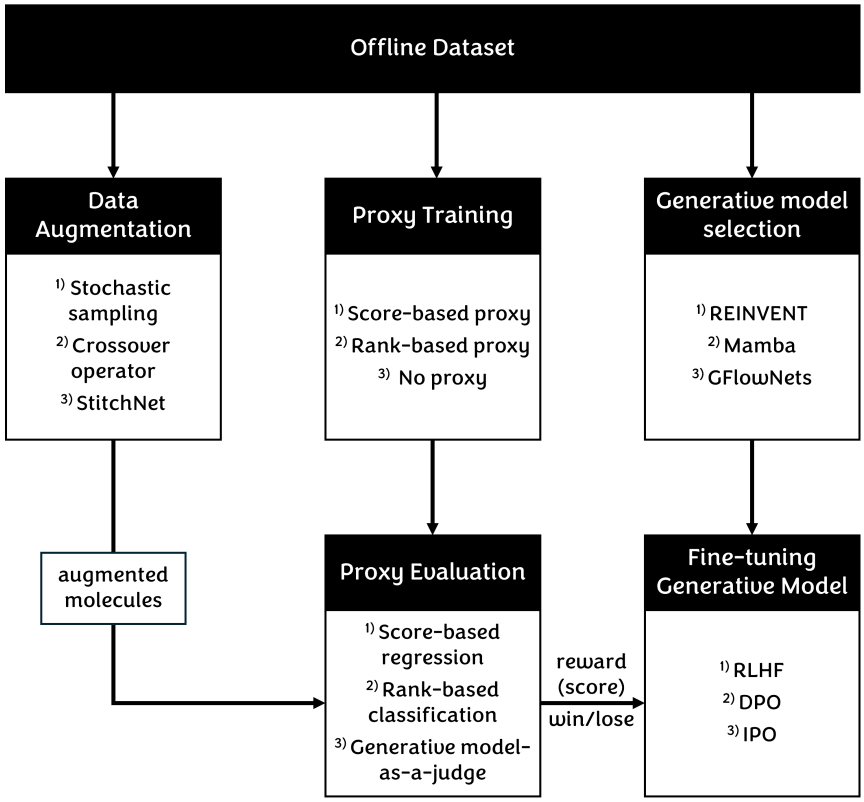


Figure 9: An illustration of the overall workflow for the offline molecular optimization process.

However, we anticipate that as the problem complexity increases, the vanilla proxy may encounter challenges and yield unreliable predictions. To mitigate this, we propose a rank-based proxy that learns the ranking relationships between pairs of molecules based on desired properties, thereby classifying which molecule is more favorable. This transformation from a regression task to a classification task simplifies the proxy's role, enhancing its reliability in providing feedback to the generative model. It is worth noting that a proxy model is not always necessary; in some cases, the generative model itself can evaluate new synthetic data, a mechanism referred to as the "model-as-a-judge".

Generative model selection. Several generative models are available for molecular optimization. In this work, we employ REINVENT as our main generative model due to its widespread use and recognition in various molecular optimization tasks. Nonetheless, recent advancements have introduced new generative models such as Mamba and GFlowNets. To ensure the robustness and versatility of our MolStitch, we also evaluate various backbone generative models within this framework.

Fine-tuning the generative model. With synthetic data, a trained proxy model, and a trained generative model in place, the final step involves fine-tuning the generative model using the synthetic data guided by the proxy model. This fine-tuning process can be considered analogous to the preference optimization process used in large language models. Therefore, we explore various preference optimization techniques within the context of offline MOMO. The first option is RLHF, where the proxy model serves as a reward model to generate rewards that are directly optimized. Another option is DPO, which bypasses reward modeling and focuses on optimizing preferences directly. Lastly, IPO can be applied as an extension of DPO, providing a more theoretically sound and principled approach to preference optimization.

Overview of MolStitch components. Table 6 presents a detailed summary of the components constituting the MolStitch framework, including its variants and the methods examined in our ablation studies. We hope that this table helps to understand the function of each component in our framework and facilitates a clearer understanding of the structure of MolStitch and its variants.

Table 6: Summary of our MolStitch framework components, its variants, and the methods utilized in our ablation studies.

Experiment	Data Augmentation	Proxy Training	Generative Model	Proxy Evaluation	Fine-tuning
MolStitch (Table 1)	StitchNet	Rank-based proxy	REINVENT	Rank-based classification	IPO
Score-based proxy (Table 3)	Stochastic sampling	Score-based proxy	REINVENT	Score-based regression	RLHF
Stochastic sampling (Table 4)	Stochastic sampling	Rank-based proxy	REINVENT	Rank-based classification	IPO
Crossover operator (Table 4)	Crossover operator	Rank-based proxy	REINVENT	Rank-based classification	IPO
StitchNet & RLHF (Table 5)	StitchNet	Rank-based proxy	REINVENT	Score-based regression	RLHF
StitchNet & DPO (Table 5)	StitchNet	No proxy	REINVENT	Generative model-as-a-judge	DPO
StitchNet & IPO (Table 5)	StitchNet	No proxy	REINVENT	Generative model-as-a-judge	IPO
Mamba + MolStitch (Table 13)	StitchNet	Rank-based proxy	Mamba	Rank-based classification	IPO
GFlowNets + MolStitch (Table 13)	StitchNet	Rank-based proxy	GFlowNets	Rank-based classification	IPO

H.3 Descriptions of the Molecular Objectives

In this work, we adopted four commonly used molecular objectives—JNK3, GSK3β, QED, and SA—for the MPO task. For the docking score optimization task, we targeted the docking scores of five proteins—parp1, fa7, jak2, braf, and 5ht1b—alongside QED and SA. The docking scores were calculated following the experimental protocol of prior work (Guo & Schwaller, 2024b), using the normalized QuickVina2 docking score (Alhossary et al., 2015). Specifically, the normalized docking score (DS) is calculated using the given equation:

$$\text{Normalized DS} = -\frac{\text{DS}}{20}.$$

where DS represents the original docking score. Detailed descriptions of each molecular objective and protein are provided below.

JNK3. JNK3 is a member of the c-Jun N-terminal kinases (JNKs) family, which belongs to the mitogen-activated protein kinase (MAPK) pathway and is primarily expressed in the central nervous system (Bogoyevitch & Kobe, 2006). It plays a crucial role in mediating cellular responses to stress, including apoptosis, inflammation, and neuronal damage (Bogoyevitch & Kobe, 2006). Targeting JNK3 inhibition is one of the key molecular objectives in drug discovery because it may prevent or reduce neuronal cell death and inflammation, making it a promising therapeutic target for neurodegenerative diseases such as Alzheimer's disease (Resnick & Fennell, 2004).

GSK3β. Glycogen synthase kinase 3 beta (GSK3β) is a serine/threonine protein kinase involved in various cellular processes, including glycogen metabolism, cell proliferation, differentiation, and apoptosis (Beurel et al., 2015). It has gained significant attention in neurodegenerative disease research due to its role in regulating tau protein phosphorylation, amyloid precursor protein processing, and neuronal survival (Jope et al., 2007). Inhibiting GSK3β is considered a vital molecular

objective in drug discovery, as it could modulate these pathological processes and potentially slow or prevent the progression of neurodegenerative diseases (Jope et al., 2007).

QED. Quantitative Estimate of Drug-likeness (QED) is a metric widely used in molecular optimization to evaluate the drug-likeness of a molecule (Bickerton et al., 2012). It consists of several physicochemical properties, including molecular weight, lipophilicity (logP), topological polar surface area (TPSA), the number of hydrogen bond donors and acceptors, and the count of aromatic rings and rotatable bonds (Guan et al., 2019). It provides a score ranging from 0 to 1, with higher scores indicating molecules that are more likely to have favorable drug-like properties.

SA. Synthetic Accessibility (SA) is a metric used in molecular optimization to assess the ease with which a molecule can be synthesized in a laboratory setting (Ertl & Schuffenhauer, 2009). It considers various structural features that influence synthesis complexity, such as the presence of complex ring systems, functional groups, stereocenters, and the overall size and branching of the molecule (Ertl & Schuffenhauer, 2009). The SA score ranges from 1 to 10, with lower scores indicating higher synthetic feasibility. In this work, we transform the SA score into the normalized SA score, following prior studies (Lee et al., 2023; Guo & Schwaller, 2024b), to formulate it as a maximization objective. Specifically, the normalized SA score is given by the following equation:

$$\text{Normalized SA} = \frac{10 - \text{SA}}{9}.$$

This adjustment ensures that higher normalized SA scores correspond to molecules that are easier to synthesize, within the score range of 0 to 1.

parp1. Poly (ADP-ribose) polymerase 1 (parp1) is a protein enzyme that plays a crucial role in DNA damage detection and repair (Rouleau et al., 2010). It is involved in various cellular processes, including chromatin remodeling, transcriptional regulation, and cell death signaling (Ray Chaudhuri & Nussenzweig, 2017). In recent years, dysregulation of parp1 activity has been linked to several neurodegenerative diseases, such as Parkinson's disease, where excessive activation of parp1 can lead to neuronal death through a process known as parthanatos (Liu et al., 2022). Consequently, targeting parp1 has become a key molecular objective in drug discovery, not only for cancer treatment but also for developing neurotherapeutics aimed at preventing neuronal loss (Zhang et al., 2023).

fa7. Coagulation factor VII (fa7), also known as proconvertin, is a vital protein in the blood coagulation pathway (Hall et al., 1964). It plays a crucial role in initiating the clotting process by activating factor X in the presence of tissue factor (TF), leading to the conversion of prothrombin to thrombin and ultimately forming a blood clot (Eigenbrot, 2002). Targeting fa7 represents another key molecular objective in drug discovery, particularly for managing thrombotic and cardiovascular diseases. Specifically, inhibitors of fa7 are being explored as potential anticoagulants to prevent and treat conditions such as deep vein thrombosis, embolism, and stroke (Robinson et al., 2010).

jak2. Janus kinase 2 (jak2) is a non-receptor tyrosine kinase that plays a critical role in the signaling pathways of various cytokines (Yamaoka et al., 2004). It is involved in various cellular processes, including cell growth, differentiation, and immune function (Seavey & Dobrzanski, 2012). In drug discovery, jak2 has gained attention due to its association with myeloproliferative neoplasms and other hematological malignancies (Senkevitch & Durum, 2017). Inhibiting jak2 is considered a key molecular objective, as it can potentially provide therapeutic benefits in inflammatory and autoimmune disorders (Seavey & Dobrzanski, 2012).

braf. B-Raf proto-oncogene (braf) encodes a serine/threonine kinase that is part of the MAPK/ERK signaling pathway, which plays a crucial role in regulating cell growth and migration during various cellular processes (González-González et al., 2020). Mutations in the braf gene are commonly found in various cancers, including melanoma, colorectal cancer, and thyroid cancer (Atiqur Rahman et al., 2014). Therefore, targeting the braf can be a critical therapeutic objective in oncology to target these cancer-specific mutations and halt the progression of the disease (Sanz-Garcia et al., 2017).

5ht1b. 5-Hydroxytryptamine receptor 1B (5ht1b) is a G protein-coupled receptor that binds serotonin (Launay et al., 2002). It is widely expressed in the central nervous system and plays important roles in regulating neurotransmitter release, neuronal firing, mood, and appetite (Fink & Göthert, 2007). 5ht1b has emerged as an important molecular target in drug discovery for neurological and psychiatric disorders, particularly in the treatment of migraine and depression (Giniatullin, 2022).

H.4 Hyperparameters and Implementation Details

Implementation of the generative model. We closely followed the architecture settings for REINVENT as described in the PMO benchmark (Gao et al., 2022), while the settings for GFlowNet were based on GeneticGFN (Kim et al., 2024a), and those for Mamba were taken from Saturn (Guo & Schwaller, 2024b). Since all of these generative models were originally designed for an online setting, we made necessary adjustments to the number of molecule updates and the experience replay to adapt them for our offline settings. The final hyperparameters for the generative models were primarily determined based on the performance of REINVENT, which served as our backbone generative model, and are detailed in Table 7.

Stabilizing GFlowNets. During the training of GFlowNets, we encountered instability with the original setting of the `logZ` parameter, which plays a crucial role in trajectory balancing and needs to be adjusted according to specific settings (Malkin et al., 2022). To be more specific, it was initially set to a high value (`logZ` = 5.0) with a learning rate of 0.1, as specified in GeneticGFN. To stabilize the training process, we reduced the `logZ` value to 0.001 and aligned the learning rate with that of the generative model (from 0.1 to 0.0005). This adjustment resulted in more stable training and significantly improved performance. Additionally, during preference optimization, while both REINVENT and Mamba require only the generative model's likelihood as input, we recommend using the sum of likelihood and `logZ` for GFlowNets in order to further improve performance.

Hyperparameters for StitchNet. Recall that StitchNet combines two parent molecules as input and generates stitched molecules in an auto-regressive manner. Therefore, it operates by computing the hidden dimensions h_1 and h_2 of two parent molecules m_1 and m_2, respectively, and then averaging these hidden dimensions as $\frac{h_1+h_2}{2}$. StitchNet is built upon the REINVENT architecture. During the self-supervised training process for StitchNet, we applied a similarity threshold $\delta = 0.8$ between the original molecules and the stitched molecules. During the molecular stitching process, StitchNet combines two parent molecules, each sampled with different weight configurations through priority sampling. The resulting stitched molecules are stored in a buffer. Once the buffer is full, two molecules are randomly sampled to create non-overlapping pairs. These pairs are then evaluated by the proxy model to identify the winning and losing molecules. Subsequently, the IPO-like loss is applied to increase the likelihood of generating winning molecules while reducing the likelihood of generating losing molecules. The hyperparameter settings for Stitchnet are summarized in Table 8.

Table 7: The hyperparameter settings for generative models in MolStitch framework.

REINVENT		GFlowNets		Mamba	
Batch size	200	Batch size	200	Batch size	200
Embedding dimension	128	Embedding dimension	128	Embedding dimension	256
Hidden dimension	512	Hidden dimension	512	Hidden dimension	256
Number of layers	3	Number of layers	3	Number of layers	12
Sigma	500	Sigma	500	Sigma	500
Experience replay size	300	Experience replay size	300	Experience replay size	300
Augmentation round	8	Augmentation round	8	Augmentation round	8
Batch update	2	Batch update	2	Batch update	2
Learning rate	5e-04	Learning rate	5e-04	Learning rate	5e-04
		`logZ`	0.001		

Table 8: The hyperparameter settings for StitchNet.

Molecular stitching	
α for priority sampling	1.0
Number of stitch rounds	16
Stitched molecules per stitch round	250
Population pool	1000
Temperature β for IPO	0.2

I COMPETING METHODS DETAILS

In this section, we present a comprehensive review of the competing methods, highlighting their core principles, methodologies, and their comparative position relative to our proposed framework. Before delving into the details, we first aim to explain how molecular optimization methods, such as REINVENT (Olivecrona et al., 2017), are adapted to offline settings. While we use REINVENT as an example, this approach applies to all competing molecular optimization methods. In online settings, REINVENT actively generates molecules, queries the oracle to obtain objective scores as rewards, and updates the log-likelihood of generating those molecules based on the feedback. In contrast, in offline settings, it relies on a pre-existing offline dataset containing pairs of molecules and their corresponding objective scores, rather than actively generating and evaluating new molecules through oracle queries. This offline dataset becomes the sole source of information for training and optimizing the generative model. In this context, REINVENT computes the log-likelihood of a molecule and utilizes the corresponding objective scores from the offline dataset as rewards, updating itself in a supervised manner. This adaptation enables REINVENT to operate in offline settings, leveraging the available offline data to refine its generative capabilities.

- **REINVENT** (Olivecrona et al., 2017) is a reinforcement learning (RL) approach designed for molecular generation, where an agent interacts with its environment to create molecules. This approach autoregressively generates molecules as SMILES strings, with each new element (token) in the sequence building upon the previously generated elements. Note that this generation process is guided by a pre-trained model that enforces chemical grammar rules, ensuring the validity of the generated molecules. REINVENT has demonstrated superior performance in molecular optimization tasks, as highlighted by the PMO benchmark. This remarkable performance has led numerous follow-up studies to adopt REINVENT as their backbone generative model. Following this established trend, we have also integrated REINVENT as our backbone model to take advantage of its proven effectiveness in various molecular optimization tasks. As a competing method in our study, REINVENT serves as a baseline, as it is trained exclusively on the offline dataset without applying further offline MBO techniques. This straightforward approach positions it as a reference point for evaluating the effectiveness of various MBO techniques, which leverage proxy models to fine-tune the generative model beyond the constraints of the offline dataset.
- **REINVENT-BO** (Tripp et al., 2021) integrates an RL-based method with the Bayesian optimization (BO) framework. To construct REINVENT-BO, we adopt the same mechanism and framework as GPBO (Tripp et al., 2021) but use REINVENT as the backbone model instead of GraphGA (Jensen, 2019). This adaptation leverages the BO process to enhance the molecular generation capabilities of REINVENT, enabling efficient exploration of the optimization landscape. In our study, REINVENT-BO is designed to demonstrate the potential performance enhancements that the BO framework can achieve in the offline MOMO problem, positioning it as a reference point for assessing the impact of BO.
- **Augmented Memory (AugMem)** (Guo & Schwaller, 2024a) builds upon the REINVENT method by incorporating molecular data augmentation techniques and experience replay to enhance performance. The authors report that AugMem has achieved state-of-the-art results on the PMO benchmark, showcasing its effectiveness in molecular optimization tasks. In the context of offline optimization, offline MBO techniques typically use proxy models to guide the generation of synthetic data. This process involves the generative model producing new data points, which are then evaluated by the proxy model. The resulting augmented dataset allows the generative model to explore beyond the initial offline dataset. AugMem, in contrast, introduces a different approach to data augmentation specifically designed for molecular generation. By implementing AugMem in our study, we establish a valuable reference point for comparing specialized molecular data augmentation techniques against the proxy model-guided approaches used in conventional offline MBO.

- **DST** (Fu et al., 2022) is a gradient-based optimization method that utilizes a graph neural network (GNN) to edit molecular structures represented as chemical graphs. Specifically, DST backpropagates derivatives from the target molecular properties to optimize and refine the graph representation. In our study, DST serves as a valuable reference point for employing a GNN proxy model to guide the molecular optimization process.

- **GraphGA** (Jensen, 2019) is a method based on genetic algorithms that generates molecules by evolving a population through repeated cycles of selection, crossover, and mutation, all driven by a fitness function. GraphGA utilizes domain knowledge from the chemical experts to develop effective mutation and crossover strategies that facilitate an efficient exploration of molecular space. In our study, GraphGA serves as a key reference for implementing rule-based crossover operations, which we have also incorporated into our framework.
- **GeneticGFN** (Kim et al., 2024a) integrates genetic algorithms into the GFlowNets model for molecular generation. Specifically, this method leverages domain-specific genetic operators to efficiently explore the chemical space, enabling the generative model to implicitly acquire relevant domain knowledge. Consequently, the generative model's performance is enhanced through the strategic guidance provided by the genetic algorithm. The authors also highlight a complementary relationship between the two components: the genetic algorithm enhances GFlowNets' capacity for effective exploitation, while GFlowNets, in turn, increases the population diversity for the genetic algorithm. In our study, GeneticGFN serves as a crucial reference point for evaluating the effectiveness of genetic algorithms in the offline MOMO problem. Specifically, it allows us to assess the advantages gained from incorporating domain-specific knowledge through genetic operators in this context.
- **Saturn** (Guo & Schwaller, 2024b) builds upon the core mechanism of REINVENT while introducing significant architectural improvements. While REINVENT employs a GRU architecture, Saturn replaces it with the more powerful Mamba architecture. This substitution is motivated by Mamba's potentially greater capacity for modeling complex molecular structures more effectively. Furthermore, Saturn incorporates genetic algorithms into its Mamba-based model, drawing parallels to GeneticGFN's approach. This integration allows Saturn to leverage domain-specific genetic operators, potentially enhancing its ability to navigate the chemical space effectively. In our study, Saturn serves as a valuable reference point for two key aspects: first, it demonstrates the application of the Mamba architecture in molecular optimization tasks, and second, it provides insights into the benefits of incorporating domain-specific genetic operators in the context of offline MOMO.
- **Grad** (Zinkevich, 2003) represents the most straightforward offline MBO approach for tackling the offline MOMO problem. In particular, it employs a vanilla proxy model that directly approximates the true objective scores, training this proxy on the offline dataset. To address the generative aspect of the offline MOMO problem, Grad utilizes REINVENT as its backbone generative model, the same approach used in our proposed framework. This choice is consistently applied across all offline MBO-based competing methods to ensure a fair comparison. After training the vanilla proxy model, Grad fine-tunes the generative model using gradient ascent with respect to the trained vanilla proxy model's predictions. In our study, Grad serves as a crucial reference point as it demonstrates the basic application of offline MBO in the context of offline MOMO. Specifically, Grad enables us to investigate whether a vanilla proxy model is sufficient for this task, or if more sophisticated approaches are necessary for meaningful improvements in the offline MOMO problem.
- **COMs** (Trabucco et al., 2021) represents a more sophisticated offline MBO approach. Unlike Grad's vanilla proxy model, COMs employs adversarial learning to encourage the proxy model to provide conservative estimates of the true objective functions. This method establishes lower bounds on the objective estimates, which are then used during the offline optimization process. By doing so, COMs aims to prevent erroneous overestimation caused by distributional shift, a common challenge in various offline optimization scenarios. In our study, COMs enables us to investigate whether these sophisticated methods offer significant improvements in the context of offline MOMO.
- **IOM** (Qi et al., 2022) considers offline MBO from a domain adaptation perspective. This method aims to train a proxy model that can accurately predict true objective scores ('target domain') when trained solely on the given offline dataset ('source domain'). To achieve this, IOM introduces invariant representation learning, which enforces alignment between the learned distribution of the offline dataset and the distribution of optimized decisions. In our study, IOM serves as a reference point similar to COMs, enabling us to evaluate the effectiveness of invariant representation learning in addressing distributional shifts and enhancing performance in the offline MOMO problem.

- **RoMA** (Yu et al., 2021) also addresses the challenge of overestimation issues when approximating true objective scores. To mitigate this issue, RoMA proposes robust model adaptation by incorporating a local smoothness prior as a regularizer. This regularizer aims to enforce a flat loss landscape, thereby enhancing the proxy model's generalization capabilities and ensuring stable training. In our study, RoMA serves as a reference point, similar to COMs and IOM, allowing us to assess the effectiveness of using regularization techniques to improve robustness and performance in the offline MOMO problem.
- **Ensemble proxy** (Trabucco et al., 2022) takes a different offline MBO approach by leveraging multiple proxy models through ensemble learning. This approach addresses the limitations of a single proxy model, which can be prone to overfitting issues. Ensemble proxy uses multiple proxies with different initializations and averages their predictions to approximate true objective scores. In our study, Ensemble proxy serves as a reference point, enabling us to evaluate the effectiveness of ensemble learning in the offline MOMO problem and assess whether the potential performance gains justify the increased computational cost associated with using multiple proxy models.
- **ICT** (Yuan et al., 2023) utilizes multiple proxies, similar to Ensemble proxy, but enhances the approach through a co-teaching process. This process facilitates information exchange between proxies and encourages knowledge transfer. Additionally, ICT incorporates a meta-learning-based sample reweighting mechanism that iteratively updates the importance weights of samples to mitigate potential inaccuracies in pseudo-labels. In our study, ICT serves as a reference point, enabling us to evaluate the effectiveness of advanced ensemble techniques, such as co-teaching and meta-learning, in the offline MOMO problem.
- **Tri-Mentoring** (Chen et al., 2023a) is closely related to ICT, utilizing multiple proxies and facilitating learning between them through a mentoring process. However, Tri-Mentoring shifts its focus to generating pairwise comparison labels rather than directly approximating objective scores. Instead of averaging predictions, it employs majority voting to combine decisions from each proxy model. In our study, Tri-Mentoring serves as a crucial reference point, enabling us to evaluate the effectiveness of using the rank-based proxy over the score-based proxy, aligning closely with the approach of our proxy model.
- **BIB** (Chen et al., 2023b) employs a bidirectional learning approach that utilizes both forward and backward mappings to generate input configurations likely to produce optimal outputs, while conforming to the data distribution of the offline dataset. BIB constructs its proxy model using a pre-trained language model and applies a deep linearization scheme to derive a closed-form loss function. It is recognized as one of the best models for tackling the offline biological sequence design problem. In our study, BIB serves as a reference point to evaluate how well a high-performing method designed for offline biological sequence design performs in the offline MOMO problem.
- **BootGen** (Kim et al., 2023) employs a bootstrapping technique to enhance the optimization process by iteratively augmenting the offline dataset with self-generated data, using the proxy model as a pseudo-labeler. The goal is to align and refine the generative model through iterative training, where high-quality samples are added to the augmented dataset based on the proxy model's guidance. BootGen is also recognized as one of the best models for offline biological sequence design. In our study, BootGen serves as a reference point to evaluate the effectiveness of the bootstrapping technique in offline optimization, and, similar to BIB, to assess how well a high-performing method designed for offline biological sequence design can be adapted to tackle the offline MOMO problem.

J Details on Evaluation Metrics

This section provides an overview of the evaluation metrics used in this study: the hypervolume (HV) indicator (Zitzler et al., 2003) and the R2 indicator (Brockhoff et al., 2012). Both metrics are widely employed in multi-objective optimization due to their effectiveness in evaluating solution quality across conflicting objectives. The HV indicator quantifies the volume of the objective space dominated by the Pareto front relative to a reference point, reflecting convergence and diversity. In contrast, the R2 indicator measures how well the Pareto front aligns with a set of reference directions, assessing solution distribution. Using both metrics together provides complementary insights into the performance of optimization algorithms and the exploration of trade-offs among objectives.

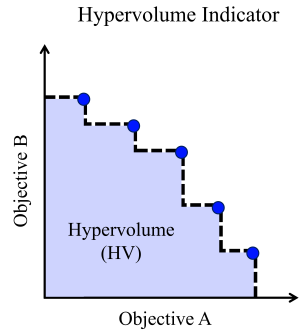


Figure 10: Visualization of the hypervolume (HV) indicator in a two-dimensional space, where the HV corresponds to the volume of the shaded region.

J.1 Hypervolume Indicator

The HV indicator denoted as I_H, measures the volume in the objective space that is dominated by the Pareto front derived from the optimization algorithm. To be more specific, the HV indicator is defined as the volume in the objective space that is dominated by a set of solutions $\mathcal{X}$ relative to a reference point z^r. Of note, the reference point z^r is chosen such that it is dominated by all solutions in $\mathcal{X}$, representing the worst acceptable value for each objective. Mathematically, the HV can be expressed using the Lebesgue integral as follows:

$$I_H(\mathcal{X}, z^r) = \int_{\mathbb{R}^n} \mathbb{I}_{\{z^r | z^r \leq x \text{ for some } x \in \mathcal{X}\}}(z^r)\, dz^r,$$

where $\mathbb{I}$ is the indicator function that equals to 1 if the reference point $z^r \in \mathbb{R}^n$ is dominated by at least one solution $x \in \mathcal{X}$, i.e., $z^r \leq x$ for some $x \in \mathcal{X}$, and 0 otherwise. This formulation essentially measures the volume of the region in the objective space that is dominated by the solutions in $\mathcal{X}$ and bounded above by the reference point z^r. Alternatively, the HV can be calculated more practically as follows:

$$I_H(\mathcal{X}, z^r) = \text{Vol}\left(\bigcup_{x \in \mathcal{X}} [x, z^r]\right),$$

where $[x, z^r]$ denotes the hyperrectangle with lower corner x and upper corner z^r. This representation provides a more intuitive understanding of the HV indicator as it directly corresponds to the union of hyperrectangles formed by each solution in $\mathcal{X}$ with respect to z^r. In a nutshell, the HV indicator quantifies the size of the objective space that is simultaneously dominated by all solutions in $\mathcal{X}$ and is within the bounds defined by z^r. A larger HV value indicates a more preferable set of solutions, as it implies that a greater portion of the objective space is covered by the set $\mathcal{X}$.

To provide a clear understanding, we visualized HV as shown in Figure 10, where the blue points represent a Pareto front composed of non-dominated solutions. Then the HV is defined as a measure of the region in the objective space that is dominated by the Pareto front and bounded by a reference point. In this study, as we have normalized all objective values between 0 and 1, 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.

J.2 R2 Indicator

The R2 indicator (Brockhoff et al., 2012) is a set-based performance metric used in multi-objective optimization to evaluate the quality of a set of solutions $\mathcal{X}$ in approximating the true Pareto front. Unlike the HV indicator, which measures the volume of the dominated region, the R2 indicator uses a set of predefined weight vectors to assess how well the solutions in $\mathcal{X}$ represent various trade-offs among objectives. It is defined as the maximum of the worst-case weighted distances between the solutions in $\mathcal{X}$ and an ideal or utopian point. A lower R2 value indicates better performance, as it signifies that the solutions in $\mathcal{X}$ are closer to the ideal point for all considered weight vectors.

Mathematically, let $\mathcal{W}$ be a set of weight vectors $\mathbf{w} = (w_1, w_2, \ldots, w_m)$, where $w_i \geq 0$ and $\sum_{i=1}^{m} w_i = 1$, representing different priorities for the objectives. The R2 indicator, denoted as

$R2(\mathcal{X}, \mathcal{W})$, can be defined as:

$$R2(\mathcal{X}, \mathcal{W}) = \max_{\mathbf{w} \in \mathcal{W}} \min_{\mathbf{x} \in \mathcal{X}} \left\{ \sum_{i=1}^{m} w_i \cdot [f_i^*(\mathbf{x}) - f_i(\mathbf{x})] \right\},$$

where $f_i(\mathbf{x})$ is the value of the i-th objective for the solution $\mathbf{x}$, and $f_i^*(\mathbf{x})$ is the value of the i-th objective for the ideal or utopian point (typically the maximum achievable value for maximization problems). This formulation calculates the deviation of the solution set $\mathcal{X}$ from the ideal point for each weight vector $\mathbf{w}$ and then takes the maximum of these deviations across all weight vectors in $\mathcal{W}$. The use of the maximum operator ensures that the R2 indicator focuses on the worst-case scenario for any given weight vector, reflecting the least favorable trade-off among objectives that the solution set $\mathcal{X}$ can achieve. A lower R2 value means that $\mathcal{X}$ is closer to the ideal point across all weight vectors, indicating a better approximation of the Pareto front.

In summary, the R2 indicator quantifies the worst-case performance of a set of solutions $\mathcal{X}$ in terms of their proximity to an ideal point for a given set of weight vectors $\mathcal{W}$. A lower R2 value is better as it indicates a closer approximation to the ideal performance across all weight vectors.

K Additional Results

K.1 Evaluating Molecular Optimization Methods Using Average Property Score (APS) of Top 10 and Top 100 Molecules

In our main results, we presented performance using the Hypervolume (HV) and R2 indicator metrics, which are widely regarded as the most appropriate evaluation metrics for multi-objective optimization tasks. However, within the molecular optimization community, the average property score (APS) is another commonly used metric, specifically tailored for assessing molecular optimization methods. To provide a more comprehensive assessment, we conducted additional experiments to report APS for various molecular optimization methods. The methods we evaluated include GraphGA (Jensen, 2019), which generates molecules by using rule-based crossover operations to combine features from parent molecules; LigGPT (Bagal et al., 2021), which is suitable for offline settings as it does not require oracle calls during molecule generation; DST (Fu et al., 2022), which leverages a proxy model to facilitate precise functional group editing; and REINVENT (Olivecrona et al., 2017), our backbone generative model, known for its robust performance in molecular optimization tasks. Additionally, we also considered AugMem (Guo & Schwaller, 2024a), a leading model in the PMO benchmark, Saturn (Guo & Schwaller, 2024b), which enhances sample efficiency in molecular design, and GeneticGFN (Kim et al., 2024a), which integrates GFlowNets with genetic algorithms to achieve state-of-the-art performance across various molecular optimization tasks. In this experiment, we calculated the APS of the top 10 and top 100 molecules generated by each method and reported the mean APS. As shown in Table 9, our MolStitch framework consistently outperformed all competing methods, even when evaluated with the molecule-specific metric. This result demonstrates the robustness and superiority of MolStitch across diverse evaluation criteria.

Table 9: Experimental results on molecular property optimization tasks under the full-offline setting. The evaluation metric is the average property score (APS) of the top 10 and 100 molecules.

Molecular objectives	GSK3β+JNK3		GSK3β+JNK3+QED		GSK3β+JNK3+QED+SA	
Method	top10 (↑)	top100 (↑)	top10 (↑)	top100 (↑)	top10 (↑)	top100 (↑)
REINVENT	0.515±0.076	0.312±0.036	0.464±0.018	0.383±0.005	0.564±0.018	0.491±0.003
AugMem	0.558±0.066	0.374±0.036	0.515±0.041	0.407±0.010	0.579±0.015	0.505±0.005
LigGPT	0.335±0.027	0.199±0.005	0.461±0.027	0.380±0.005	0.548±0.014	0.485±0.002
GraphGA	0.466±0.079	0.313±0.058	0.512±0.048	0.415±0.012	0.593±0.038	0.507±0.010
DST	0.456±0.058	0.315±0.037	0.531±0.059	0.451±0.039	0.601±0.027	0.539±0.029
Saturn	0.559±0.074	0.358±0.037	0.546±0.032	0.443±0.041	0.608±0.043	0.513±0.041
GeneticGFN	0.540±0.077	0.379±0.078	0.548±0.058	0.451±0.051	0.599±0.027	0.524±0.029
REINVENT-BO	0.539±0.055	0.346±0.025	0.485±0.021	0.392±0.007	0.572±0.024	0.498±0.005
MolStitch (Ours)	**0.627±0.056**	**0.432±0.039**	**0.591±0.040**	**0.468±0.016**	**0.671±0.041**	**0.564±0.024**

K.2 R2 PERFORMANCE FOR THE DOCKING SCORE OPTIMIZATION TASK

Results for R2 performance. We present additional R2 performance results for the docking score optimization task in Table 10. Consistent with the findings in Table 2 of the main manuscript, our MolStitch framework demonstrated superior performance by achieving the lowest R2 indicator score compared to all competing methods.

Table 10: Experimental results on docking score optimization tasks under the full-offline setting. The evaluation metric is the R2 indicator, with the best values highlighted in bold.

Target protein	parp1	jak2	braf	fa7	5ht1b
Method	R2(↓)	R2(↓)	R2(↓)	R2(↓)	R2(↓)
REINVENT	1.426±0.090	1.589±0.042	1.497±0.044	1.791±0.033	1.454±0.054
AugMem	1.374±0.163	1.523±0.159	1.471±0.044	1.729±0.220	1.421±0.064
Saturn	1.376±0.053	1.501±0.155	1.420±0.176	1.726±0.201	1.350±0.139
GeneticGFN	1.326±0.148	1.589±0.039	1.484±0.025	1.701±0.228	1.410±0.057
REINVENT-BO	1.421±0.061	1.569±0.036	1.483±0.049	1.800±0.062	1.410±0.063
Grad	1.422±0.032	1.555±0.079	1.461±0.036	1.750±0.184	1.401±0.134
COMs	1.448±0.041	1.568±0.089	1.467±0.109	1.816±0.031	1.459±0.045
IOM	1.402±0.041	1.597±0.045	1.488±0.070	1.806±0.034	1.421±0.160
RoMA	1.431±0.053	1.604±0.044	1.434±0.153	1.738±0.241	1.449±0.058
Ensemble Proxy	1.415±0.035	1.568±0.062	1.491±0.036	1.800±0.028	1.470±0.038
BIB	1.425±0.045	1.573±0.029	1.500±0.034	1.801±0.028	1.478±0.043
BootGen	1.320±0.136	1.521±0.037	1.420±0.030	1.712±0.142	1.336±0.184
ICT	1.428±0.024	1.591±0.029	1.473±0.100	1.810±0.028	1.472±0.045
Tri-Mentoring	1.373±0.155	1.553±0.083	1.428±0.098	1.793±0.033	1.443±0.045
MolStitch (Ours)	**1.276±0.153**	**1.445±0.177**	**1.312±0.174**	**1.674±0.261**	**1.231±0.165**

K.3 SEMI-OFFLINE OPTIMIZATION

Definition of semi-offline optimization. Semi-offline optimization, also referred to as batch hybrid learning (Xiong et al., 2024), is an optimization approach that bridges the gap between offline and online optimization. In this semi-offline setting, models are trained on a combination of pre-existing offline datasets and periodically collected new data, enabling periodic updates without the need for continuous or real-time oracle queries. Unlike the full-offline setting, where the model is trained exclusively on a static offline dataset, the semi-offline setting allows for the periodic incorporation of new data in large batches, facilitating a more dynamic learning process. This semi-offline optimization is particularly useful in scenarios where obtaining new data in real-time is either too costly or logistically challenging, yet some level of interaction or adaptation to new data is beneficial.

Semi-offline optimization in LLMs. Semi-offline optimization has gained considerable attention in the field of large language models (LLMs). Several studies (Bai et al., 2022; Touvron et al., 2023) have implemented a strategy of iteratively applying the RLHF process on a weekly cadence. This involves periodically deploying updated RLHF models to interact with users or crowdworkers to collect new preference data. The models are then fine-tuned with this feedback on a regular schedule. Recently, Xiong et al. (2024) further extended this approach by formulating it as a batch hybrid framework, establishing a more general setting for the hybrid learning process.

Experimetal setup for semi-offline optimization. Motivated by these practical applications, we conducted additional experiments on MPO tasks under the semi-offline setting. We began by constructing an initial offline dataset using 5,000 oracle calls. In contrast to the full-offline setting, where all remaining 5,000 oracle calls were used for evaluation, the semi-offline setting employed a different allocation strategy. Specifically, we allocated 2,500 oracle calls for the periodic integration of new molecular data in large batches. This allocation enabled the generative model to iteratively update and adapt based on the newly acquired data. The remaining 2,500 oracle calls were reserved for the final evaluation, allowing us to assess the model's performance under the semi-offline setting.

Results for semi-offline optimization. As illustrated in Table 11, our MolStitch framework consistently outperformed all competing methods under the semi-offline setting. Notably, we observed a general improvement in performance compared to the full-offline setting, as shown in Table 1 of

Table 11: Experimental results on molecular property optimization tasks for the **semi-offline** setting. The evaluation metrics are the hypervolume (HV) and R2 indicators, with the best values in bold.

Molecular objectives	GSK3β+JNK3		GSK3β+JNK3+QED		GSK3β+JNK3+QED+SA	
Method	HV(↑)	R2(↓)	HV(↑)	R2(↓)	HV(↑)	R2(↓)
REINVENT	0.581±0.057	0.694±0.109	0.208±0.065	2.372±0.300	0.175±0.064	4.053±0.747
AugMem	0.636±0.063	0.602±0.113	0.348±0.075	1.888±0.237	0.292±0.087	3.225±0.650
GraphGA	0.521±0.084	0.819±0.136	0.392±0.102	1.623±0.277	0.265±0.080	3.493±0.537
DST	0.493±0.049	0.867±0.090	0.313±0.051	2.065±0.194	0.297±0.064	3.274±0.396
Saturn	0.623±0.049	0.621±0.086	0.428±0.040	1.581±0.160	0.382±0.088	2.686±0.510
GeneticGFN	0.642±0.065	0.592±0.107	0.414±0.123	1.660±0.425	0.361±0.086	2.879±0.569
REINVENT-BO	0.662±0.109	0.556±0.203	0.350±0.083	2.885±0.470	0.268±0.115	2.131±0.401
Grad	0.584±0.075	0.708±0.136	0.216±0.086	2.458±0.371	0.180±0.037	4.109±0.455
COMs	0.571±0.058	0.717±0.105	0.219±0.073	2.505±0.351	0.186±0.046	3.956±0.505
IOM	0.603±0.061	0.647±0.081	0.221±0.077	2.349±0.395	0.205±0.065	3.899±0.621
RoMA	0.588±0.067	0.680±0.109	0.215±0.070	2.414±0.258	0.180±0.036	4.105±0.414
Ensemble Proxy	0.602±0.084	0.648±0.146	0.227±0.071	2.435±0.332	0.216±0.069	3.730±0.573
BIB	0.563±0.066	0.713±0.122	0.215±0.078	2.440±0.388	0.189±0.070	4.062±0.735
BootGen	0.608±0.057	0.646±0.098	0.233±0.093	2.399±0.462	0.219±0.090	3.924±0.651
ICT	0.601±0.078	0.662±0.143	0.216±0.089	2.455±0.389	0.185±0.048	4.094±0.454
Tri-Mentoring	0.592±0.078	0.678±0.144	0.219±0.054	2.467±0.241	0.206±0.073	3.966±0.603
MolStitch (Ours)	**0.689±0.041**	**0.514±0.073**	**0.539±0.045**	**1.238±0.157**	**0.493±0.050**	**2.014±0.202**

Table 12: Performance of various preference optimization techniques for the **semi-offline** setting.

Molecular objectives	GSK3β+JNK3		GSK3β+JNK3+QED		GSK3β+JNK3+QED+SA	
Method	HV(↑)	R2(↓)	HV(↑)	R2(↓)	HV(↑)	R2(↓)
Baseline (REINVENT)	0.462±0.133	0.921±0.259	0.196±0.083	2.646±0.327	0.168±0.046	3.969±0.664
+ StitchNet & RLHF	0.675±0.059	0.526±0.091	0.448±0.066	1.540±0.221	0.383±0.082	2.647±0.463
+ StitchNet & DPO	0.685±0.047	0.520±0.083	0.507±0.078	1.342±0.221	0.447±0.060	2.320±0.331
+ StitchNet & IPO	0.681±0.042	0.521±0.069	0.527±0.055	1.256±0.133	0.462±0.055	2.187±0.299
+ MolStitch (Ours)	**0.689±0.041**	**0.514±0.073**	**0.539±0.045**	**1.238±0.157**	**0.493±0.050**	**2.014±0.202**

the main manuscript. This finding highlights the benefits of incorporating periodic new data, as it enables the generative model to be fine-tuned and trained on newly acquired samples, thereby further enhancing its optimization capabilities. Consistent with the trends observed in the full-offline setting, Saturn and GeneticGFN maintained strong performance among competing methods, highlighting the effectiveness of genetic algorithms in offline MOMO. Their success could be attributed to the inherent strengths of genetic algorithms in maintaining population diversity and effectively exploring the Pareto front through crossover operations. This finding aligns with our framework, which employs a mechanism analogous to crossover, but with the added advantage of incorporating chemical feedback. Additionally, we conducted experiments for preference optimization techniques under the semi-offline setting, as depicted in Table 12. The trends observed were similar to those in the full-offline setting, our MolStitch consistently achieved the highest performance among all competing methods. While RLHF performed well on the two-objective scenario, its performance declined significantly as the number of objectives increased. Both DPO and IPO demonstrated strong performance, with IPO showing a slight edge over DPO.

K.4 Evaluating Mamba and GFlowNets as Additional Backbone Models

Various backbone models. In this work, we chose REINVENT as our backbone generative model due to its widespread use and reputation as one of the top-performing models for various molecular optimization tasks. However, as previously mentioned, Saturn and GeneticGFN demonstrated strong performance in numerous offline MOMO experiments. Since these methods utilized Mamba and GFlowNets as their respective backbone models, we conducted additional experiments using Mamba and GFlowNets as the backbone generative model for our MolStitch framework.

Results for backbone models. As shown in Table 13, we report the performance of each backbone generative model—REINVENT, Mamba, and GFlowNets—on MPO tasks under the full-offline setting, alongside the performance of integrating either our rank-baesd proxy or MolStitch framework

with each backbone model (e.g., REINVENT + MolStitch). Similarly, Table 14 presents the performance of the backbone generative models and their respective integrations with MolStitch under the semi-offline setting. As shown in Table 13, both the rank-based proxy and the MolStitch framework provide performance improvements across various generative models. However, the integration with the rank-based proxy still falls short compared to the full MolStitch framework, emphasizing the additional benefits brought by StitchNet and priority sampling. Notably, Mamba + MolStitch and GFlowNets + MolStitch outperformed REINVENT + MolStitch in both three-objective and four-objective scenarios. This superior performance could be attributed to the greater capacity of Mamba and GFlowNets to manage the increased complexity associated with optimizing multiple objectives beyond two. Overall, the consistent performance improvements across different backbone generative models under both full-offline and semi-offline settings demonstrate the robustness and versatility of our MolStitch. Moreover, these additional results highlight the MolStitch's ability to seamlessly integrate with a range of backbone models, demonstrating its adaptability beyond a single model architecture.

Table 13: Performance comparison of different generative models on molecular property optimization tasks under the **full-offline** setting.

Molecular objectives	GSK3β+JNK3		GSK3β+JNK3+QED		GSK3β+JNK3+QED+SA	
Method	HV(↑)	R2(↓)	HV(↑)	R2(↓)	HV(↑)	R2(↓)
REINVENT	0.462±0.133	0.921±0.259	0.196±0.083	2.646±0.327	0.168±0.046	3.969±0.664
+ Rank-based Proxy	0.545±0.063	0.773±0.120	0.319±0.059	1.928±0.314	0.251±0.084	3.504±0.634
+ MolStitch (Ours)	0.579±0.070	0.698±0.128	0.403±0.065	1.649±0.259	0.352±0.080	2.953±0.571
Mamba	0.531±0.087	0.785±0.159	0.293±0.058	1.977±0.280	0.281±0.058	3.339±0.280
+ Rank-based Proxy	0.538±0.068	0.758±0.105	0.327±0.100	1.946±0.404	0.281±0.072	3.317±0.486
+ MolStitch (Ours)	0.544±0.071	0.761±0.128	0.407±0.077	1.617±0.199	0.361±0.063	2.893±0.424
GFlowNets	0.482±0.073	0.869±0.117	0.309±0.087	1.990±0.365	0.237±0.066	3.630±0.453
+ Rank-based Proxy	0.522±0.040	0.805±0.085	0.364±0.070	1.809±0.305	0.323±0.054	2.953±0.304
+ MolStitch (Ours)	0.525±0.063	0.770±0.111	0.415±0.087	1.685±0.343	0.366±0.088	2.708±0.652

Table 14: Performance comparison of different generative models on molecular property optimization tasks under the **semi-offline** setting.

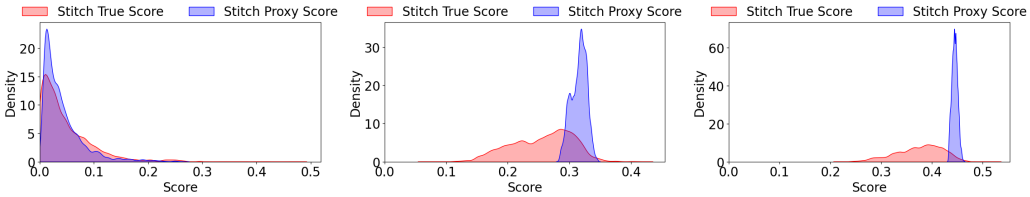
Molecular objectives	GSK3β+JNK3		GSK3β+JNK3+QED		GSK3β+JNK3+QED+SA	
Method	HV(↑)	R2(↓)	HV(↑)	R2(↓)	HV(↑)	R2(↓)
REINVENT	0.581±0.057	0.694±0.109	0.208±0.065	2.372±0.300	0.175±0.064	4.053±0.747
+ MolStitch (Ours)	0.689±0.041	0.514±0.073	0.539±0.045	1.238±0.157	0.493±0.050	2.014±0.202
Mamba	0.623±0.049	0.621±0.086	0.428±0.040	1.581±0.160	0.382±0.088	2.686±0.510
+ MolStitch (Ours)	0.653±0.046	0.580±0.090	0.485±0.054	1.430±0.196	0.434±0.044	2.385±0.176
GFlowNets	0.642±0.065	0.592±0.107	0.414±0.123	1.660±0.425	0.361±0.086	2.879±0.569
+ MolStitch (Ours)	0.658±0.068	0.563±0.108	0.579±0.041	1.137±0.130	0.482±0.076	2.181±0.438

L Detailed Analysis of Rank-based Proxy

In this section, we provide an in-depth analysis of both rank-based and score-based proxies. Our study suggests that the formulation of rank-based proxy simplifies the proxy's task, thereby enabling it to deliver more reliable feedback to the generative model. To further explore this, we delve deeper into the performance of each proxy type, examining whether the rank-based proxy truly surpasses the score-based proxy in handling complex multi-objective molecular optimization tasks.

Proxy models. In the context of utilizing proxy models, they offer distinct advantages, but they also present notable challenges. Specifically, Grad is built upon REINVENT and incorporates a vanilla score-based proxy that directly approximates objective scores. As shown in Table 1 of our main manuscript, while Grad outperforms the baseline REINVENT, its performance gains gradually diminish as the number of objectives increases from two to four. This suggests that with the rise in

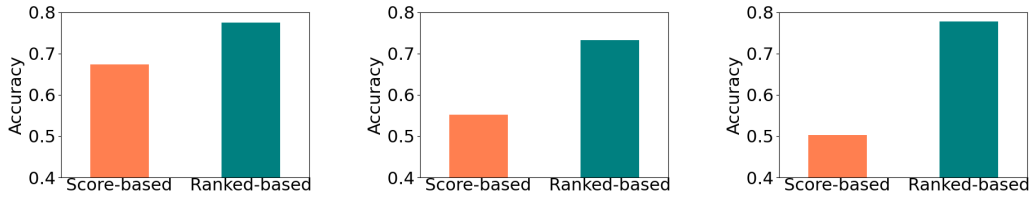
the number of objectives, the problem complexity increases, causing the vanilla proxy to struggle to accurately approximate the objective scores. In contrast, our framework demonstrates particularly strong performance in the three and four objective scenarios, which highlights the effectiveness of reformulating the proxy model's task from direct property score regression to pairwise classification.



(a) GSK3β+JNK3 (2 objectives) (b) GSK3β+JNK3+QED (3 objectives) (c) GSK3β+JNK3+QED+SA (4 objectives)

Figure 11: Distribution comparison of true objective scores (red) and score-based proxy model predictions (blue) for stitched molecules across varying numbers of objectives: (a) 2 objectives, (b) 3 objectives, and (c) 4 objectives. As the number of objectives increases, the score-based proxy model's predictions show less variability and exhibit a sharper central peak, failing to accurately represent the true score distribution.

Score-based proxy. As shown in Figure 11, we visualize the distribution of the true scores for the stitched molecules alongside the predicted scores from the score-based proxy model. Compared to the distribution of true objective scores, the predictions made by the score-based proxy model are significantly more confined to a narrow range. This issue becomes more pronounced as the number of objectives increases, with the score-based proxy model's predictions showing even less variability and a stronger central peak, failing to represent the true score distribution accurately. Therefore, this result indicates that the score-based proxy model fails to provide meaningful feedback to the generative model, potentially leading to suboptimal optimization. To address these limitations, we propose a rank-based proxy model that learns the relative ranking between pairs of molecules based on desired properties, determining which molecule is more favorable. This approach bypasses the direct approximation of true objective scores and instead focuses on ranking relationships, providing more reliable feedback signals for the generative model.



(a) GSK3β+JNK3 (2 objectives) (b) GSK3β+JNK3+QED (3 objectives) (c) GSK3β+JNK3+QED+SA (4 objectives)

Figure 12: Accuracy comparison of score-based and rank-based proxy models in predicting the ranking of randomly selected molecule pairs across varying numbers of objectives: (a) 2 objectives, (b) 3 objectives, and (c) 4 objectives.

Result for proxy models. To demonstrate the effectiveness of the rank-based proxy, we compare the performance of score-based and rank-based proxy models in predicting the rank of randomly selected pairs of molecules. As shown in Figure 12, the rank-based model consistently outperforms the score-based model across all scenarios with varying objectives. This performance gap widens as the number of objectives increases, with the rank-based model maintaining relatively high accuracy even with four objectives, while the score-based model's accuracy drops significantly. These findings validate the superiority of the rank-based proxy over the score-based proxy in effectively addressing the complexities of offline MOMO tasks.

M ADDITIONAL EXPERIMENTS ON MULTIPLE PROXIES

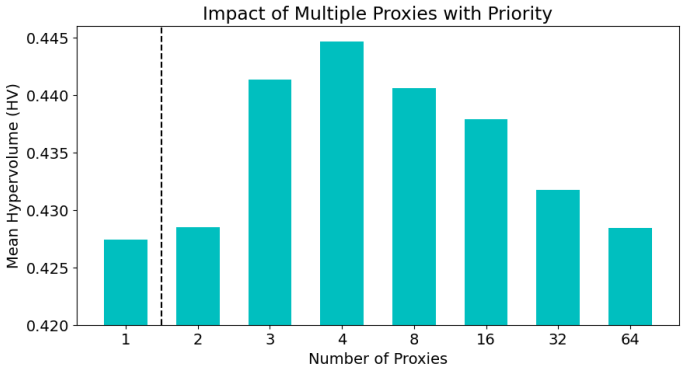


Figure 13: An illustration of the impact of employing multiple proxies with priority sampling in our framework. The evaluation metric is the mean hypervolume across all numbers of objectives for the MPO task under the full-offline setting. The results demonstrate that the optimal configuration for our framework is four proxies, achieving the best performance before a decline due to redundancy.

Motivation for multiple proxies. In this section, we provide a detailed process and analysis of employing multiple proxy models within our framework. The motivation for experimenting with multiple proxies arises from observations in both offline MBO and LLM research. In offline MBO, methods employing multiple proxies—such as Ensemble Proxy, ICT, and Tri-Mentoring—generally outperform single proxy methods like Grad. This finding aligns with a recent study in large language models (LLMs) (Chakraborty et al., 2024), which highlights the drawbacks of using a single reward model to represent human preferences. Researchers note that human preferences are inherently diverse, and a single model often fails to reflect this variability, leading to biased or suboptimal outcomes. To address this, they propose using multiple reward models to capture a broader spectrum of preference distributions, thereby enhancing alignment with diverse human judgments.

Setup for multiple proxies. Inspired by these insights, we enhance our proxy model by incorporating ensemble learning through the use of multiple proxies. In the context of LLMs, preferences reflect human sentiments, opinions, or judgments about desirable outputs. In molecular optimization, however, preference represents the relative importance or priority of each objective within the optimization process. To effectively capture this diversity of priorities, we employ priority sampling for each proxy model, allowing them to prioritize objectives differently according to their assigned importance. Specifically, each proxy receives weight configurations sampled from a Dirichlet distribution, enabling it to focus more on certain objectives than others. As a result, each proxy can determine which molecule in a given pair is superior from its unique perspective. These individual assessments are then combined using a majority voting strategy, providing a comprehensive evaluation of molecules from multiple viewpoints to determine the overall superior molecule.

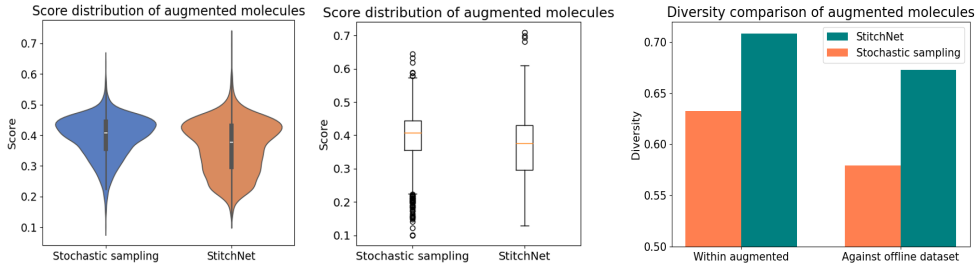
Results for multiple proxies. As demonstrated in Figure 13, the performance of our framework increases with the number of proxies, peaking at four before gradually declining thereafter. The observed decline in performance beyond four proxies can be attributed to the balance between ensemble diversity and redundancy. For an ensemble to be effective, the individual proxy models should be diverse, each providing unique insights into molecule evaluation. While adding proxy models up to a certain point enhances performance by capturing a wider range of priorities, adding too many proxies can introduce redundancy. Beyond the optimal number, additional proxies may become similar to existing ones, offering little new information and potentially amplifying common errors. In addition, with a large number of proxies, majority voting can overlook minority opinions, reducing ensemble diversity and neglecting smaller yet significant priorities. Lastly, note that all configuration settings—whether employing a single proxy or multiple proxies—outperform all competing methods, underscoring the effectiveness of our framework.

Analysis for multiple proxies. One might question how majority voting works with an even number of proxies, as it could lead to a tie. In such cases where the proxies are evenly split in their assessments (e.g., two proxies favor a molecule while two do not), we interpret this as an indication

of uncertainty or difficulty in evaluating the molecule. Rather than making a hasty decision that could misguide the optimization process, we choose to pass and skip these uncertain molecules. This approach ensures that only molecules with a higher degree of consensus among the proxies influence the optimization, enhancing the reliability of the feedback signals. The results also validate that employing four proxies surpasses the performance of using three proxies. In the four-proxy setup, a molecule must receive at least three favorable votes to be considered superior, raising the confidence threshold compared to the two-out-of-three votes required in the three-proxy setup. The stricter criterion in the four-proxy setup leads to more reliable and accurate feedback, contributing to improved optimization performance.

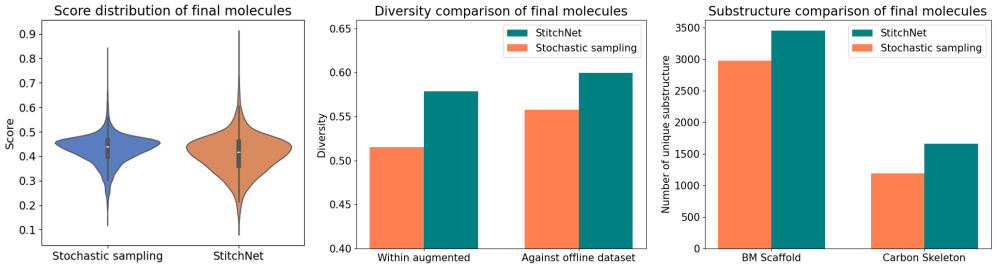
N Additional Analysis on Molecular Diversity of StitchNet

Additional diversity metrics. In the manuscript, we compared the diversity achieved by StitchNet with that of its data augmentation counterpart, stochastic sampling. We found that StitchNet exhibits greater diversity, which we attribute to its crossover-like mechanism that enables the generation of considerably more diverse molecules than stochastic sampling. To further investigate the diversity achieved by StitchNet, we propose the use of additional diversity metrics to provide a more comprehensive analysis from multiple perspectives. To quantify the diversity of the augmented molecules, we employed the inverse of the Tanimoto similarity (Bender & Glen, 2004). Specifically, we calculated the maximum Tanimoto similarity for each augmented molecule with respect to all other augmented molecules, then averaged these values and subtracted the result from 1, which we term the '`Within augmented`' diversity metric. In addition, we computed the maximum Tanimoto similarity between each augmented molecule and the molecules in the offline dataset, similarly subtracting this value from 1 to derive the '`Against offline dataset`' diversity metric.



(a) Score distribution (b) Score distribution (c) Diversity comparison

Figure 14: Diversity analysis of augmented molecules generated by StitchNet and its data augmentation counterpart, stochastic sampling. The results demonstrate the superior capability of StitchNet in generating a diverse and novel set of augmented molecules.



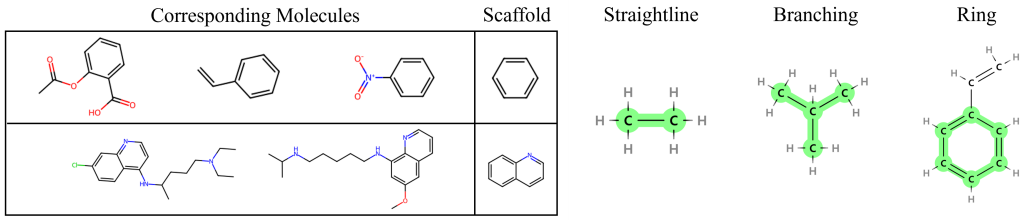
(a) Score distribution (b) Diversity comparison (c) Substructure comparison

Figure 15: Diversity analysis of final molecules produced by the generative model fine-tuned with StitchNet and with stochastic sampling. The results demonstrate that the generative model fine-tuned with StitchNet consistently achieves higher diversity and performance across all diversity metrics.

Additional diversity results. The results in Figure 14 (a-b) demonstrate that StitchNet produces a much broader and more varied score distribution compared to stochastic sampling. This broader

distribution highlights the StitchNet's capability to generate augmented molecules with higher diversity, thereby enriching the fine-tuning process for the generative model. Moreover, the additional diversity metrics further emphasize the advantages of StitchNet over stochastic sampling. As shown in Figure 14 (c), StitchNet consistently achieves higher values in both the `Within augmented` and `Against offline dataset` diversity metrics. This indicates that augmented molecules generated by StitchNet not only show greater diversity among themselves but also display more novelty in comparison to the molecules present in the offline dataset. Additionally, we evaluated the final molecules produced by the generative model fine-tuned with StitchNet against those fine-tuned with stochastic sampling, as shown in Figure 15. The generative model fine-tuned with StitchNet outperforms its counterpart in every aspect: (a) score distribution of final molecules, (b) diversity metrics for both `Within augmented` and `Against offline dataset`, and (c) diversity based on Bemis-Murcko (BM) scaffolds and Carbon Skeletons (CS) (Bemis & Murcko, 1996).

BM scaffolds & Carbon Skeletons. BM scaffolds are an essential tool for breaking down organic molecules to identify their core chemical substructures. As shown in Figure 16 (a), BM scaffolds simplify molecules by removing side chains while preserving the core substructures—such as ring systems and connecting linkers—representing the molecular backbone. This approach allows for a more effective quantitative assessment of structural diversity by comparing the backbones of different molecules. Another method for assessing structural diversity is through CS, which describe various configurations of carbon atoms, including straightline, branching, and ring, as depicted in 16 (b). In particular, straightline skeletons consist of carbon atoms connected in a linear arrangement, while branching skeletons contain side chains that extend from the main carbon chain that potentially affects the molecule's reactivity and interactions with biological targets. Ring skeletons are closed loops of carbon atoms, commonly found in biologically active compounds. Both BM scaffolds and CS serve as complementary methods for simplifying and categorizing molecular structures to better understand their properties and interactions. While BM scaffolds focus on the core substructures by removing side chains and functional groups, CS emphasizes the basic carbon framework of a given molecule. By incorporating both approaches in our analysis, we believe we can conduct a more comprehensive evaluation of structural diversity across the generated molecules.



(a) Bemis-Murcko (BM) scaffolds

(b) Carbon Skeletons (CS)

Figure 16: Visual representations of (a) the Bemis-Murcko scaffolds and (b) the Carbon Skeletons.

O FUTURE WORK AND LIMITATIONS

In this study, we focused on optimizing the properties of small molecules and docking scores for five specific proteins. A natural extension of this work would be to apply our framework to material discovery, particularly for optimizing inorganic molecules, thereby broadening its applicability beyond small molecules. Additionally, while we investigated both full-offline and semi-offline optimization settings, there remains considerable potential to enhance the semi-offline optimization. One promising direction is the use of a behavior policy to improve exploration of chemical space when periodically incorporating new molecule data. This strategy would enable the inclusion of molecules that were not present in the initial offline dataset, leading to more effective integration of newly obtained data. Even in cases where the initial offline dataset contains lower-quality molecules, a behavior policy could progressively improve the quality of the data over time. Moreover, our results suggest that employing multiple proxies yields valuable insights and substantial performance gains in specific cases. As such, future work will focus on further developing and optimizing multiple proxy methods to fully realize their potential in molecular discovery. In addition, during the molecular stitching process, our StitchNet learns from rule-based crossover operator, which is predefined by domain experts using chemical knowledge. Another promising avenue for future work is to incorporate additional domain-specific knowledge into the molecular stitching process, particularly focusing on fundamental chemical relationships between molecular structure and functionality, such as stereoisomerism, reactivity patterns, and steric effects.

P QUANTITATIVE ASSESSMENT OF STITCHNET'S ABILITY TO LEARN CROSSOVER OPERATIONS

	High Scoring	Middle Scoring	Low Scoring	Similarity
Assigned Score	43%	31%	26%	0.644

Table 15: Assigned Scores and overall similarity between StitchNet and Crossover operator

In this section, we present the quantitative results evaluating how effectively StitchNet learns the crossover operation. To assess this, we generated 300 offspring molecules using rule-based crossover operations, and 100 molecules using StitchNet with the same parent molecule pairs. Then, the 300 molecules from rule-based crossover were categorized into three groups based on their mean target objective scores (GSK3β+JNK3+QED+SA): high-scoring, middle-scoring, and low-scoring. For each group, we calculated the mean Tanimoto similarity score with the 100 molecules generated by StitchNet. Each StitchNet-generated molecule was then assigned to the group with which it exhibited the highest similarity score. The results, presented in Table 15, demonstrate that the overall similarity scores are reasonable, suggesting that StitchNet effectively learns crossover operations through its unsupervised pretraining process. Importantly, StitchNet-generated molecules were most frequently assigned to the top-scoring group, with the lowest assignment to the low-scoring group. This outcome highlights the advantages of StitchNet's self-supervised training process, which effectively integrates chemical feedback. As a result, StitchNet can perform crossover operations in a way that preferentially generates offspring molecules with higher objective scores.

Q EFFECTIVENESS AND CONTRIBUTION OF STITCHNET WITHIN OUR FRAMEWORK

	QED	SA	JNK3	GSK3β
Improvement	-5.79%	-3.15%	+16.10%	+42.18%

Table 16: Overall improvement in objective scores when comparing stitched molecules against existing molecules in the offline dataset.

To assess the quality of the newly generated molecules from StitchNet, we measured the improvement and non-improvement in objective scores (GSK3β, JNK3, QED, SA) between the stitched molecules and the existing molecules in the offline dataset. Table 16 presents the results, showing the percentage of improvement and non-improvement. Compared to the existing molecules in the offline dataset, the newly generated molecules from StitchNet exhibited significant increases in challenging objectives such as GSK3β and JNK3, while showing slight decreases in easier-to-optimize objectives like QED and SA. This suggests that StitchNet effectively provides diversity beyond the offline dataset and enhances performance in challenging objectives with only a minor reduction in easier objectives. Consequently, the generative model can learn from this enriched set of high-quality molecules generated by StitchNet, leading to an overall improvement in performance.

R Reward hacking problem in Multi-objective Optimization

In our study, we address the multi-objective molecular optimization problem, which involves simultaneously optimizing multiple objectives. However, during this process, we observed that certain molecular objectives conflicted with each other. To investigate further, we conducted an in-depth analysis of each property score within a four-objective scenario (GSK3β, JNK3, QED, and SA).

We found that models often prioritized easier objectives, such as QED and SA, over more challenging ones like GSK3β and JNK3. As noted by Gao et al. (2022), QED is often considered too trivial, allowing most models to achieve high scores on this objective with minimal effort. This suggests that increasing and optimizing the QED score is much simpler compared to tackling more challenging objectives. For instance, models like REINVENT, which receive rewards based on the average property score, may focus on easily attainable objectives to maximize the overall reward. Consequently, this creates the reward hacking problem, where the model overfits to easier objectives while neglecting the more challenging ones. This behavior highlights the inherent difficulty in multi-objective optimization, particularly when some objectives are easier to optimize than others.

One possible approach to address this issue could be adjusting the weights assigned to each objective to balance their influence—placing more emphasis on the challenging objectives and less on the easier ones. However, this approach relies on having prior domain knowledge about the difficulty of each objective, which is not always available. Moreover, in offline settings, immediate feedback to refine weights is limited, making this approach impractical.

To overcome these challenges, we introduced priority sampling using a Dirichlet distribution within our MolStitch framework for Pareto optimization. This approach efficiently generates diverse weight configurations, ensuring a balanced exploration of all objectives. By using priority sampling within our framework, we promote the generation of a diverse set of stitched molecules that do not disproportionately favor easier objectives, thereby mitigating the risk of reward hacking.

	QED	SA	JNK3	GSK3β
w/o MolStitch	**0.843**	**0.889**	0.128	0.397
MolStitch (Ours)	0.709	0.802	**0.485**	**0.688**

Table 17: Property scores for each objective in a four-objective scenario (GSK3β, JNK3, QED, SA).

To validate the effectiveness of our MolStitch framework, we compared the property scores for each objective in a four-objective scenario (GSK3β, JNK3, QED, and SA) before and after applying our MolStitch framework that incorporates priority sampling. The results, presented in Table 17, clearly indicate that without MolStitch, the models suffer from the reward hacking problem, achieving disproportionately high scores on easier objectives like QED and SA while exhibiting extremely low scores on more challenging objectives such as JNK3 and GSK3β. In contrast, applying our MolStitch framework results in a more balanced optimization, with relatively improved and well-distributed scores across all objectives.

S MOLECULE EXAMPLES

In this section, we first present visual examples of molecules generated by StitchNet, which combines parent molecules to produce stitched molecules. These stitched molecules serve as valuable training samples for fine-tuning the generative model. We then provide visual examples of molecules generated by the fine-tuned generative model, which aims to produce novel molecules that surpass the best-known molecules in the offline dataset. Specifically, we present representative molecules sampled from the Pareto front in the four-objective optimization scenario (QED+SA+JNK3+GSK3β). Each molecule illustrates a distinct trade-off among these objectives, demonstrating the diverse range of solutions on the Pareto front. These examples emphasize the ability of our framework to explore diverse molecules that effectively balance multiple objectives.

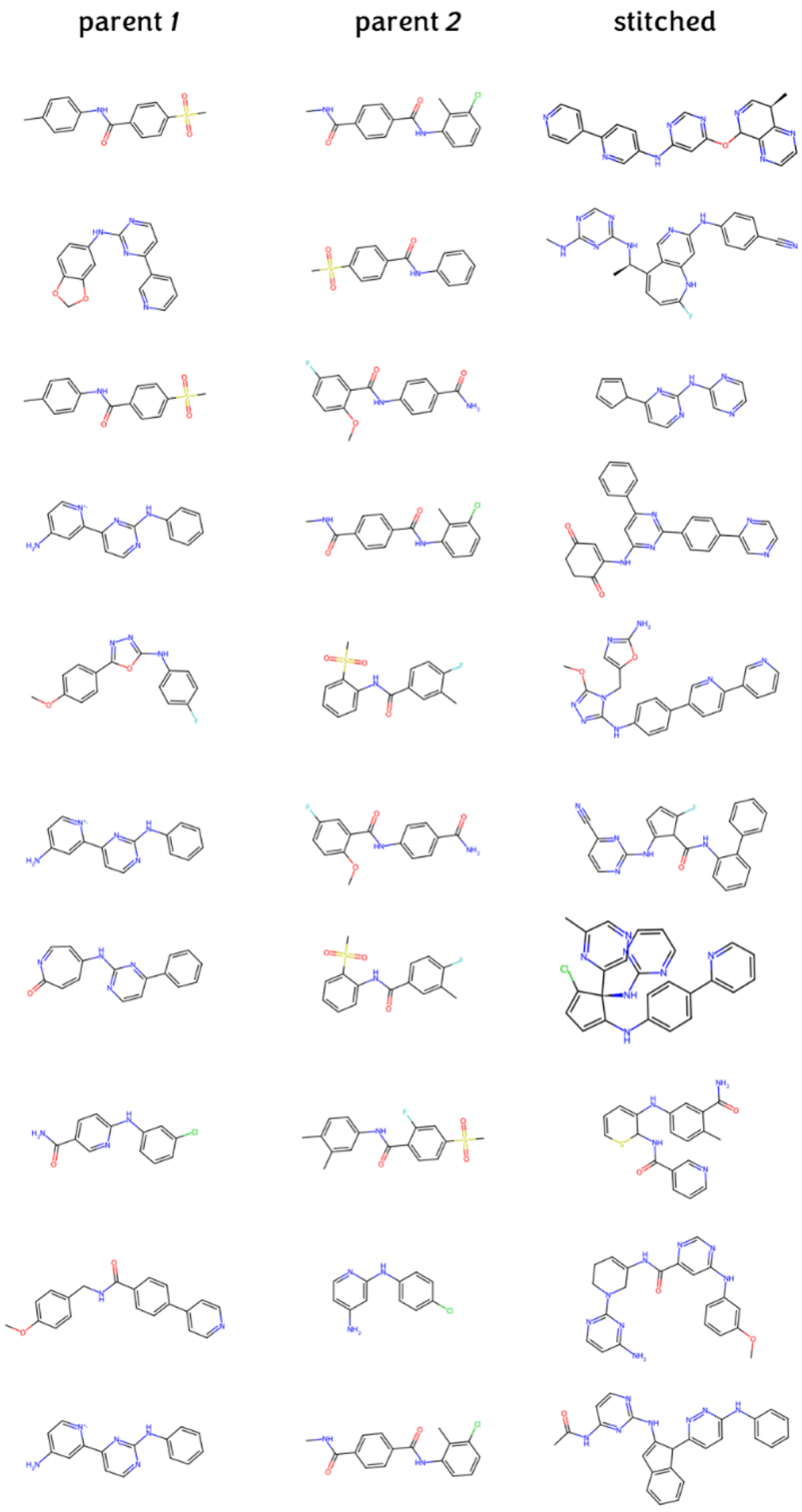


Figure 17: Examples of parent molecules and their corresponding stitched molecules generated by StitchNet. The parent molecules are shown on the left, while the stitched molecules—produced by combining structural fragments from the parent molecules—are displayed on the right.

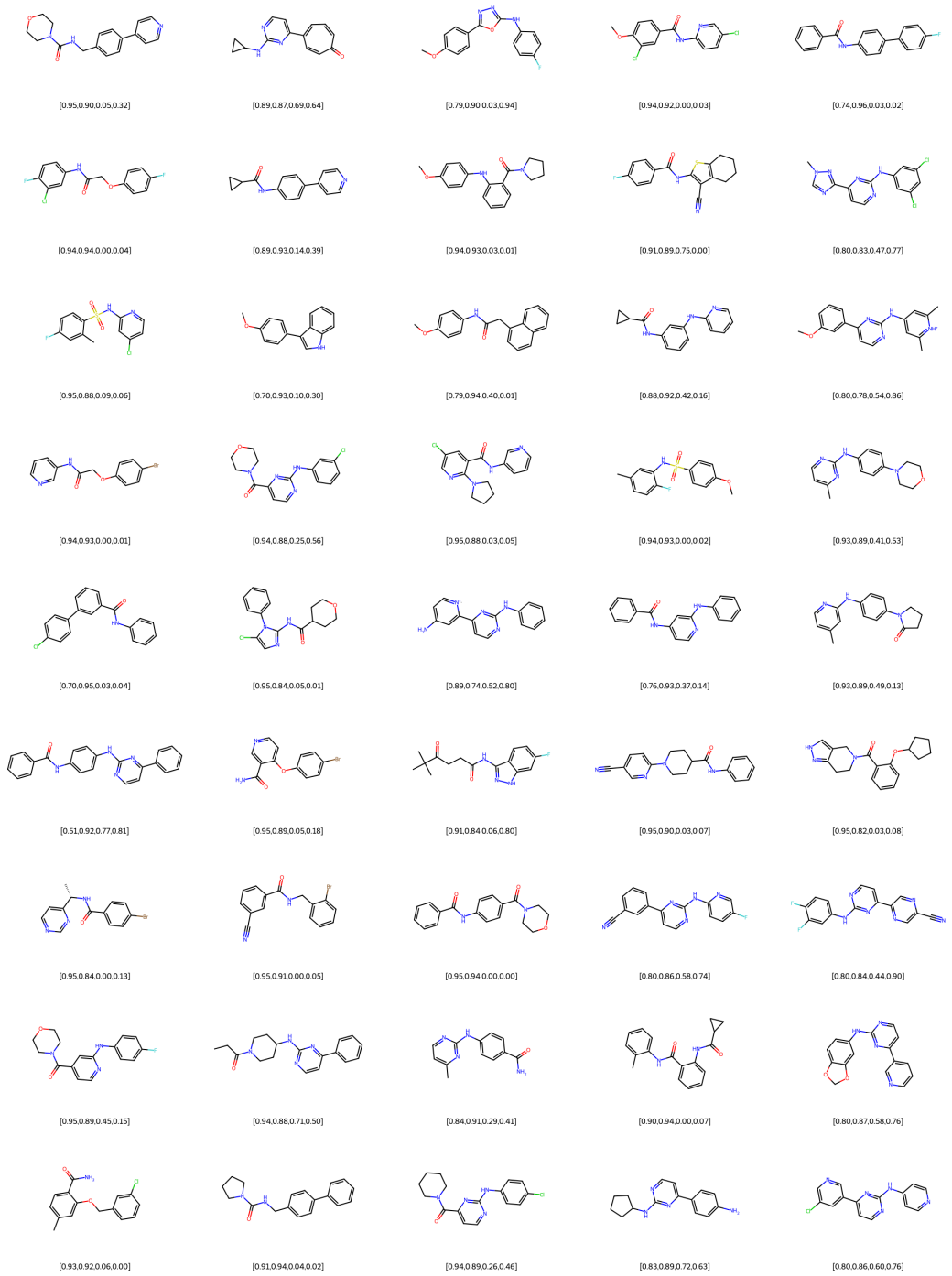


Figure 18: Representative molecules sampled from the Pareto front in the four-objective optimization scenario (QED+SA+JNK3+GSK3β). The numerical scores for each objective are displayed below the respective molecular structures. Each molecule reflects a distinct trade-off among these objectives, highlighting the diverse range of solutions on the Pareto front.