Conditional Latent Space Molecular Scaffold Optimization for Accelerated Molecular Design

Anonymous authors
Paper under double-blind review

Abstract

The rapid discovery of new chemical compounds is essential for advancing global health and developing treatments. While generative models show promise in creating novel molecules, challenges remain in ensuring the real-world applicability of these molecules and finding such molecules efficiently. To address this challenge, we introduce Conditional Latent Space Molecular Scaffold Optimization (CLaSMO), which integrates a Conditional Variational Autoencoder (CVAE) with Latent Space Bayesian Optimization (LSBO) to strategically modify molecules while preserving similarity to the original input, effectively framing the task as constrained optimization. Our LSBO setting improves the sample-efficiency of the molecular optimization, and our modification approach helps us to obtain molecules with higher chances of real-world applicability. CLaSMO explores substructures of molecules in a sample-efficient manner by performing BO in the latent space of a CVAE conditioned on the atomic environment of the molecule to be optimized. Our extensive evaluations across diverse optimization tasks—including rediscovery, docking score, and multi-property optimization—show that CLaSMO efficiently enhances target properties, delivers remarkable sample-efficiency crucial for resource-limited applications while considering molecular similarity constraints, achieves state of the art performance, and maintains practical synthetic accessibility. We also provide an open-source web application that enables chemical experts to apply CLaSMO in a Human-in-the-Loop setting.

1 Introduction

The accelerated discovery of chemical compounds represents a crucial challenge with the potential to revolutionize global health, offering new ways to combat diseases and viruses. The ability to efficiently discover and develop new chemical compounds could lead to groundbreaking treatments and therapies, addressing some of the most pressing health issues of our time. As the importance of this field grows, so too does the research focused on finding effective solutions. Over the past few years, artificial intelligence (AI) has emerged as a powerful tool in this endeavor. The combination of increased computational power and advancements in generative modeling has brought us closer than ever to achieving significant breakthroughs in accelerated discovery.

Generative models offer various approaches to exploring and creating new chemical compounds. A common strategy involves training a generative model on a comprehensive database of chemical compounds. Once trained, the model can generate entirely new compounds (Gómez-Bombarelli et al., 2018; Tripp et al., 2020; Griffiths & Hernández-Lobato, 2020; Boyar & Takeuchi, 2024; Grosnit et al., 2021; Boyar et al., 2024). This approach opens the door to discovering novel molecules that are unlike any currently known, potentially revealing a vast, untapped universe of chemical possibilities. However, while the generation of these novel molecules is exciting, their practical applicability is often constrained. The challenges lie in synthesizing these novel molecules and in the limited understanding of their properties, which makes it difficult for domain experts to assess their viability (Lim et al., 2020).

To address these challenges, another strategy for designing new molecules focuses on modifying/editing existing compounds using generative models (Bradshaw et al., 2019; Lim et al., 2020), reinforcement learning

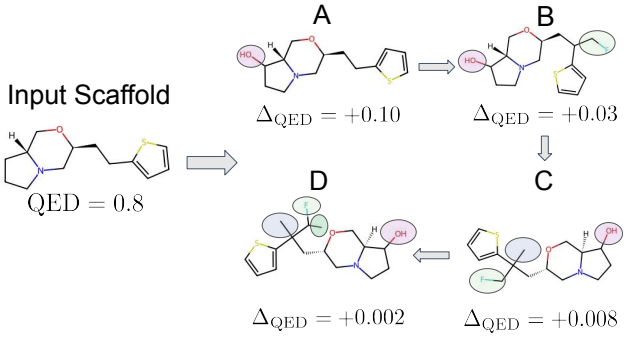

Figure 1: An example of the CLaSMO framework updating a scaffold for Quantitative Estimation of Drug-likeness (QED) optimization. CLaSMO identifies optimal regions in the CVAE latent space and selects bonding points on the scaffold. Chemical features from these points are used to guide the generation of substructures, which are then integrated into the scaffold (A, B, C, D) through small, targeted modifications to improve molecular properties.

(Gottipati et al., 2020), genetic algorithms (Jensen, 2019), or from the domain experts themselves by trial and error (Bemis & Murcko, 1996; Schreiber, 2000; Welsch et al., 2010). These approaches are more likely to produce synthesizable molecules because the base molecule is known to exist and, therefore, can be more tractable for real-world applications. However, even with this strategy, a significant obstacle remains: sample-efficiency, i.e., how efficiently a method finds a promising molecular modification with a limited number of molecular property evaluations. Evaluating the properties of a chemical compound is a time-consuming and costly process, and many existing methods in the literature require numerous trials, making them less practical for accelerated discovery. Consequently, there is a critical need for a methodology that can generate target chemical compounds in a more sample-efficient manner.

In this study, we introduce the Conditional Latent Space Molecular Scaffold Optimization (CLaSMO) method, a framework designed to address two critical challenges in chemical compound discovery: real-world applicability and sample-efficiency. CLaSMO combines a Conditional Variational Autoencoder (CVAE) (Higgins et al., 2016) with Latent Space Bayesian Optimization (LSBO) (Gómez-Bombarelli et al., 2018) to strategically modify input molecules and optimize their chemical properties.

In drug discovery, a common strategy for generating synthesizable molecules is to work with molecular scaffolds—key substructures that serve as the foundation for chemical modifications and drug design (Bemis & Murcko, 1996). Building on this approach, our framework integrates small substructures into these scaffolds to improve key molecular properties. For modifications to be both effective and synthesizable, it's crucial that the generative model understands how new substructures bond with the scaffold and enhance its properties. To achieve this, we condition substructure generation on the scaffold's chemical features using a CVAE. Our novel data preparation and training strategy enables the CVAE to generate substructures that align with specific atomic environments. We further optimize this process using LSBO, which efficiently explores both the latent space of the CVAE and the scaffold's chemical features. This allows CLaSMO to effectively select regions for modification and generate substructures that are chemically meaningful. By tracking molecular similarity between the initial scaffold and the optimized molecule, CLaSMO is designed to operate in low-budget scenarios and costly settings, such as wet-lab experiments, where high sample-efficiency is crucial. It ensures that modifications are both realistic and applicable to real-world contexts. Ultimately, this approach accelerates the discovery of novel, synthesizable compounds with improved properties. Figure 1 illustrates an example of molecular optimization with CLaSMO.

We evaluate our approach on a diverse suite of 20 molecular optimization tasks that span a wide range of objectives, including rediscovery of known compounds, multi-property optimization, and drug-likeness enhancement. These tasks present different challenges, enabling a comprehensive assessment of our model's performance across varied molecular design scenarios. To further test robustness, we incorporate experimental settings with varying levels of similarity constraints between input and optimized molecules, reflecting

both flexible and constrained optimization regimes. In addition to property optimization, we also assess the synthetic accessibility of the generated molecules. Our results show that CLaSMO consistently improves target properties in a sample-efficient manner, demonstrating strong generalizability and effectiveness across these diverse tasks.

1. We propose CLaSMO, a pioneering CVAE and LSBO-based molecule modification algorithm for molecular design. We use a novel data preparation strategy that enables CVAE to learn how substructures bond with target molecules, providing tailored generations.
2. We show that CLaSMO improves target properties with sample-efficiency while keeping the optimized molecules structurally similar to the input scaffolds, increasing the likelihood of identifying synthesizable compounds with desirable properties.
3. We open source a web-application, https://clasmo.streamlit.app/, that enables interactive optimization of input molecules via CLaSMO, which allows chemical experts to decide the region to modify in input molecule, enabling Human-in-the-Loop optimization settings.

2 Related Works

Molecular design strategies can broadly be divided into two categories: from-scratch-generation of molecules and modification-based approaches. Both categories have made significant strides in recent years, yet they also face unique challenges, particularly regarding real-world applicability and sample-efficiency.

From-scratch-generation approaches focus on creating entirely new molecules by conducting a search to optimize the target property. A seminal work by Gómez-Bombarelli et al. (2018) introduced latent space optimization-based methodology, using a VAE to generate novel compounds by navigating the latent space of molecular representations. LSBO builds on this by efficiently reducing the number of expensive black-box evaluations required for molecular optimization, enabling the discovery of compounds with desirable properties in a continuous latent space. Since then, numerous studies have further refined and expanded the LSBO framework, focusing on method development and practical applications (Grosnit et al., 2021; Tripp et al., 2020; Maus et al., 2022; Griffiths & Hernández-Lobato, 2020; Boyar & Takeuchi, 2024; Boyar et al., 2024). However, like many other generation-from-scratch methods, such methodologies struggle with real-world applicability—i.e., the difficulty of synthesizing the generated molecules in real-world settings (Lim et al., 2020). Other generation methods, such as genetic algorithms (Jensen, 2019), Grammar VAE (Kusner et al., 2017), and Junction Tree (JT) VAE (Jin et al., 2018), aim to improve the chemical validity of generated structures. Recent advancements like GP-MOLFORMER (Ross et al., 2024) utilizes large language model-like approaches for molecular design, but the real-world applicability challenge remains a major limitation across these methodologies.

Modification-based approaches, on the other hand, focus on adding substructures to existing molecules or scaffolds, often leading to more synthesizable and interpretable designs. Methods like Scaffold-GGM (Lim et al., 2020) employ a graph generative model to modify molecular scaffolds, thus improving properties with a higher chance of obtaining synthesizable molecules. Weller & Rohs (2024) introduces DrugHIVE, a deep hierarchical variational autoencoder that leverages scaffold modification to generate novel molecular compounds. There are many other scaffold-based optimization methodologies (Li et al., 2019; Langevin et al., 2020), not limited to generative modeling (Schreiber, 2000; Welsch et al., 2010; Miao et al., 2011). Techniques like Bradshaw et al. (2019)'s model ensure chemical validity in molecular modifications, and Gottipati et al. (2020)'s PGFS model uses reinforcement learning to guide additions to the base molecule. These approaches aim to avoid the low real-world viability problem faced by generation-from-scratch methods, as they build upon known molecular scaffolds, however, they lack advanced optimization methodologies that take sample-efficiency into account.

CLaSMO combines the strengths of both categories, leveraging LSBO in the latent space of a CVAE for improved sample-efficiency while focusing on scaffold-based modifications. Unlike current LSBO-based molecular design methodologies in the literature, CLaSMO does not generate molecules from scratch but instead optimizes molecular properties by adding substructures to existing scaffolds. This approach mitigates the

real-world applicability problem, and increases the chance of obtaining molecules that are both effective and practical for real-world synthesis. To evaluate the sample-efficiency of CLaSMO and benchmark it against established methodologies, we follow the sample-efficiency benchmark introduced by Gao et al. (2022). This benchmark provides a suite of challenging molecular property optimization tasks, using a variety of oracle functions, and compares the performance of numerous molecular design methods from the literature.

A key group of methods included in this benchmark are genetic algorithm-based approaches, which remain competitive and widely applicable to scaffold optimization. Notable examples include Graph-GA (Jensen, 2019), which uses graph-based genetic algorithms in combination with a generative model; Smiles-GA (Brown et al., 2019), which operates on SMILES string representations (Weininger, 1988); and STONED (Nigam et al., 2021), which leverages SELFIES (Krenn et al., 2020), a robust alternative to SMILES for molecular encoding. These methods have demonstrated strong performance and robustness, and benchmarking against them is highly recommended, as emphasized in Tripp & Hernández-Lobato (2023).

Beyond genetic algorithms, other scaffold-optimization-capable methodologies have also gained attention. For example, MARS Xie et al. (2021) employs a Markov Chain Monte Carlo approach for molecule generation and editing, and has shown strong performance across various tasks, particularly in multi-property optimization. Additionally, reinforcement learning techniques such as MolDQN (Zhou et al., 2019b) have proven effective for property optimization tasks. Recent advances in GFlowNets (Bengio et al., 2021) have also opened new avenues in molecular design. For instance, SynFlowNet (Cretu et al., 2024), a GFlowNet-based framework, supports scaffold optimization and has demonstrated competitive performance.

We adopt and modify the experimental setting proposed by Gao et al. (2022) in Section 5.2 to better align with the scope of our work—namely, molecular optimization in low-budget scenarios, where oracle evaluations are limited and sample-efficiency is essential. Within this setting, we demonstrate that CLaSMO consistently outperforms baseline methodologies across a diverse set of tasks. By combining a targeted scaffold-modification strategy with efficient search, CLaSMO offers a balanced and effective solution to molecular optimization under constrained evaluation budgets.

3 Preliminaries and Problem Setup

In this section, we provide preliminary knowledge on CVAEs, LSBO, and scaffolds. We then discuss the challenges of property optimization and scaffold modifications.

3.1 Conditional Variational Autoencoders (CVAEs)

A VAE (Kingma & Welling, 2014) consists of an encoder $f_\phi^{\text{enc}} : \mathcal{X} \to \mathcal{Z}$ and a decoder $f_\theta^{\text{dec}} : \mathcal{Z} \to \mathcal{X}$, where $\mathcal{X}$ represents the input space and $\mathcal{Z}$ the latent space. CVAEs extend the framework of VAEs by incorporating additional condition vector $\boldsymbol{c}$ into the latent variable model, facilitating the controlled generation of new instances. In the CVAE architecture, the encoder $q_\phi(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{c})$ maps an input $\boldsymbol{x}$ and a condition $\boldsymbol{c}$ to a latent representation $\boldsymbol{z}$. Simultaneously, the decoder $p_\theta(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{c})$ reconstructs $\boldsymbol{x}$ using both $\boldsymbol{z}$ and $\boldsymbol{c}$. The training of CVAEs is formulated as the minimization of the conditional variational lower bound:

$$\mathcal{L}(\theta, \phi; \boldsymbol{x}, \boldsymbol{c}) = -\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{c})} \left[\log p_\theta(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{c})\right] + \text{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{c}) \parallel p(\boldsymbol{z})\right), \tag{1}$$

where KL denotes the Kullback-Leibler divergence. In this model, the prior distribution $p(\boldsymbol{z})$ over the latent variables is typically assumed to be a standard normal distribution, $\mathcal{N}(0, I)$. This assumption simplifies the learning process by standardizing the latent space, ensuring that the encoder learns a distribution that closely aligns with a prior distribution, thus enhancing the generative capability of the decoder conditioned on specific contexts.

3.2 Latent Space Bayesian Optimization (LSBO)

In BO, we start with numerous *unlabeled* instances $\{\boldsymbol{x}_i\}_{i\in[\mathcal{U}]}$ and a smaller set of *labeled* instances $(\boldsymbol{x}_i, y_i)_{i\in[\mathcal{L}]}$, where an input $\boldsymbol{x}_i \in \mathcal{X}$ represents a chemical compound, and a label $y_i \in \mathcal{Y} \subseteq \mathcal{R}$ indicates its properties such as docking scores. BO seeks to optimize a costly black-box function $f^{\text{BB}} : \mathcal{X} \to \mathcal{Y}$, which corresponds to

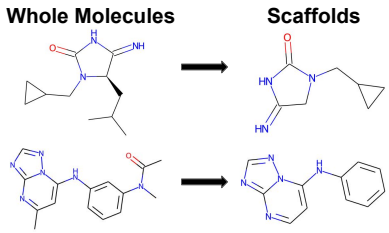

Figure 2: Examples of scaffold extraction from whole molecules. In both the upper and lower rows, side chains are removed, leaving the core structure of the molecule. These resulting scaffolds act as starting points for novel chemical design.

obtaining the physical properties of chemical compounds through experiments or time-consuming simulations in the context of molecular design problems. The goal is to maximize f^{BB} with minimal evaluations, using typically a Gaussian Process (GP) surrogate, trained using the labeled instances $\mathcal{L}$, to predict the function over $\mathcal{X}$. BO uses the surrogate to select an input $\boldsymbol{x}$ that may yield values surpassing the current maximum $\max_{i \in \mathcal{L}} y_i$. However, building a GP surrogate in high-dimensional spaces like chemical compounds is challenging. LSBO tackles this by employing a VAE/CVAE trained on the unlabelled instances $\mathcal{U}$ to reduce dimensionality, encoding instance in $\mathcal{X}$ to lower dimensional latent space $\mathcal{Z}$. This simplifies surrogate modeling and optimization because $\mathcal{Z}$ has lower-dimension than $\mathcal{X}$. During LSBO iterations, the acquisition function applied to the GP's predictions selects new points in $\mathcal{Z}$ to evaluate. The chosen latent variable $\boldsymbol{z}_{i'}$ is decoded into a new input $\boldsymbol{x}_{i'} = f_\theta^{\mathrm{dec}}(\boldsymbol{z}_{i'})$. This new instance is evaluated by f^{BB}, and the results update $\mathcal{L}$ and refine the GP model. This cycle repeats until optimal results are achieved or resources are exhausted. In contexts like molecular design, LSBO aims to discover chemical compounds with optimal properties by efficiently navigating the reduced latent space.

3.3 Scaffolds

Scaffolds (Bemis & Murcko, 1996) are the stable core structures within molecules that serve as the framework for chemical modifications in drug design. Scaffolds retain the essential biological activity of the molecule. They play a crucial role in molecular design by providing a foundation for chemical modifications aimed at optimizing properties like QED. Researchers often use scaffolds to systematically explore chemical variations (Schreiber, 2000; Welsch et al., 2010), which can lead to the discovery of new compounds with improved properties.

In this study, we follow the scaffold extraction method from Bemis & Murcko (1996), where non-essential components like side chains are removed, leaving the core structure. This extracted scaffold serves as a starting point for further modifications, allowing efficient exploration of chemical space. By focusing on scaffolds, molecular design becomes more streamlined, increasing the likelihood of identifying novel, synthesizable compounds with the desired biological activity. Examples of whole molecule and scaffold pairs are provided in Fig. 2.

3.4 Problem Definition

We denote the scaffold, which is the base of the modification, as $\boldsymbol{S}$, and the modified molecule as $\boldsymbol{S}'$. Our goal is to efficiently find the modification that maximizes the molecular property P which is evaluated by $f^{\mathrm{BB}}(\boldsymbol{S}')$, while keeping the difference between $\boldsymbol{S}$ and $\boldsymbol{S}'$ small. Directly iterating over all possible $\boldsymbol{S}'$ to find the best modification that maximizes $f^{\mathrm{BB}}(\boldsymbol{S}')$ is impractical, as it involves high complexity and costly evaluations of f^{BB}.

The primary challenge in optimizing molecular scaffolds lies in i) determining the optimal bonding point on the base scaffold $\boldsymbol{S}$, and ii) selecting the appropriate substructures added to the bonding point to ensure meaningful improvements in the desired property $\mathcal{P}$. A molecular scaffold $\boldsymbol{S}$, composed of several atoms $p_1, p_2, \ldots, p_k$, may have atoms with the remaining capacity to form additional chemical bonds. These atoms

serve as potential candidates for bonding with newly generated substructures. Therefore, the task involves not only selecting the right substructure but also identifying the most suitable bonding point p_i to optimize scaffold properties. This adds complexity, as the need for precise modifications must be balanced with the challenges of high-dimensional search spaces and evaluation costs. Consequently, a more efficient approach is required to explore scaffold modifications effectively while minimizing the number of evaluations.

To address this, the problem can be reframed as an optimization task in a reduced latent space $\mathcal{Z}$, obtained through a CVAE. In this space, each point $\boldsymbol{z} \in \mathcal{Z}$ corresponds to a potential substructure that can be integrated into the scaffold. By encoding the molecular substructures into this lower-dimensional space $\mathcal{Z} \in \mathbb{R}^d$, the search becomes more tractable. The objective is to find the optimal latent representation $\boldsymbol{z^*}$ conditioned on the optimal bonding point p_i^* that, together, maximize the desired property $\mathcal{P}$ when the generated substructure $\boldsymbol{s'} \leftarrow f^{\text{dec}}(\boldsymbol{z})$ is added to the scaffold. Let us denote this modification as $\boldsymbol{S}^{'} \leftarrow g(\boldsymbol{S} \oplus \boldsymbol{s'}, p_i)$, where the function $g()$ adds substructure $\boldsymbol{s}$ to the scaffold $\boldsymbol{S}$ at p_i. The optimization problem is then formulated as:

$$\boldsymbol{z^*}, p_i^* = \arg \max_{\boldsymbol{z} \in \mathcal{Z}, p_i \in B(\mathbf{S})} f^{\text{BB}}(\boldsymbol{S}^{'}) = \arg \max_{\boldsymbol{z} \in \mathcal{Z}, p_i \in B(\mathbf{S})} f^{\text{BB}} g(\boldsymbol{S} \oplus f^{\text{dec}}(\boldsymbol{z}), p_i), \tag{2}$$

where $B(\boldsymbol{S})$ is the set of possible bonding points on the scaffold $\boldsymbol{S}$. In Section 4.2, we demonstrate that atomic features at p_i are used as condition vectors to generate new substructures, enabling targeted substructure generation for atom p_i.

3.4.1 Controlling Molecular Similarity

Current modification-based methods often fail to account for how changes impact molecular similarity between the original scaffold $\boldsymbol{S}$ and the updated scaffold $\boldsymbol{S'}$, or any structure in general. Adding substructures typically increases molecular weight, which can hinder real-world applicability, especially when exceeding 500 Daltons, as indicated by Lipinski's Rule of Five (Lipinski et al., 2001). Higher molecular weight compounds are more difficult to synthesize, making them less suitable for molecular design. Additionally, it is sometimes necessary to ensure that modifications result in only minor adjustments to avoid drastic changes.

Thus, a key challenge is to guide the optimization process by considering molecular similarity, ensuring that the modified molecules remain structurally close to the original scaffold. Such a framework can increase the likelihood of obtaining synthesizable compounds by limiting divergence from known molecules. In Section 5, we show that our method effectively solves the optimization problem in Eq. (2) while ensuring molecular similarity between $\boldsymbol{S}$ and $\boldsymbol{S'}$.

4 Proposed Method

Our proposed CLaSMO framework comprises two key components: the CVAE and the LSBO algorithm. However, an essential first step in our approach is the data preparation required to train the CVAE. In this section, we will begin by outlining the data preparation process, followed by an explanation of the CVAE and the CLaSMO methodology.

4.1 Data Preparation

Our proposed method requires a uniquely tailored dataset because no existing dataset in the literature fully meets the specific needs of our approach. To create this dataset, we developed a BRICS (*Breaking Retrosynthetically Interesting Chemical Substructures*) (Degen et al., 2008) based approach. BRICS is an algorithm designed to decompose organic molecules into smaller, synthetically feasible substructures by identifying breaking points within a molecule's structure based on chemical retrosynthetic rules. These breaking points, also known as division points, are the connections between subgraphs within the molecule that BRICS identifies. The algorithm systematically breaks down a molecule $\boldsymbol{M}$ into k substructures, Fig. 3 demonstrates this procedure. BRICS is particularly well-suited for this task because it ensures that the generated substructures are synthetically feasible and chemically valid, making it ideal for scaffold-based molecular optimization. Unlike other decomposition methods, BRICS explicitly adheres to retrosynthetic rules, ensuring compatibility with real-world chemical synthesis.

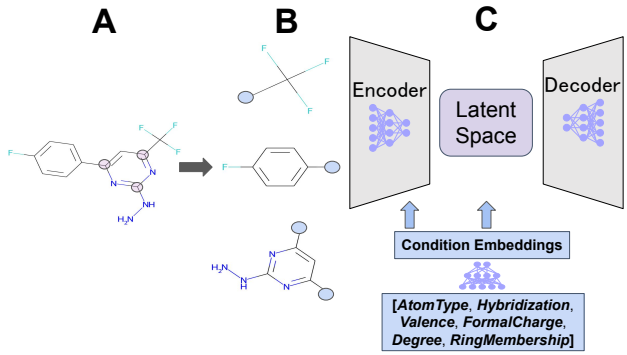

Figure 3: Illustration of the BRICS algorithm and its integration with the CVAE. The molecule in **A** is decomposed into substructures using the BRICS algorithm (**B**), with specific breaking points highlighted. Atomic environment features are extracted from these points to provide critical information about the bonding environment. Embeddings of these features are used as condition vectors, which are concatenated both at the encoder input along with the substructures and at the latent space of the CVAE (**C**), guiding the generation of substructures that are compatible with the scaffold.

The division points found by BRICS are of particular importance because they serve as both the points where the molecule is split into substructures and the potential bonding sites where these substructures can be reattached. Therefore, they provide us a valuable information about these substructures in terms of what kind of bonds they can form. This dual role makes the division points a crucial piece of information for our generative model. By preserving the properties and features of atoms at these division points, we capture the chemical context around the atom that dictates how substructures can bond with other parts of a molecule. The atomic features we consider are *atom type, hybridization, valence, formal charge, degree, and ring membership*. Detailed explanations of these features are provided in the Appendix A.1. These features, when used as condition vectors in our CVAE, are essential for guiding the generation of substructures that can successfully bond with the scaffold, as they provide a detailed understanding of the chemical environment at each division point, such as their atom types and if the selected atom has the capacity for forming additional bonds. Therefore, our dataset for CVAE training includes the substructures obtained via the BRICS algorithm and six different atomic features of the atom at their breaking points. These substructures are represented using a string-based representation method referred to as SELFIES Krenn et al. (2020). The illustration of the BRICS algorithm, atomic feature extraction points, and their integration with CVAE is provided in Fig. 3.

4.2 Substructure CVAE

Our CVAE consists of an encoder, parameterized by f_ϕ^{enc}, and a decoder, parameterized by f_θ^{dec}. The encoder takes the substructure $\boldsymbol{s}$ and the associated condition vector $\boldsymbol{c}$, and maps this input into a latent space, producing a latent representation $\boldsymbol{z}$. The decoder then takes a point $\boldsymbol{z}$ from this latent space, along with the condition vector $\boldsymbol{c}$, and generates a substructure $\boldsymbol{s}'$ that is conditioned on the given atomic properties. The loss function of the proposed CVAE is defined as:

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}; \mathrm{s}, \boldsymbol{c}) = \|\boldsymbol{s} - \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{c})}[p_{\boldsymbol{\theta}}(\boldsymbol{s}|\boldsymbol{z}, \boldsymbol{c})]\|^2 + \beta \cdot D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{s}, \boldsymbol{c}) \| p(\boldsymbol{z})), \tag{3}$$

where $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{s}, \boldsymbol{c})$ is the encoder, $p_{\boldsymbol{\theta}}(\boldsymbol{s}|\boldsymbol{z}, \boldsymbol{c})$ is the decoder, and β balances the reconstruction and KL divergence terms (Higgins et al., 2016). The condition vectors allow the CVAE to learn how different substructures interact with specific atomic environments, building a deeper understanding of their bonding behavior. Figure 3B demonstrates the input substructures that CVAE uses in its training, and Fig. 3C visualizes the CVAE architecture along with the condition vectors.

After training, if during the generation phase, we provide the CVAE with a condition vector that specifies the atomic environment of the target scaffold, the decoder, using a latent vector along with the condition vector, generates substructures that are tailored to bond effectively with the scaffold. This conditioning mechanism

ensures that the generated substructures are not random but specifically designed to fit the scaffold's atomic environment, improving the efficiency of scaffold modifications and the likelihood of successful bonding. In the proposed CLaSMO method, we jointly optimize the bonding position p_i and the latent vector $\boldsymbol{z}$, conditioned on the atomic properties $\boldsymbol{c}$ of the bonding site in order to optimize both the bonding position and the substructure added to the position.

4.2.1 Condition Vector Embeddings

In Section 4.1, we detailed our data preparation process and the extraction of six atomic environmental features, resulting in a 6-dimensional condition vector. As we will explain in Section 5.1, the relatively simple nature of the substructures used in CVAE training allows for a much lower latent dimension than the actual conditional vector dimension of 6. To align with this simplicity and ensure efficient representation, we employed an Autoencoder model to generate d-dimensional embeddings of the atomic features. These embeddings preserve the key characteristics within the atomic environments and learn the relationship among distinct atomic features while reducing dimensionality, which helps lower computational complexity and improves the model's ability to generalize. After its training, the encoder of the Autoencoder, $f^{\text{enc}}_{\text{CondEmd}}$, is used to provide the final condition vector $\boldsymbol{c}$ to be inputted to CVAE by encoding the six-dimensional atomic environmental feature vector. This process is visualized in Fig. 3C.

4.3 CLaSMO

In CLaSMO, upon training of the CVAE, we perform a targeted search in the latent space $\mathcal{Z} \in \mathbb{R}^d$ of the CVAE, where decodings from each point $\boldsymbol{z}$ represent a potential substructure to be added to the scaffold. The challenge is to modify the scaffold $\boldsymbol{S}$ by selecting a substructure and integrating it at an appropriate bonding point p_i, optimizing the desired molecular property $\mathcal{P}$. Therefore, the search space Ω we consider in our optimization tasks is defined as the Cartesian product of the latent space and the set of possible bonding points on the scaffold:

$$\Omega = \mathbb{R}^d \times \{p_1, p_2, \ldots, p_n\}.$$

Within this search space, the goal is to modify the scaffold $\boldsymbol{S}$ to maximize the BB function $f^{\text{BB}}(\boldsymbol{S'})$.

However, although our main goal is optimizing the target property, we also aim to keep the similarity between $\boldsymbol{S}$ and $\boldsymbol{S'}$ at a certain level. Therefore, in CLaSMO, we apply a similarity constraint using the *Dice Similarity* (Dice, 1945) metric to compare the input scaffold $\boldsymbol{S}$ with the modified molecule $\boldsymbol{S'}$ before evaluating the BB function. In order to measure the similarity between $\boldsymbol{S}$ and $\boldsymbol{S'}$, we use Morgan Fingerprints (Morgan, 1965; Rogers & Hahn, 2010). The Morgan fingerprint of a molecule is represented as a binary vector, where each bit indicates the presence or absence of a specific substructure within the molecule, making them effective and popular for comparing molecular similarities. Using these, the similarity between $\boldsymbol{S}$ and $\boldsymbol{S'}$ is measured by computing the Dice Similarity of their Morgan fingerprint vectors[1], defined as:

$$\text{DICE}(\boldsymbol{S}, \boldsymbol{S'}) = \frac{2|\boldsymbol{M_S} \cap \boldsymbol{M'_S}|}{|\boldsymbol{M_S}| + |\boldsymbol{M'_S}|}$$

where $\boldsymbol{M_S}$ and $\boldsymbol{M'_S}$ are the Morgan fingerprint vectors of $\boldsymbol{S}$ and $\boldsymbol{S'}$, respectively. Dice Similarity, ranging from 0 (no overlap) to 1 (identical), helps us track structural changes during optimization.

The final optimization objective, incorporating both the search for optimal substructures and the similarity constraint, is given by:

$$\boldsymbol{z^*}, p_i^* = \arg \max_{(\boldsymbol{z}, p_i) \in \Omega} f^{\text{BB}}(\boldsymbol{S'}) \tag{4}$$

$$\text{subject to:} \quad \text{DICE}(\boldsymbol{S}, \boldsymbol{S'}) \geq \tau,$$

where $\boldsymbol{S}^{'} \leftarrow g(\boldsymbol{S} \oplus f^{\text{dec}}(\boldsymbol{s'}|\boldsymbol{z}^*, \boldsymbol{c}))$. By this approach, CLaSMO efficiently navigates the latent space, optimizing molecular properties while allowing modifications only if $\text{DICE}(\boldsymbol{S}, \boldsymbol{S'}) \geq \tau$, effectively controlling the degree of divergence from the input scaffold.

[1]Bajusz et al. (2015) discusses that the dice similarity is one of the best metrics to assess similarity between molecules using Morgan fingerprints.

For the joint optimization of substructures and bonding positions, we consider a GP model: $(\boldsymbol{z}, p) \mapsto y'_\Delta$, where $\boldsymbol{z}$ is the latent vector representing a substructure, p is the bonding position of the modification, and $y'_\Delta := y - y'$ represents the improvement in the property, with y and y' indicating the properties of molecules S and S', respectively. To prepare the training dataset $\mathcal{D}$ for the GP, we conducted a random sampling of substructures from random latent vectors $\boldsymbol{z}$, each paired with randomly selected bonding regions p on the scaffold $\boldsymbol{S}$, and evaluated the y'_Δ obtained from these additions (e.g., in the case of QED optimization, QED score differences between $\boldsymbol{S}$ and $\boldsymbol{S'}$ are calculated). These triplets of $(\boldsymbol{z}, p, y'_\Delta)$ are used in GP training. This setup allows the GP model to learn the relationship between the latent space representations, atoms in the input scaffold, and the resulting property changes, guiding the optimization process toward regions of the latent space that are more likely to yield beneficial modifications to the scaffold.

Among many possible choices, we employ the Upper Confidence Bound (UCB) acquisition function to guide the optimization process. To ensure LSBO samples from regions that meet the similarity constraint, we introduced a penalization mechanism. Specifically, we assign a negative improvement value y'_Δ when the condition $\mathrm{DICE}(\boldsymbol{S}, \boldsymbol{S'}) \geq \tau$ is not satisfied after sampling from $f^{\mathrm{dec}}([\boldsymbol{z}^*, \boldsymbol{c}])$. Additionally, we also apply a penalty when a substructure cannot bond with the target molecule. We outline our approach in Algorithm 1, and provide details about the selection of hyperparameters within our framework in the Appendix A.3.

4.3.1 Kernel Design of CLaSMO

The problem we try to solve requires simultaneous optimization over continuous latent vectors and discrete bonding points. To handle this complexity, we use a GP model with a covariance function k that accommodates the mixed input space of continuous and discrete variables. We define separate kernels for these inputs:

$$k_{\mathrm{cont}}(\boldsymbol{z}, \boldsymbol{z'}) = \exp\left(-\frac{1}{2\ell^2}\|\boldsymbol{z} - \boldsymbol{z'}\|^2\right), \quad k_{\mathrm{cat}}(p_i, p_j) = \exp\left(-\frac{\delta_{p_i,p_j}}{\ell}\right),$$

where $k_{\mathrm{cont}}(\boldsymbol{z}, \boldsymbol{z'})$ is an RBF kernel, and $k_{\mathrm{cat}}(p_i, p_j)$ measures the similarity between atoms in the molecule that are ready for additional bond via Kronecker delta function, measuring equality of atom positions, and ℓ is the lengthscale parameter. The combined kernel used in CLaSMO is then expressed as:

$$k_{\mathrm{CLaSMO}} = k((\boldsymbol{z}, p_i), (\boldsymbol{z'}, p_j)) = k_{\mathrm{cont}}(\boldsymbol{z}, \boldsymbol{z'}) \cdot k_{\mathrm{cat}}(p_i, p_j).$$

5 Experiments

In this section, we first describe the training procedure for our CVAE model (Section 5.1). We then evaluate the performance of CLaSMO across 20 molecular property optimization tasks Gao et al. (2022), which span a range of objectives including rediscovery of known compounds, similarity-based optimization, multi-property optimization, and improvements in drug-likeness.

To assess robustness across different optimization regimes, each task is repeated under three levels of molecular similarity constraints between the input and optimized molecules, defined by Dice similarity thresholds of 0.50, 0.25, and 0. This results in a total of 60 optimization settings. Higher thresholds enforce stricter structural similarity, enabling controlled, local edits, while lower thresholds allow for broader exploration of chemical space. We also evaluate the synthetic accessibility of the generated molecules to assess their feasibility from a chemical synthesis perspective.

Section 5.2 presents the results of these optimization experiments and the corresponding synthetic accessibility analysis. In Section 5.3, we provide an ablation study to examine the impact of conditional generation. Finally, Section 5.4 discusses the extension of CLaSMO to a human-in-the-loop optimization setting.

5.1 CVAE Training

For our data preparation, we used the QM9 dataset (Ruddigkeit et al., 2012; Ramakrishnan et al., 2014), which contains 130,000 small molecules. QM9 was chosen for its simplicity and suitability for our task, as it consists of smaller molecules compared to other well-known datasets like ZINC (Irwin & Shoichet, 2005;

Algorithm 1 CLaSMO

Input: GP training data $\mathcal{D}$, Trained CVAE, Trained Autoencoder encoder $f^{\text{enc}}_{\text{CondEmd}}$ for condition embeddings, Input molecules $\boldsymbol{M}$, Optimization budget per molecule K, Similarity threshold τ, Penalization terms λ_1, λ_2
Fit GP using $\mathcal{D}$
for $\boldsymbol{i} = 1$ to $\boldsymbol{M}$ **do**
  Pick molecule $\boldsymbol{m}_i$, obtain its scaffold $\boldsymbol{S}_i \leftarrow \text{Scaffold}(\boldsymbol{m}_i)$,
  Evaluate target property value $y \leftarrow f^{\text{BB}}(\boldsymbol{m}_i)$
  for $j = 1$ to K **do**
    Identify available atoms in the scaffolds for bonding, $p_j \in B(\boldsymbol{S})$[2]
    Find $\boldsymbol{z^*}, p_i^* = \arg\max_{(\boldsymbol{z}_i, p_i) \in \Omega} f^{\text{BB}}(S')$
    Create condition vector $\boldsymbol{c^*}$ for atom p_i^* in molecule $\boldsymbol{S}_i$
    Obtain condition embeddings $\boldsymbol{c} \leftarrow f^{\text{enc}}_{\text{CondEmd}}(\boldsymbol{c^*})$
    Generate substructure $\boldsymbol{s^*} \leftarrow f^{\text{dec}}([\boldsymbol{z}^*, \boldsymbol{c}])$
    Add substructure $\boldsymbol{s^*}$ to molecule $\boldsymbol{S}_i$ at region p_i^* to obtain $\boldsymbol{S}'_i$
    if $\boldsymbol{S}_i \neq \boldsymbol{S}'_i$ **then**
      if $\text{DICE}(\boldsymbol{S}_i, \boldsymbol{S}'_i) > \tau$ **then**
        Evaluate new property: $y' = f^{\text{BB}}(\boldsymbol{S}'_i)$
        Compute improvement $y'_\Delta = (y - y')$
        if $y'_\Delta > 0$ **then**
          Update $\boldsymbol{S}_i \leftarrow \boldsymbol{S}'_i$
        end if
      else
        Set y'_Δ to penalization term λ_1
      end if
    else
      Set y'_Δ to penalization term λ_2
    end if
    Update $\mathcal{D} \leftarrow \mathcal{D} \cup \{[\boldsymbol{z}^*, p^*], y'_\Delta\}$
    Update GP with $\mathcal{D}$
  end for
end for

Irwin et al., 2020). Using our data preparation strategy, we extracted 18,706 unique substructure pairs along with their atomic environment features to train our CVAE. Most of the substructures obtained from QM9 through our method are under 100 Daltons, making them ideal for training a model focused on generating small substructures to modify the input scaffold. Since our goal is to optimize the scaffold while maintaining a high degree of similarity in the modifications, using such a dataset allows CVAE to learn to generate small substructures, which is crucial for achieving our goals. Of the 18,706 instances, 80% were used for model training, with the remaining data allocated for testing and validation. We represented the molecules using SELFIES (Krenn et al., 2020), a string-based molecular representation, in the form of one-hot encoding matrices. Our CVAE architecture consists of three fully connected layers in the encoder and three GRU layers in the decoder. See Appendix A.3 for the selection of hyperparameters in training the CVAE model.

Given the simplicity of the dataset, we determined that a 2-dimensional latent space was sufficient to achieve over 99% reconstruction accuracy on the test set. This lower-dimensional space also enhances the performance of BO, which typically excels in smaller-dimensional spaces. Additionally, to maintain this simplicity, as explained in Section 4.2.1, we generated a 2-dimensional embedding for the condition vectors using an Autoencoder model, which is trained with fully connected layers in both the encoder and decoder, achieved 93% reconstruction accuracy for the condition vectors. Prior to CVAE training, we obtained the condition

[2]This identification is performed by finding atoms within the molecule that have additional capacity to form bonds. For example, a carbon atom can form up to four bonds, and if it has only three bonds within the molecule, it will be identified as available for bonding.

vector embeddings via the Autoencoder model, and used them during CVAE training. As a result, the CVAE was trained with both a 2-dimensional latent space for the substructures and 2-dimensional condition vectors, effectively balancing simplicity and optimization performance. Further discussion on this design choice is provided in Appendix A.2.

5.2 Molecular Optimization Experiments

To comprehensively evaluate the sample-efficiency of our methodology, we applied CLaSMO to the sample-efficiency benchmark proposed by Gao et al. (2022), where we use 20 diverse molecular optimization tasks. These tasks span various objectives, including a broad range of pharmaceutically-relevant oracle functions, and challenging multi-property optimization (MPO) scenarios (Brown et al., 2019). Instead of limiting the evaluation to narrowly defined tasks, this broad selection enables a more comprehensive assessment of each method's generalizability and adaptability across chemically and functionally diverse scenarios.

Here, sample-efficiency refers to the efficiency with respect to the number of evaluations of the black-box function for the target optimization or editing task. For instance, when evaluating molecules through wet-lab experiments, it is often the case that no more than about 100 evaluations can be conducted. Our proposed method, CLaSMO, is specifically designed to achieve such sample-efficient molecular editing. On the other hand, in the numerical experiments presented in this paper, we selected problems where the evaluation of the target black-box functions can be performed at relatively low computational cost, allowing for repeated and comprehensive comparisons between the proposed method and existing methods. Hereafter, we refer to these functions as oracle functions. Experiments were conducted with a fixed budget of 100 oracle evaluations per seed across 10 seeds, ensuring consistent comparisons. The choice of 100 oracle iterations (in contrast to the 10,000 iterations used in Gao et al. (2022)) reflects our focus on developing a methodology tailored for resource-constrained scenarios, which are more representative of real-world applications.

CLaSMO was benchmarked against six widely used baseline methods—Smiles-GA (Brown et al., 2019), Graph-GA (Jensen, 2019), Stoned (Nigam et al., 2021), MolDQN (Zhou et al., 2019a), MARS (Xie et al., 2021), and SynFlowNet (Cretu et al., 2024)—using the same set of 100 scaffolds for all methods as starting molecules. As explained below, different similarity constraints were applied, and open-source implementations of all baseline methods were used to evaluate their ability to perform molecular optimization while preserving the given scaffold. We note that some of these baseline methods were not designed for sample-efficient settings, and therefore may not perform well under the problem settings aimed at sample-efficient molecular editing, which is the focus of this study.[3]

5.2.1 Oracle Definitions and Grouping

To facilitate clearer comparisons and highlight performance patterns, we group the 20 oracles used in our experimental setting into four task-oriented categories. The groups and their associated oracle functions are as follows:

- **Group 1: Similarity and Rediscovery Tasks** This group comprises oracles that either assess molecular similarity to a known drug or target the rediscovery of a known molecule. The oracles in this group include: *albuterol_similarity, celecoxib_rediscovery, mestranol_similarity, thiothixene_rediscovery, troglitazone_rediscovery.*
- **Group 2: Multi-Property Optimization (MPO) Tasks** These involve optimizing multiple molecular properties simultaneously. The oracles in this group include: *osimertinib_mpo, perindopril_mpo, ranolazine_mpo, sitagliptin_mpo, zaleplon_mpo.*
- **Group 3: Structural Modification Tasks** This group encompasses tasks that evaluate a model's ability to generate structurally diverse molecules while preserving specific chemical features. The

[3]Among the competing approaches, Graph-GA, Smiles-GA, and Stoned lack a mechanism to ensure preservation of the given scaffold. We implemented a hyperparameter optimization for these genetic algorithm-based approaches to find the best balance for exploration of the search space and scaffold-preserving capabilities. Please refer to Appendix A.4 for further details on experimental setting.

hopping tasks challenge models to modify the molecule while aiming to maintain or enhance biological activity despite structural changes. In contrast, the *isomers* tasks focus on generating different structural arrangements (isomers) of a molecule that share the same molecular formula but differ in connectivity or spatial orientation. Unlike similarity-based tasks that prioritize closeness to a reference molecule, these tasks emphasize structural innovation and diversity, assessing a model's capacity to explore novel chemical spaces while retaining desired properties. The oracles in this group include: *deco_hop, scaffold_hop, isomers_c7h8n2o2, isomers_c9h10n2o2pf2cl.*

- **Group 4: Activity and Benchmark Tasks** This group includes oracles that predict biological activity or are part of broader benchmarking suites for drug-likeness and diversity. The oracles in this group include: *drd2, gsk3b, median1, median2, qed, valsartan_smarts.*

Further details and descriptions of these oracles are provided in Appendix A.5.[4]

5.2.2 Property Optimization Results

In this section, we discuss the performance of CLaSMO and benchmark methodologies under three similarity-constraint regimes. In the high constraint setting ($\tau = 0.50$), the optimized molecule must have a Dice similarity of at least 0.50 to the input scaffold. In the moderate constraint setting ($\tau = 0.25$), a similarity of at least 0.25 is required, allowing for more flexibility in modifying the scaffold. In these settings, even if the target property can be substantially improved, the result will not be accepted if the similarity falls below the specified threshold. Lastly, in the no constraint setting ($\tau = 0$), there are no restrictions on similarity, and any substructure additions to the scaffold are allowed.

Optimization with High Constraints (Similarity Threshold $\tau = 0.50$) Figure 4 shows the performance of the methods when the similarity constraint is very strict.

- In the **Similarity and Rediscovery Tasks**, CLaSMO is consistently the top performer, providing the top performance from 5 tasks out of 5.
- Within the **MPO Tasks**, CLaSMO and MARS methodologies are the two top performers. MARS, a method that is specifically designed for MPO settings, outperforms CLaSMO in two out of 5 MPO objectives, and provides a similar performance in *osimertinib_mpo* task.
- For the **Structural Tasks**, CLaSMO gives the best results in *deco_hop, scaffold_hop* and *isomers_c7h8n2o2* while MARS achieves a similar result with CLaSMO in the remaining *isomers_c9h10n2o2pf2cl* task.
- In the **Activity and Benchmark Tasks**, CLaSMO records the highest values for *drd2, median1* and *qed*, while losing to MARS in *gsk3b* and *median2*. None of the methodologies provided an improvement in *valsartan_smarts* task.

Overall, under high constraints, CLaSMO proves to be the best approach, providing the highest Top-10 average scores in 13 out of 20 property optimization tasks.

Optimization with Moderate Constraints (Similarity Threshold $\tau = 0.25$) Figure 5 summarizes the results under a moderate similarity constraint, which allows for more chemical modification while still preserving significant scaffold features.

- In the **Similarity and Rediscovery Tasks**, CLaSMO maintains its lead, where it achieves the highest Top-10 average scores in 4 out of 5 tasks, losing to MARS only in *thiothixene_rediscovery.*
- Among the **MPO Tasks**, MARS is the top performer, where it provides the highest Top-10 average scores in 3 out of 5 tasks. CLaSMO outperforms MARS in only *zaleplon_mpo* task, and performs similarly with MARS in *osimertinib_mpo* task.

[4]See also the Therapeutics Data Commons for more detail: `https://tdcommons.ai/functions/oracles/`.

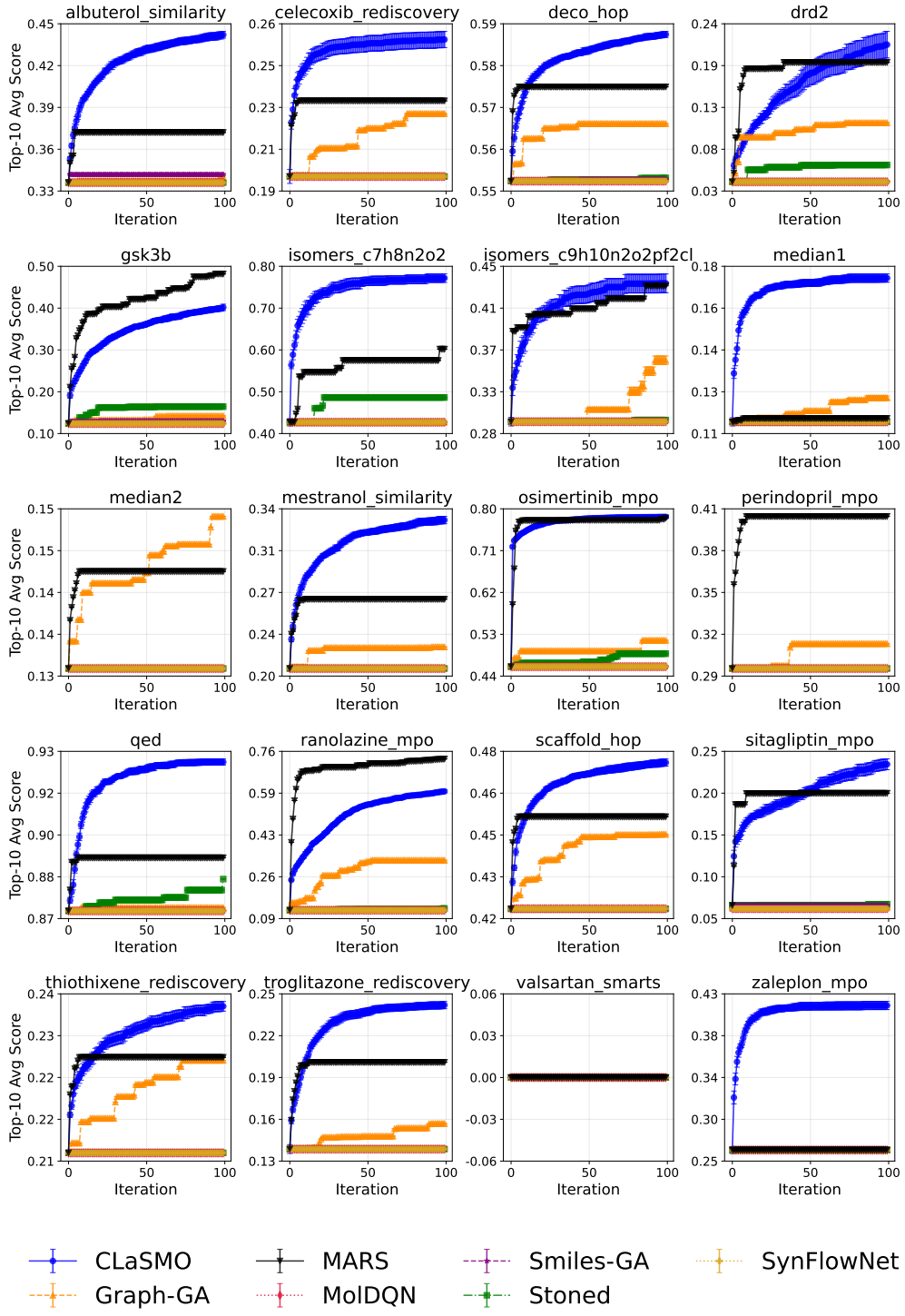

Figure 4: Comparison of Top-10 average values obtained at each iteration for CLaSMO, MARS, Smiles-GA, SynFlowNet, Graph-GA, MolDQN, and Stoned using constraint optimization setting, where Dice Similarity threshold $\tau = 0.50$. CLaSMO identifies molecules with a higher Top-10 average score after 100 iterations in most of the optimization tasks. Among benchmark methodologies, MARS is competitive, followed by Graph-GA and Stoned.

- In the **Structural Tasks**, CLaSMO gives the best results in all of the 4 tasks in this group.
- In the **Activity and Benchmark Tasks**, CLaSMO records the highest values for *median1, median2* and *qed*, while losing to MARS in *drd2* and *gsk3b*. None of the methodologies provided an improvement in *valsartan_smarts* task.

Thus, even with moderate flexibility, CLaSMO is able to harness the allowed modifications to deliver the best overall results, providing the best results in 13 optimization tasks out of 20.

Optimization with No Constraints (Similarity Threshold $\tau = 0$) When no similarity constraints are imposed (see Fig. 6), methods only need to keep the scaffold, and all additions to the scaffold are allowed.

- **Similarity and Rediscovery Tasks**, CLaSMO maintains its lead, where it achieves the highest Top-10 average scores in 4 out of 5 tasks, losing to MARS only in *thiothixene_rediscovery*.
- In the **MPO Tasks**, MARS gets the best results in 3 out of 5 objectives, and CLaSMO finds the best results in *zaleplon_mpo* task and performs similary as MARS in *osimertinin_mpo* task.
- In the **Structural Tasks**, CLaSMO gives the best results in all of the 4 tasks in this group.
- In the **Activity and Benchmark Tasks**, CLaSMO records the highest values for *median1* and *qed*, while losing to MARS in *drd2, gsk3b* and *median2*. None of the methodologies provided an improvement in *valsartan_smarts* task.

In summary, even without similarity constraints, CLaSMO effectively explores the chemical space to optimize molecular properties with high sample-efficiency. In this setting, MARS and Graph-GA methodologies provides more competitive results compared to constrained optimization settings. However, CLaSMO still achieves the best performance across 11 out of 20 optimization tasks.

Across all experimental settings, we observe notable differences in sample-efficiency among the benchmarked methods. MolDQN, a reinforcement learning based approach, and SynFlowNet, a GFlowNet-based approach consistently require substantially more oracle evaluations to achieve comparable optimization levels. These findings are consistent with those reported by (Gao et al., 2022), which highlighted the low property optimization capabilities of such methods[5]. Our results further reinforce that their slower convergence makes them less suitable for resource-constrained sample-efficient optimization tasks, where a low evaluation budget is available. In contrast, genetic algorithm-based methods, especially Graph-GA, remain competitive and often perform well across a variety of tasks. MARS, on the other hand, also shows strong performance, particularly in multi-property optimization scenarios, and achieves solid results overall. However, CLaSMO consistently outperforms all baselines across the majority of tasks and similarity constraint levels. These results highlight CLaSMO's robustness and efficiency across both constrained and unconstrained optimization settings, where sample-efficiency is essential due to limited evaluation budgets.

To illustrate how CLaSMO optimizes input scaffolds, Fig. 7 presents two representative examples from the QED optimization task. In each case, CLaSMO incrementally modifies the input scaffold by adding substructures that are conditionally generated based on the local atomic environment. Each step in the optimization process yields a molecule with an improved QED score, demonstrating the model's ability to progressively refine structure–property relationships. The examples show QED improvements from 0.615 to 0.938 and from 0.827 to 0.948, underscoring CLaSMO's ability to produce chemically valid and property-enhancing modifications through its conditional generation mechanism and LSBO-guided search.

5.2.3 Synthetic Accessibility Evaluation

To assess the synthesizability of the optimized molecules, we analyzed their synthetic accessibility (SA) scores, which estimate how difficult a compound is to synthesize based on structural complexity and fragment

[5]SynFlowNet, unlike MolDQN, was not included in the benchmark conducted by Gao et al. (2022). However, other GFlowNet-based methodologies evaluated in that study exhibited lower optimization performance compared to alternative approaches.

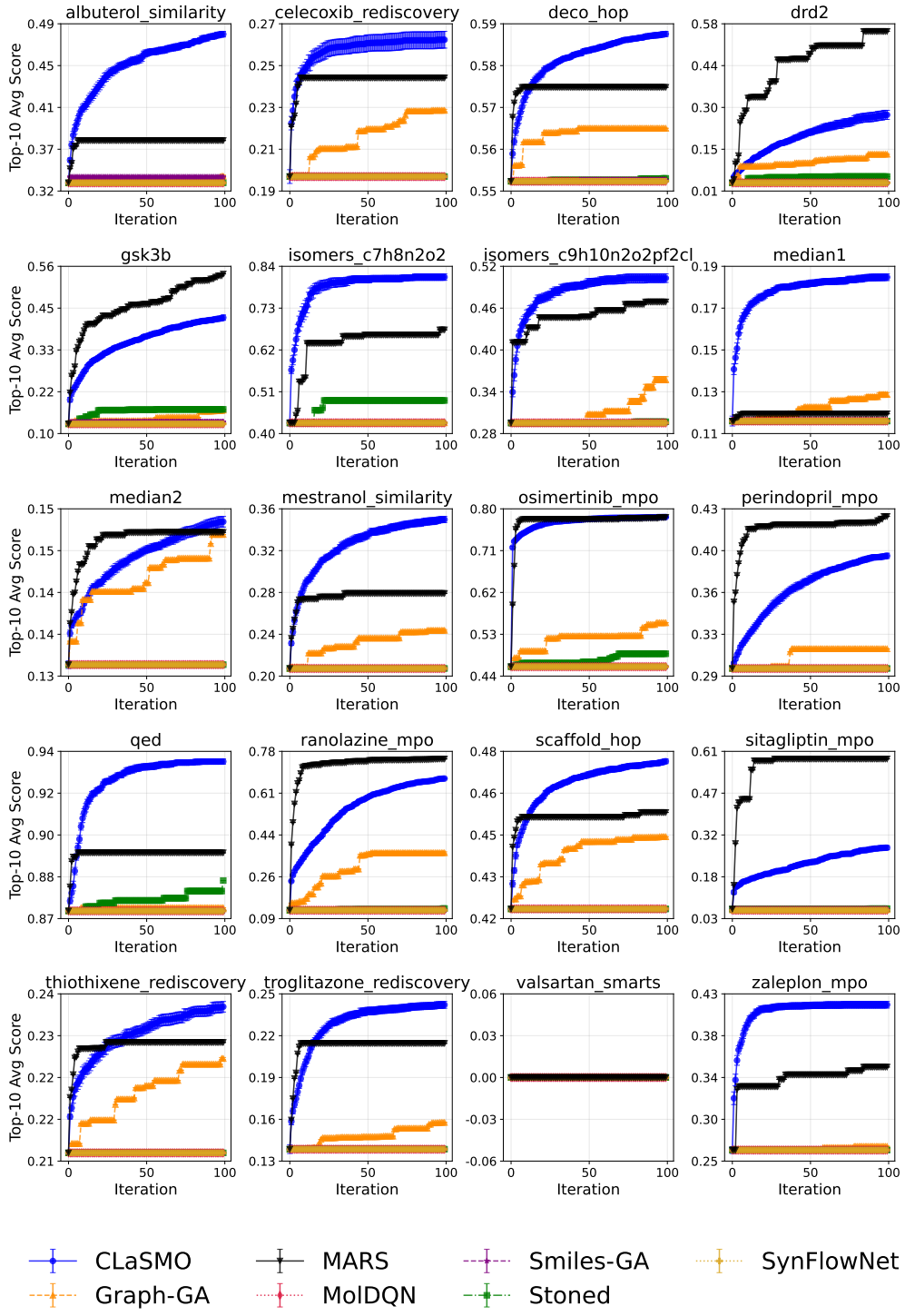

Figure 5: Comparison of Top-10 average values obtained at each iteration for CLaSMO, MARS, Smiles-GA, SynFlowNet, Graph-GA, MolDQN, and Stoned using constraint optimization setting, where Dice Similarity threshold $\tau = 0.25$. CLaSMO identifies molecules with a higher Top-10 average score after 100 iterations in most of the optimization tasks. Among benchmark methodologies, MARS is competitive, followed by Graph-GA and Stoned.

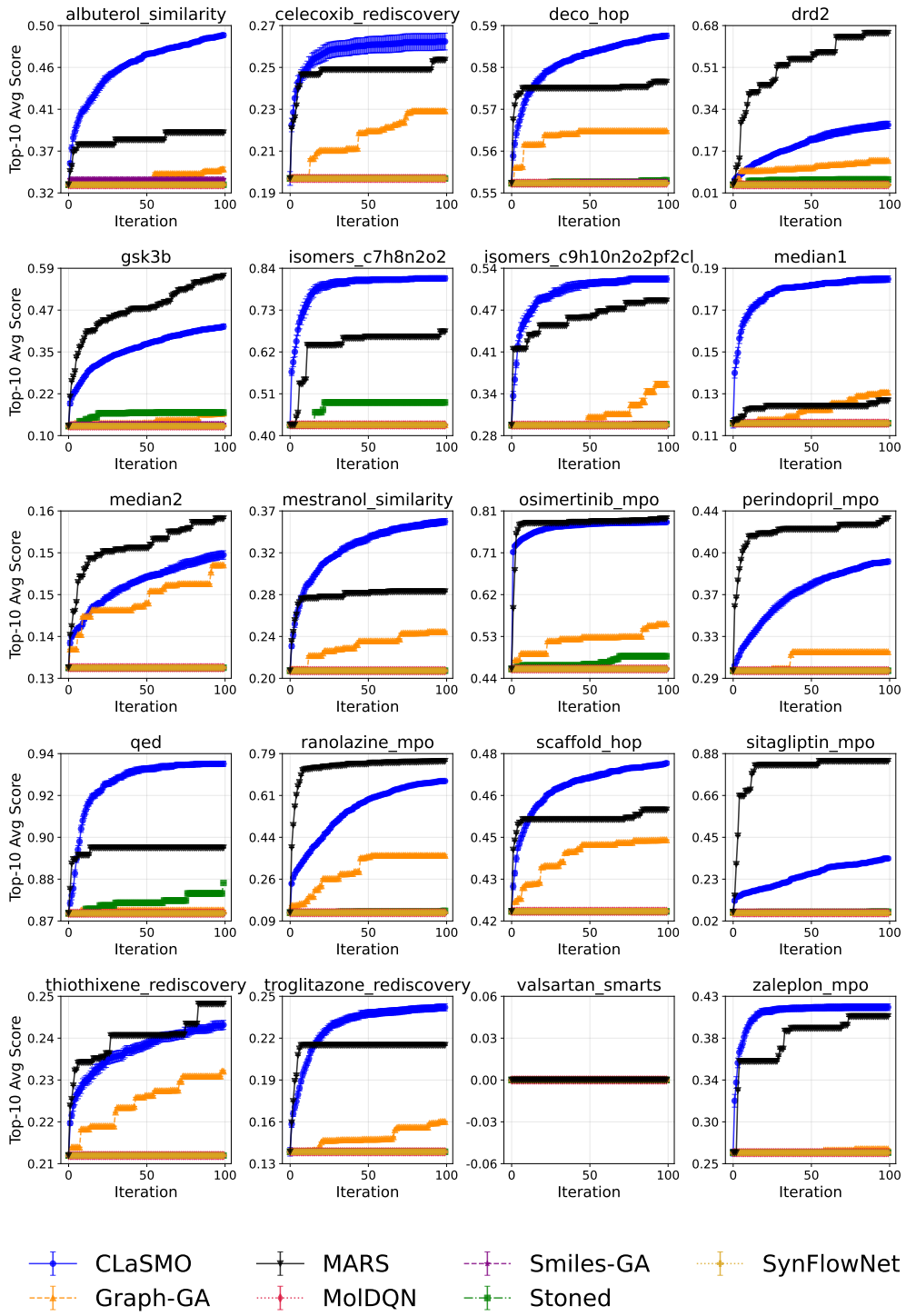

Figure 6: Comparison of Top-10 average values obtained at each iteration for CLaSMO, MARS, Smiles-GA, SynFlowNet, Graph-GA, MolDQN, and Stoned, without any constraint threshold. CLaSMO identifies molecules with a higher Top-10 average score after 100 iterations in most of the optimization tasks. Among benchmark methodologies, MARS is highly competitive, followed by Graph-GA and Stoned.

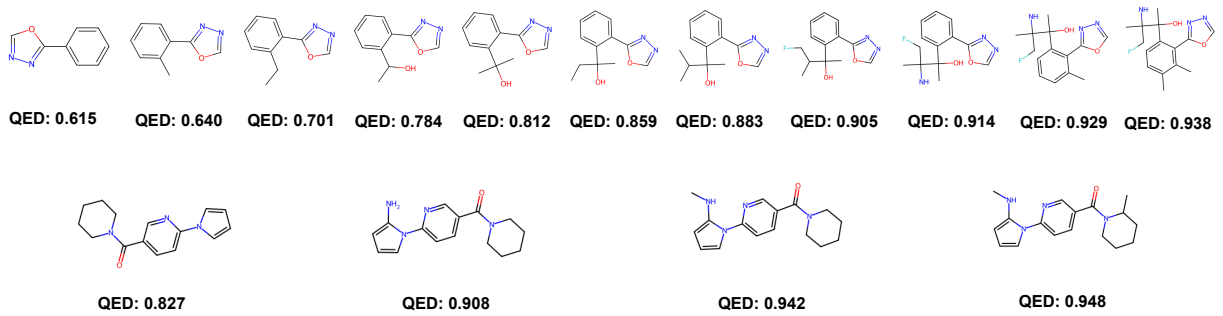

Figure 7: Examples of molecules obtained from different CLaSMO experiments in the QED setting. In both rows, the molecules on the left represent the input scaffolds. Each molecule to the right shows a step of substructure addition along with its resulting QED score. The results demonstrate QED improvements from 0.615 to 0.938 and from 0.827 to 0.948 in these examples.

contributions. Lower SA scores indicate molecules that are easier to synthesize, with values typically ranging from 1 (very easy) to 10 (very difficult). Scores above 5 are generally considered indicative of challenging synthesis Anede et al. (2024).

Given that CLaSMO and MARS exhibited the strongest performance in property optimization, we focused our SA analysis on these two methods[6]. We compared the average SA score obtained from each oracle optimization task under the lowest and highest similarity constraints, $\tau \in \{0, 0.50\}$. Figure 8 presents the distributions of these scores. The horizontal dashed line represents the average SA score of the input scaffolds, which is 3.00.

Both methods generate molecules that are, on average, more complex than the input scaffolds, as the optimized molecules exhibit higher molecular weight and structural complexity. While such increases can pose challenges for synthesis, our results demonstrate that with targeted, small-scale modifications, good synthesizability can still be maintained. At $\tau = 0$, the average SA score is 3.88 for CLaSMO and 3.74 for MARS. At $\tau = 0.50$, these scores improve slightly to 3.73 and 3.68, respectively. Although MARS consistently produces molecules with slightly better synthetic accessibility, CLaSMO maintains competitive synthesizability while achieving stronger performance in property optimization. As indicated by the distribution of SA scores, the generated molecules consistently score below 5, suggesting that their synthesis is likely to be straightforward (Anede et al., 2024). These results demonstrate CLaSMO's effectiveness in balancing synthetic feasibility with meaningful property gains, even under varying similarity constraints.

5.3 Ablation Study

We conducted an ablation study to evaluate the effect of conditioning on atomic environment features within our generative model. In this setup, the CVAE in the CLaSMO framework is replaced by a standard VAE, which generates substructures without conditioning on atomic properties. This allows us to isolate the contribution of the conditional generation mechanism.

We repeated the QED optimization experiments—one of the oracle functions used in our evaluation—using this ablated model. Performance was measured by the improvement in QED values of the input scaffolds, reported as mean improvement (Mean Imp.) and maximum improvement obtained among all input scaffolds (Max Imp.). As shown in Table 1, incorporating atomic environment features in the CVAE leads to significantly better substructure generation. The conditioning mechanism enables the model to propose fragments

[6]It should be noted that there is a strong correlation between the SA score and the similarity to the original molecule. When the similarity to the original molecule is high, the SA score also tends to be favorable, as the original molecule is synthesizable. Conversely, when the similarity is low, the SA score tends to be unfavorable. As a result, molecules generated by methods that involve minimal modifications to the original molecule tend to receive good SA scores. Therefore, we focus on comparing CLaSMO and MARS, both of which achieve high scores in property optimization.

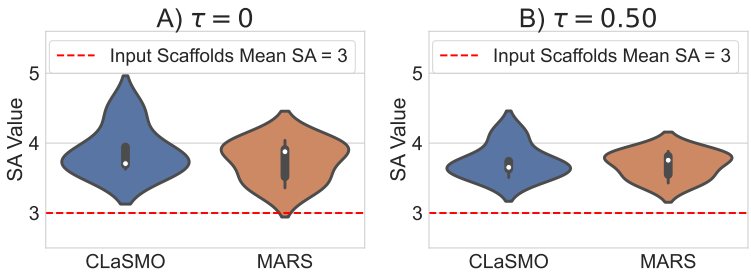

Figure 8: Distributions of average SA scores for molecules optimized by CLaSMO and MARS under similarity constraints $\tau = 0$ (**A**) and $\tau = 0.50$ (**B**). Lower scores correspond to higher synthesizability.

that bond more effectively with the scaffold, resulting in greater success in molecular property optimization. While the unconditioned VAE still benefits from the LSBO framework, conditioning substantially enhances the quality of the generated molecules.

Table 1: QED optimization results: Comparison of VAE and CVAE at different τ thresholds, reporting mean and max improvement rates.

Metric	$\boldsymbol{\tau = 0.25}$	$\boldsymbol{\tau = 0.50}$
Mean Imp. (VAE)	12.31%	8.41%
Mean Imp. (CVAE)	19.14%	16.08%
Max Imp. (VAE)	68.16%	32.28%
Max Imp. (CVAE)	114.57%	74.74%

5.4 Interactive Optimization

In CLaSMO, the modification region of the input scaffold is typically selected during the automated optimization process. However, the framework also supports an interactive mode, where a chemical expert manually selects the region of the molecule to modify. In this mode, the expert identifies the specific atom or region for modification, rather than relying on CLaSMO's automated selection. Once the region is chosen, the rest of the process remains unchanged—CLaSMO continues to optimize the molecule by generating substructures using the CVAE and refining them through LSBO to improve target properties.

This interactive approach offers several key advantages. It enables the integration of expert knowledge into the optimization process, allowing CLaSMO to operate in a Human-in-the-loop setting. Experts can leverage their domain-specific insights to target specific regions they find promising, ensuring that modifications are not only data-driven but also aligned with scientific understanding. Meanwhile, CLaSMO maintains its efficient optimization process, using LSBO to enhance molecular properties and preserve sample-efficiency. Detailed user guidelines for this open-source web application are provided in the Appendix A.7.

6 Conclusion

In this paper, we introduced CLaSMO, a novel framework that combines CVAE and LSBO for scaffold-based molecular optimization. Our approach efficiently explores latent spaces to optimize molecular properties in wide variety of objectives, while maintaining structural similarity with the input scaffold to improve the chances of real-world viability of optimized molecules. By conditioning substructure generation on the atomic environment of the target region in the input molecule, CLaSMO generates chemically meaningful modifications. The experimental results—encompassing tasks such as rediscovery of molecules and multi-property optimization settings—demonstrate that CLaSMO consistently delivers superior performance across diverse settings and optimization objectives of varying complexity. By utilizing a pioneering LSBO-based molecular editing framework that exhibits superior sample-efficiency, CLaSMO proves to be a highly effective methodology for low-budget molecular optimization scenarios. Furthermore, CLaSMO's ability to control

structural divergence via similarity constraints enables consistent performance across diverse optimization tasks, while also yielding favorable outcomes in terms of synthetic accessibility. Although this paper focuses on scaffold-based modifications, CLaSMO is fully compatible with whole molecules, requiring no changes to its methodology. Additionally, we have open-sourced a web application to allow chemical experts to use CLaSMO in a Human-in-the-Loop setting, further extending its practical applicability. Overall, CLaSMO exemplifies the power of combining scaffold-based strategies with LSBO, offering a highly effective tool for targeted drug discovery and broader molecular design challenges.

Limitations and Scope

As discussed throughout this manuscript, CLaSMO is particularly well-suited for low-budget optimization scenarios where sample-efficiency is critical. It is inherently designed to perform additive modifications on a given scaffold, which gradually increases the molecular size and weight with each editing step. By operating on scaffolds, CLaSMO helps limit the extent of molecular growth, ensuring that the optimized molecules do not become too large compared to their original counterparts, with the help of its constrained optimization capabilities. However, the current framework does not support the removal or replacement of existing substructures, limiting its flexibility in certain optimization settings. CLaSMO may not be ideal in scenarios where maintaining a fixed molecular size is crucial, or where more flexible transformations—such as fragment replacement or crossovers between molecular series—are required. In such settings, its utility may be limited without integration into broader pipelines.

Another limitation, related to the discussion above, is that CLaSMO is not suitable for de novo, from-scratch molecule generation. While it is effective at adding substructures to a given molecule, it is not designed to construct molecules from scratch. In principle, CLaSMO could be adapted to grow a molecule starting from a single atom or fragment by iteratively generating substructures, but such usage falls outside the intended design and is beyond the scope of this work.

Nonetheless, CLaSMO remains a valuable tool when scaffold-based additions are permissible. It can be effectively combined with other methods presented in this study, serving as a component within a modular molecular optimization framework tailored to diverse design constraints.

Broader Impact Statement

The work presented in this paper has the potential to accelerate the discovery of new chemical compounds, which can positively impact various industries, particularly pharmaceuticals and materials science. By improving the efficiency of molecular optimization, CLaSMO could contribute to the development of more effective drugs, especially in regions facing significant public health challenges, such as the need for rapid vaccine development. Moreover, CLaSMO's focus on real-world applicability increases the chances that the compounds discovered are not just theoretical but can be realistically produced, which is critical for translating scientific innovation into real-world solutions. Moreover, CLaSMO's ability to work in a Human-in-the-Loop setting enables domain experts to directly contribute to the optimization process, enhancing collaboration between artificial intelligence and human expertise.

References

Nada Anede, Mebarka Ouassaf, Kannan R. R. Rengasamy, Shafi Ullah Khan, and Bader Y. Alhatlani. Identification and evaluation of natural compounds as potential inhibitors of ns2b-ns3 zika virus protease: A computational approach. *Molecular Biotechnology*, 2024. ISSN 1559-0305. doi: 10.1007/s12033-024-01357-6. URL `https://doi.org/10.1007/s12033-024-01357-6`.

D. Bajusz, A. Rácz, and K. Héberger. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics*, 7:20, 2015. doi: 10.1186/s13321-015-0069-3. URL `https://doi.org/10.1186/s13321-015-0069-3`.

G. W. Bemis and M. A. Murcko. The properties of known drugs. 1. molecular frameworks. *Journal of Medicinal Chemistry*, 39(15):2887–2893, Jul 1996. doi: 10.1021/jm9602928.

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

Onur Boyar and Ichiro Takeuchi. Latent Space Bayesian Optimization With Latent Data Augmentation for Enhanced Exploration. *Neural Computation*, 36(11):2446–2478, 10 2024. ISSN 0899-7667. doi: 10.1162/neco_a_01708. URL `https://doi.org/10.1162/neco_a_01708`.

Onur Boyar, Yanheng Gu, Yuji Tanaka, Shunsuke Tonogai, Tomoya Itakura, and Ichiro Takeuchi. Crystal-lsbo: Automated design of de novo crystals with latent space bayesian optimization, 2024. URL `https://arxiv.org/abs/2405.17881`.

John Bradshaw, Brooks Paige, Matt J Kusner, Marwin Segler, and José Miguel Hernández-Lobato. A model to search for synthesizable molecules. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Nathan Brown, Marco Fiscato, Marwin HS Segler, and Alain C Vaucher. Guacamol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling*, 59(3):1096–1108, 2019.

Miruna Cretu, Charles Harris, Ilia Igashov, Arne Schneuing, Marwin Segler, Bruno Correia, Julien Roy, Emmanuel Bengio, and Pietro Liò. Synflownet: Design of diverse and novel molecules with synthesis constraints. *arXiv preprint arXiv:2405.01155*, 2024.

Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. *Advances in neural information processing systems*, 30, 2017.

Jörg Degen, Christof Wegscheid-Gerlach, Andrea Zaliani, and Matthias Rarey. On the art of compiling and using 'drug-like' chemical fragment spaces. *ChemMedChem*, 3(10):1503–1507, 2008. doi: https://doi.org/10.1002/cmdc.200800178.

Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945. ISSN 00129658, 19399170. URL `http://www.jstor.org/stable/1932409`.

Wenhao Gao, Tianfan Fu, Jimeng Sun, and Connor Coley. Sample efficiency matters: a benchmark for practical molecular optimization. *Advances in neural information processing systems*, 35:21342–21357, 2022.

Sai Krishna Gottipati, Boris Sattarov, Sufeng Niu, Yashaswi Pathak, Haoran Wei, Shengchao Liu, Shengchao Liu, Simon Blackburn, Karam Thomas, Connor Coley, Jian Tang, Sarath Chandar, and Yoshua Bengio. Learning to navigate the synthetically accessible chemical space using reinforcement learning. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3668–3679. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/gottipati20a.html`.

Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained Bayesian optimization for automatic chemical design using variational autoencoders. *Chemical science*, 11(2):577–586, 2020.

Antoine Grosnit, Rasul Tutunov, Alexandre Max Maraval, Ryan-Rhys Griffiths, Alexander Imani Cowen-Rivers, Lin Yang, Lin Zhu, Wenlong Lyu, Zhitang Chen, Jun Wang, Jan Peters, and Haitham Bou-Ammar. High-dimensional bayesian optimisation with variational autoencoders and deep metric learning. *ArXiv*, abs/2106.03609, 2021.

Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018. doi: 10.1021/acscentsci.7b00572.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2016.

John J Irwin and Brian K Shoichet. Zinc- a free database of commercially available compounds for virtual screening. *Journal of chemical information and modeling*, 45(1):177–182, 2005.

John J Irwin, Khanh G Tang, Jennifer Young, Chinzorig Dandarchuluun, Benjamin R Wong, Munkhzul Khurelbaatar, Yurii S Moroz, John Mayfield, and Roger A Sayle. Zinc20—a free ultralarge-scale chemical database for ligand discovery. *Journal of chemical information and modeling*, 60(12):6065–6073, 2020.

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

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International Conference on Machine Learning*, pp. 2323–2332, 2018.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (selfies): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, oct 2020. doi: 10.1088/2632-2153/aba947.

Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *International Conference on Machine Learning*, pp. 1945–1954, 2017.

Maxime Langevin, Hervé Minoux, Maximilien Levesque, and Marc Bianciotto. Scaffold-constrained molecular generation. *Journal of Chemical Information and Modeling*, 60(12):5637–5646, 2020.

Yibo Li, Jianxing Hu, Yanxing Wang, Jielong Zhou, Liangren Zhang, and Zhenming Liu. Deepscaffold: a comprehensive tool for scaffold-based de novo drug discovery using deep learning. *Journal of chemical information and modeling*, 60(1):77–91, 2019.

Jaechang Lim, Sang-Yeon Hwang, Seokhyun Moon, Seungsu Kim, and Woo Youn Kim. Scaffold-based molecular design with a graph generative model. *Chem. Sci.*, 11:1153–1164, 2020. doi: 10.1039/C9SC04503A. URL `http://dx.doi.org/10.1039/C9SC04503A`.

C. A. Lipinski, F. Lombardo, B. W. Dominy, and P. J. Feeney. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced Drug Delivery Reviews*, 46(1-3):3–26, March 2001. doi: 10.1016/s0169-409x(00)00129-0.

Natalie Maus, Haydn Jones, Juston Moore, Matt J Kusner, John Bradshaw, and Jacob Gardner. Local latent space bayesian optimization over structured inputs. *Advances in Neural Information Processing Systems*, 35:34505–34518, 2022.

Zhenhuan Miao, Julian Levi, and Zhen Cheng. Protein scaffold-based molecular probes for cancer molecular imaging. *Amino Acids*, 41(5):1037–1047, 2011. doi: 10.1007/s00726-010-0503-9. URL `https://doi.org/10.1007/s00726-010-0503-9`.

H. L. Morgan. The generation of a unique machine description for chemical structures - a technique developed at chemical abstracts service. *Journal of Chemical Documentation*, 5(2):107–113, 1965. ISSN 0021-9576. doi: 10.1021/c160017a018. URL `https://doi.org/10.1021/c160017a018`.

AkshatKumar Nigam, Robert Pollice, Mario Krenn, Gabriel dos Passos Gomes, and Alan Aspuru-Guzik. Beyond generative models: superfast traversal, optimization, novelty, exploration and discovery (stoned) algorithm for molecules using selfies. *Chemical science*, 12(20):7079–7090, 2021.

Raghunathan Ramakrishnan, Pavlo Dral, Matthias Rupp, and Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 08 2014. doi: 10.1038/sdata.2014.22.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010. ISSN 1549-9596. doi: 10.1021/ci100050t. URL https://doi.org/10.1021/ci100050t.

Jerret Ross, Brian Belgodere, Samuel C Hoffman, Vijil Chenthamarakshan, Youssef Mroueh, and Payel Das. Gp-molformer: A foundation model for molecular generation. *arXiv preprint arXiv:2405.04912*, 2024.

Lars Ruddigkeit, Ruud van Deursen, Lorenz C. Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of Chemical Information and Modeling*, 52(11):2864–2875, 2012. doi: 10.1021/ci300415d. PMID: 23088335.

Stuart L. Schreiber. Target-oriented and diversity-oriented organic synthesis in drug discovery. *Science*, 287 (5460):1964–1969, 2000. doi: 10.1126/science.287.5460.1964. URL https://www.science.org/doi/abs/10.1126/science.287.5460.1964.

Austin Tripp and José Miguel Hernández-Lobato. Genetic algorithms are strong baselines for molecule generation. *arXiv preprint arXiv:2310.09267*, 2023.

Austin Tripp, Erik Daxberger, and José Hernández-Lobato. Sample-efficient optimization in the latent space of deep generative models via weighted retraining. In *Advances in Neural Information Processing Systems*, 06 2020.

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. doi: 10.1021/ci00057a005.

Jesse A Weller and Remo Rohs. Structure-based drug design with a deep hierarchical generative model. *Journal of Chemical Information and Modeling*, 64(16):6450–6463, 2024.

Matthew E Welsch, Scott A Snyder, and Brent R Stockwell. Privileged scaffolds for library design and drug discovery. *Current Opinion in Chemical Biology*, 14(3):347–361, 2010. ISSN 1367-5931. doi: https://doi.org/10.1016/j.cbpa.2010.02.018. Molecular Diversity.

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*, 2021.

Chaochao Yan, Sheng Wang, Jinyu Yang, Tingyang Xu, and Junzhou Huang. Re-balancing variational autoencoder loss for molecule sequence generation. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pp. 1–7, 2020.

Zhenpeng Zhou, Steven Kearnes, Li Li, Richard N Zare, and Patrick Riley. Optimization of molecules via deep reinforcement learning. *Scientific reports*, 9(1):10752, 2019a.

Zhenpeng Zhou, Steven Kearnes, Li Li, Richard N Zare, and Patrick Riley. Optimization of molecules via deep reinforcement learning. *Scientific reports*, 9(1):1–10, 2019b.

A Appendix

A.1 Definitions of Atomic Features

In Table 2, we provide the definitions of the six atomic features we utilized in our data preparation setting.

Property	Description
Atom Type	Specifies the element (e.g., carbon, oxygen), which determines bonding capabilities and chemical behavior.
Hybridization	Describes the mixing of atomic orbitals, influencing shape, bond angles, and bonding interactions.
Valence	Refers to the number of bonds an atom can form, indicating potential for additional bonding.
Formal Charge	Represents the charge if all bonding electrons are shared equally, crucial for reactivity and bonding sites.
Degree	Denotes the number of directly attached atoms (neighbors), providing insight into the local atomic environment.
Ring Membership	Indicates if the atom is part of a ring structure, impacting substructure rigidity, stability, and bonding behavior.

Table 2: Key atomic properties used in CVAE training in CLaSMO framework.

A.2 Condition Vector Embeddings

The simplicity of our dataset allowed us to reduce the latent space to two dimensions without compromising reconstruction quality, thereby enhancing LSBO performance. Consequently, the decision to use 2-dimensional embeddings for the condition vectors, rather than the original six-dimensional features, was motivated by this low-dimensional latent space configuration in our CVAE model.

Using a six-dimensional condition vector in a CVAE with a 2-dimensional latent space poses a significant challenge: the decoder may overly rely on the condition vectors during reconstruction, potentially diminishing the latent space's ability to encode critical structural information. This can result in a loss of critical structural information within the latent space, as a disproportionate amount of information is drawn from the condition vectors. Such an imbalance can degrade the quality of the latent space, negatively affecting the surrogate model used in LSBO and ultimately hindering the optimization process. Conversely, when 2-dimensional condition vectors are used, the latent vectors must encode more information about the input instances, as the decoder's reliance on the condition vectors is reduced. This shift encourages the latent space to better capture the underlying structural properties, improving its utility for LSBO. In LSBO, latent vectors are utilized to train the surrogate model, which guides the search process by predicting target property values within specific regions of the latent space. Therefore, ensuring that the latent vectors are information-rich is crucial for achieving efficient LSBO performance.

To evaluate this effect, we compared CVAE models trained with 2-dimensional embeddings of the conditional features to those trained with the original 6-dimensional features. For both models, we obtained the 2-dimensional latent vectors of input scaffolds using the encoders of the trained CVAEs and calculated the corresponding QED values. We then trained random forest regressors to predict the QED values of the input scaffolds based on the latent vectors from each model. Our results showed that the CVAE with 2-dimensional condition embeddings achieved a 5% lower mean squared error in QED predictions, underscoring the benefits of reduced-dimensional embeddings in preserving latent space quality and improving LSBO performance.

We acknowledge that the model with 6-dimensional condition vectors converged faster and achieved slightly better reconstruction accuracy during its training compared to model with 2-dimensional condition vector embeddings. However, our primary goal was to design the latent space for LSBO, where the latent representation provided by the 2-dimensional embeddings proved more effective.

A.3 Model and Hyperparameter Selection

In chemical VAE models, it has been established that setting the KL divergence weight $\beta < 1$ can improve generative performance Yan et al. (2020), and our findings are consistent with this. We experimented with a range of β values from 1 to 1^{-7}, selecting models based on their reconstruction accuracy on the training set.

Interestingly, we found that even at very low β values, the CVAE retained its generative capacity. However, as β increased, we observed a decline in both reconstruction accuracy and the diversity of generated molecules.

To ensure robust training, we applied early stopping in combination with a learning rate scheduler (using PyTorch's ReduceLROnPlateau function). Models were evaluated based on their reconstruction performance and generative diversity, leading us to select the model with $\beta = 0.000001$ as the optimal candidate. Additionally, our CVAE leverages conditional batch normalization De Vries et al. (2017), which improves the impact of conditioning in the generative process.

For CLaSMO's penalization terms, we opted for a straightforward approach rather than an exhaustive hyperparameter optimization process. We set $\lambda_1 = -5$ to penalize cases where the similarity constraint $\text{DICE}(\boldsymbol{S}, \boldsymbol{S'}) > \tau$ was violated. Similarly, we assigned $\lambda_2 = -7.5$ for situations where the generated substructure could not be added to the input scaffold. These values were chosen to assign poor scores in cases where the sampled region did not meet the desired criteria, allowing LSBO to learn that the region is suboptimal. This approach helps guide the optimization process away from unproductive regions and toward more promising areas.

A.4 Sample-Efficiency Benchmark Experimental Setting

In the sample-efficiency benchmark experiments adopted from Gao et al. (2022), we reduced the number of optimization iterations from 10,000 to 100 to create a low-budget experimental setting. For each methodology we limited the number of oracle calls to 100.

In the original open-source implementation, Smiles-GA uses a population size of 50, while Graph-GA uses a population size of 120. Given our limited optimization budget of only 100 iterations, these values required adjustment. Furthermore, Smiles-GA allows for 500 mutations in its default configuration, and Graph-GA employs a mutation rate of 6.7%. As we report results on the cases where the scaffold is preserved, we looked for the balance between scaffold preservation and exploration. For this reason, we conducted a grid search for selecting the best-performing sets of population size and mutation rate for the Graph-GA methodology. Population size is selected among $4, 5, 10$ and the candidate set of mutation rate is $6.7\%, 4.5\%, 1\%, 0.1\%, 0.05\%, 0.01\%, 0.005\%$. The best results were obtained when the mutation rate is equal to 4.5% and the population size is equal to 5. For SMILES-GA, we experimented with population sizes of $4, 5, 10$ by setting the number of mutations to 10, and best results were obtained when the population size was set to 5.

In the case of SynFlowNet, the open-source implementation provided by the authors was not directly applicable to scaffold optimization tasks. However, the authors kindly guided us on how to adapt their approach for this setting, and we updated their codebase accordingly based on their recommendations.

The Stoned methodology, although categorized as a genetic algorithm in Gao et al. (2022), relies on random rearrangements of atoms and lacks a specific hyperparameter to control the degree of divergence from the initial scaffold. Consequently, similar to MARS, and MolDQN, we used the default hyperparameters for the experiments using Stoned methodology to optimize the given scaffolds.

A.5 Oracle Definitions

In this section, we provide brief descriptions of the oracle functions used in our experiments. For additional details and references, we refer the reader to the Therapeutics Data Commons website: `https://tdcommons.ai/functions/oracles/`.

Group 1: Similarity and Rediscovery Tasks

- **albuterol_similarity:** Measures similarity to albuterol, a bronchodilator.
- **celecoxib_rediscovery:** Regenerates celecoxib, an anti-inflammatory drug.
- **mestranol_similarity:** Quantifies similarity to mestranol, a synthetic estrogen.

- **thiothixene_rediscovery:** Seeks to recover thiothixene, an antipsychotic.
- **troglitazone_rediscovery:** Rediscovery task for troglitazone, a former diabetes drug.

Group 2: Multi-Property Optimization (MPO) Tasks

- **osimertinib_mpo:** Multi-property optimization for osimertinib, an EGFR inhibitor.
- **perindopril_mpo:** Focused on perindopril, an ACE inhibitor.
- **ranolazine_mpo:** Related to ranolazine, an anti-anginal compound.
- **sitagliptin_mpo:** MPO task for sitagliptin, an antidiabetic drug.
- **zaleplon_mpo:** Targets properties associated with zaleplon, a sedative-hypnotic.

Group 3: Structural Modification Tasks

- **deco_hop:** Scaffold hopping task with partial structural retention.
- **scaffold_hop:** Generates novel scaffolds while maintaining core activity.
- **isomers_c7h8n2o2:** Generates valid isomers with the formula $C_7H_8N_2O_2$.
- **isomers_c9h10n2o2pf2cl:** Generates valid isomers of $C_9H_{10}N_2O_2PF_2Cl$.

Group 4: Activity and Benchmark Tasks

- **drd2:** Predicts activity against dopamine receptor D2.
- **gsk3b:** Predicts activity against GSK3β, a kinase target.
- **median1, median2:** Benchmark tasks requiring similarity across multiple target molecules.
- **qed:** Quantitative Estimate of Drug-likeness.
- **valsartan_smarts:** Checks for specific SMARTS patterns associated with valsartan.

A.6 More on Synthetic Accessibility Scores

Conducting a fair comparison of synthetic accessibility (SA) scores requires focusing on settings where meaningful optimization has occurred. To this end, we included only those experimental configurations in which at least 30 out of 100 input scaffolds were successfully optimized—that is, they resulted in an improvement in the target oracle value. This criterion ensures that the SA score analysis reflects the overall synthesizability of molecules that were effectively optimized.

A.7 Human-in-the-Loop via Web-Application

In Fig. 9, we showcase a sequence of screenshots from our web application, demonstrating the process of molecule optimization. First, the user inputs a SMILES Weininger (1988) string of the chemical compound into the designated text field. Once the input is provided, the application automatically computes the molecule's QED value and generates a visual representation. Subsequently, the user selects a region of interest by drawing a rectangle around the area they wish to modify. Upon confirming the selection, the CLaSMO optimization process is initiated, targeting improvements in the selected molecular region. Upon completion, the optimized molecule is displayed, and the process can be continued by using the resulting molecule as input for further iterations. By incorporating user input in the region selection, we create a Human-in-the-Loop optimization workflow.

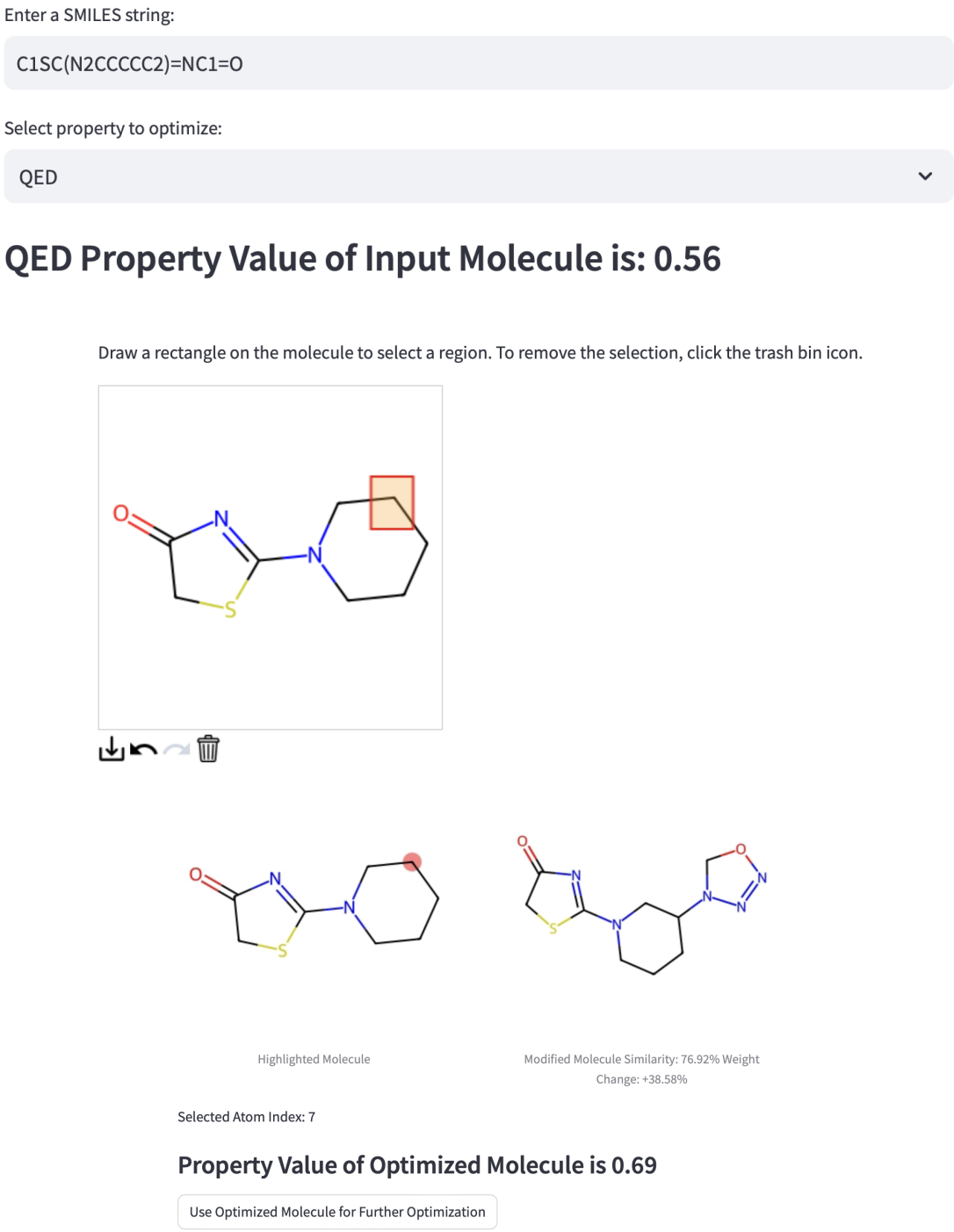

Figure 9: Screenshots from our web application showing the step-by-step process of molecule optimization using CLaSMO in an interactive setting. The process includes inputting a SMILES string, visualizing the molecule's QED value, selecting a region for modification, and initiating the optimization procedure. In this example, the QED value of the input molecule is improved from 0.56 to 0.69, where the resulting molecule is demonstrated in the bottom-right figure.