Journal Title Here, 2022, 1–7
doi: DOI HERE
Advance Access Publication Date: Day Month Year
Paper

AutoLead: An LLM-Guided Bayesian Optimization Framework for Multi-Objective Lead Optimization

Yiming Zhang,[1,2] Jun Jin Choong,[2] Kaushalya Madhawa[2] and Keisuke Ozawa[2]*

[1]Department of Computational Biology and Medical Sciences, Graduate School of Frontier Sciences, The University of Tokyo, Chiba, Japan and [2]SB Intuitions, Japan

*Corresponding author. keisuke.ozawa@sbintuitions.co.jp

Abstract

The process of lead optimization in drug discovery is a complex, multi-objective challenge that remains a major bottleneck in the development of new therapeutics. Traditional approaches often struggle to efficiently explore the vast chemical space while simultaneously optimizing multiple, and sometimes conflicting, molecular properties. In this work, we present AutoLead, a novel framework that integrates Large Language Models (LLMs) with multi-objective Bayesian optimization to tackle this challenge. By leveraging the chemical reasoning capabilities of LLMs, AutoLead effectively guides the search for novel drug-like molecules that satisfy multiple objectives. We evaluate our approach on two molecule optimization tasks, achieving state-of-the-art results. Furthermore, we introduce a new benchmark dataset designed around a more realistic lead optimization scenario, where the task is to modify compounds that violate Lipinski's Rule of Five to simultaneously meet all criteria and improve their QED score. Through extensive experiments and a detailed case study, we demonstrate the potential of combining LLMs with black-box optimization techniques for more efficient and practical drug discovery.

Key words: Drug discovery, Lead optimization, Bioinformatics, Large Language Models, Bayesian optimization

1. Introduction

Lead optimization is a critical stage in drug discovery, where chemists iteratively modify hit compounds to improve their pharmacokinetic properties while ensuring efficacy and safety Caldwell et al. [2009], Rich et al. [2024]. Unlike early-stage virtual screening, lead optimization involves complex, multi-objective trade-offs, such as simultaneously improving solubility, permeability, and drug-likeness, under strict medicinal chemistry constraints Winter et al. [2019], Eddershaw et al. [2000]. Traditional approaches often rely on expert heuristics or rule-based systems to make localized edits Bohacek et al. [1996], which typically operate in oversimplified settings and fail to scale in the face of combinatorial chemical space and tightly coupled objectives Winter et al. [2019], Yu et al. [2025].

Recent advancements in large language models (LLMs) have opened promising avenues for pipelining molecular optimization Li et al. [2025], Yu et al. [2025], Ghareeb et al. [2025], Gao et al. [2025]. This has led to two main strategies. One involves developing domain-specific LLMs that are pre-trained on vast scientific corpora to enhance their chemical reasoning Zhang et al. [2024, 2025], Dey et al. [2025]. Specifically for molecular optimization, DrugAssist Ye et al. [2025] achieved leading results by instruction-tuning an LLM on a large dataset for multi-property optimization. These works highlight the potential of LLM-guided optimization but often lean on specialized models or extensive fine-tuning.

While these approaches highlight the power of LLMs in capturing chemical reasoning and facilitating property-guided edits, they also exhibit limitations, particularly their reliance on large-scale instruction-finetuning datasets tailored for molecule editing. The other adapts powerful general-purpose models for chemical tasks. ChatDrug Liu et al. [2024] introduced a conversational framework to iteratively edit molecules through prompt design and dialogue. However, ChatDrug relies on an external molecular database for retrieval, a dependency that inherently restricts its exploration to known chemical space and limits its ability to generate potential novel candidates Jensen [2019]. Moreover, the tasks and datasets used there primarily focus on single-objective optimization or loosely coupled dual-property scenarios, falling short of the complex, tightly constrained multi-objective requirements encountered in realistic lead optimization workflows.

To overcome these limitations, we introduce a new framework—**AutoLead**—that combines the strengths of LLMs with the principled decision-making capabilities of Bayesian Optimization (BO), without requiring any external retrieval database or domain-specific training data. In our approach,

© The Author 2022. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

the LLM first proposes a diverse set of candidate molecules by leveraging its inherent knowledge of chemical structure and function. These initial proposals serve as seeds for BO, which then guides the optimization process by selectively exploring or exploiting candidates based on historical performance and uncertainty estimates modeled by Gaussian Processes (GPs) Williams and Rasmussen [2006], Gramacy [2020]. This integration allows AutoLead to dynamically alternate between LLM-driven exploration and GP-guided exploitation, enabling efficient navigation of the chemical space under multi-objective constraints.

In practical drug discovery, lead optimization often begins with modifying hit compounds that violate key drug-like properties, such as Lipinski's Rule of Five Lipinski et al. [2012], Lipinski [2004], to simultaneously satisfy multiple pharmacokinetic constraints and improve drug-likeness metrics like QED. To evaluate our method in this realistic setting, we compile a new benchmark dataset derived from the HiQBind Wang et al. [2025] dataset, explicitly designed for lead optimization where the objective is to transform molecules that fail one or more Lipinski criteria into candidates that satisfy all rules while enhancing their QED scores.

Our key contributions are summarized as follows:

- We propose AutoLead that integrates general-purpose LLMs with multi-objective BO specifically designed to generate and refine molecules under realistic pharmacokinetic constraints, without target-specific tuning or retrieval databases.
- We demonstrate that AutoLead achieves state-of-the-art performance on three benchmarks, ChatDrug-200, DrugAssist-500, and our newly introduced LipinskiFix-1000, highlighting its broad applicability and effectiveness in complex multi-objective settings.
- We construct LipinskiFix-1000, a challenging benchmark dataset derived from HiQBind Wang et al. [2025], designed to evaluate lead optimization methods on tasks that require repairing non-compliant molecules while improving overall drug-likeness.

Together, these contributions showcase how combining LLMs' generative flexibility with BO's principled exploration can yield a powerful and data-efficient solution to real-world lead optimization.

2. Materials and methods

2.1. Overview

We tackle molecular lead optimization, where given an initial molecule, the goal is to generate a sequence of improved molecules by optimizing multiple drug-relevant properties such as drug-likeness (QED), lipophilicity (logP), and Lipinski compliance, addressing the challenge of multi-objective optimization that existing methods often struggle with.

AutoLead addresses this problem via a hybrid *LLM-in-the-loop Bayesian optimization* framework (Figure 1) that couples the generative chemical reasoning capabilities of general-purpose LLMs with uncertainty-aware optimization using a GP surrogate model. We also provide a mapping strategy between SMILES strings and feature vectors to mitigate the difficulty of incorporating BO into our hybrid scheme.

2.2. Problem Formulation

Let $\mathcal{S}$ denote the set of valid SMILES Weininger [1988] strings and let

$$f : \mathcal{S} \longrightarrow \mathbb{R}^d$$

be a property evaluator that returns a vector of target properties such as QED, logP, tPSA, number of hydrogen bond donors/acceptors and molecular weight. We aim to solve the optimization problem:

$$s^* = \arg\max_{s \in \mathcal{S}} U(f(s)), \tag{1}$$

where $U(\cdot)$ aggregates the d objectives, and is by default taken as a uniform average: $U(f(s)) = \frac{1}{d}\sum_{i=1}^{d} f_i(s)$.

2.3. Surrogate Objective and LLM-driven Inverse Mapping

Since BO cannot directly optimize over discrete SMILES strings, we introduce a continuous descriptor space:

$$\phi : \mathcal{S} \longrightarrow \mathbb{R}^p,$$

where $\phi(s)$ maps a molecule s to a p-dimensional vector of normalized RDKit RDKit descriptors. To ensure all values fall within the range $[0, 1]$ for GP-based Bayesian Optimization, each descriptor is divided by a fixed normalization constant (shown in parentheses). Specifically, we use 10 molecular properties: molecular weight (500 Da), logP (5.0), polar surface area (200 Å^2), number of hydrogen bond donors (10), number of hydrogen bond acceptors (10), number of rotatable bonds (20), ring count (10), number of aromatic rings (5), number of saturated rings (5), and sp^3 carbon fraction (which is inherently bounded within $[0, 1]$. This normalization mitigates heterogeneous units and improves GP conditioning and acquisition optimization, in line with prior recommendations that objective scales and trust-region potentials are dynamic during optimization Chu et al. [2024]).

This 10-dimensional vector provides a pharmacologically meaningful and normalized space for latent space optimization. It is important to note that f is a property evaluator (e.g., computing logP), whereas ϕ is used to represent molecules for optimization.

A technical challenge, however, is converting an optimized point from this latent space back into a valid chemical structure (a SMILES string). This inverse mapping, ϕ^{-1}, is generally intractable. To overcome this, we propose a training-free LLM-driven decoder, denoted as ϕ^{-1}. Instead of attempting a direct conversion from latent space, our approach leverages the generative power of an LLM Zeng et al. [2024]. We translate the optimized latent vector $x \in [0, 1]^p (p = 10)$ into a human-readable text prompt. For instance, the vector is verbalized into a sentence such as, "Generate a molecule with a molecular weight of approximately 310 Da, a logP of 3.5, ...," instructing the LLM to produce a molecule that satisfies these 10 target properties. The optimization process is therefore approximately realized by:

$$\max_{x \in [0,1]^p} g(x), \text{where } g(x) := U(f(\tilde{\phi}^{-1}(x)))$$

In this framework, BO efficiently searches for the optimal vector in the continuous latent space, and our LLM-driven decoder acts as a bridge, translating these optimized vectors back into tangible molecules. Finally, the generated molecules are evaluated by the property evaluator f to verify if they meet the desired optimization goals.

2.4. Bayesian Surrogate Modeling

As in Chang et al. [2025], given observations up to step $(t-1)$, we define the data points as:

$$\mathcal{D}_{t-1} = \{(x_i, u_i)\}_{i=1}^{t-1}, \qquad \text{where } u_i = g(x_i).$$

We place a GP prior over the utility function $U(x)$ and obtain the following posterior distribution:

$$U(x) \mid \mathcal{D}_{t-1} \sim \mathcal{GP}\big(\mu_{t-1}(x), \sigma_{t-1}^2(x)\big),$$

where the predictive mean and variance are given by:

$$\mu_{t-1}(x) = k_{t-1}(x)^\top (K + \lambda^2 I)^{-1} y, \tag{2}$$

$$\sigma_{t-1}^2(x) = k(x,x) - k_{t-1}(x)^\top (K + \lambda^2 I)^{-1} k_{t-1}(x). \tag{3}$$

Here, $K \in \mathbb{R}^{(t-1)\times(t-1)}$ is the Gram matrix of the training inputs, $\mathcal{D}_{t-1}$, where each entry is defined as $K_{ij} = k(x_i, x_j)$ for $i, j \in \{1, \ldots, t-1\}$. The vector $k_{t-1}(x) = [k(x, x_1), \ldots, k(x, x_{t-1})]^\top$ contains the covariances between the test point x and the training inputs. The vector $y = [u_1, \ldots, u_{t-1}]^\top$ stores the observed utility values. We use a radial basis function kernel $k(\cdot,\cdot)$ as the covariance function.

2.5. LLM-in-the-Loop Optimization Strategy

We employ a transient hybrid strategy Chang et al. [2025] to gradually transition from LLM-based exploration to BO-based exploitation. The probability of using BO at round t is:

$$p_t = \min\left(\frac{t^\alpha}{T}, 1\right), \quad \alpha = 2,$$

where T is the total iteration budget. Let $z_t \sim \text{Bernoulli}(p_t)$. If $z_t = 1$ and $|\mathcal{D}_{t-1}| \geq \tau$, a threshold size of data points for switching the strategy, we invoke BO using our LLM decoding; otherwise, the LLM generates the molecule:

$$s_t = \begin{cases} \tilde{\phi}^{-1}(x_t), & \text{if } z_t = 1 \text{ and } |\mathcal{D}_{t-1}| \geq \tau, \\ G_{\text{LLM}}(x_t), & \text{otherwise.} \end{cases} \tag{4}$$

Here, $\tilde{\phi}^{-1}(x_t)$ denotes molecule generation via the BO proposal obtained by maximizing the acquisition function in descriptor space, while G_{LLM} denotes a direct molecule generation by the LLM. We set $\tau = 3$ so that BO starts only after at least three data points have been collected.

Acquisition Function.
The BO step maximizes the upper confidence bound (UCB) acquisition function:

$$\alpha_{\text{UCB}}(x) = \mu_{t-1}(x) + \beta_t \sigma_{t-1}(x),$$

where

$$\beta_t = 2\log\left(\frac{(t+1)\cdot p \cdot \pi^2}{0.6}\right),$$

and p denotes the feature dimension, as also implemented in Chang et al. [2025].

2.6. Multi-Objective Handling

1. **Weighted Utility:** $U(f(s)) = \sum_{i=1}^{d} w_i f_i(s)$, with default $w_i = 1/d$.
2. **Pareto Tracking:** maintain a Pareto front of non-dominated solutions Yong et al. [2025].

2.7. Overall Optimization Loop

Algorithm 1 outlines the overall optimization procedure of AutoLead. Starting from an initial molecule s_0, the algorithm iteratively alternates between LLM-based exploration and BO-guided exploitation, controlled by a Bernoulli variable. At each step, a new candidate is generated, evaluated, and added to the historical optimization trajectory. The surrogate model and Pareto front are continuously updated, and the final output is a set of non-dominated molecules after T iterations.

Algorithm 1 AutoLead

Require: Initial SMILES string s_0, budget T
Ensure: A set of non-dominated molecules (Pareto front)

```
D ⇐ {(φ(s_0), g(φ(s_0)))}                    ▷ Initialization
t ⇐ 1
while t ≤ T do
    Fit a GP model on the current sequence of data points D
    Sample a random variable z_t ~ Bernoulli(p_t)
    if z_t = 1 and |D| ≥ τ then                ▷ Exploitation phase
        x_t ⇐ arg max_x α_UCB(x)               ▷ Select candidate via BO
        s_t ⇐ φ̃^{-1}(x_t)                      ▷ Decode to SMILES
        if s_t is not valid or decoding fails then
            s_t ⇐ G_LLM                        ▷ Fallback to LLM generation
        end if
    else                                       ▷ Exploration phase
        s_t ⇐ G_LLM                            ▷ Generate candidate via LLM
    end if
    u_t ⇐ g(φ(s_t))                            ▷ Evaluate the new molecule
    D ⇐ D ∪ {(φ(s_t), u_t)}                    ▷ Update sequence
    Update the Pareto front with (φ(s_t), u_t)
    t ⇐ t + 1
end while
return the set of non-dominated molecules from D
```

2.8. Benchmark datasets

We evaluate AutoLead on multiple molecular optimization datasets covering diverse optimization scenarios, from single-objective property optimization to complex multi-objective lead optimization tasks.

2.8.1. LipinskiFix-1000

We further introduce LipinskiFix-1000, a benchmark dataset designed to evaluate molecular optimization strategies on realistic drug-like compounds. It consists of 1,000 unique ligands extracted from the high-quality HiQBind Wang et al. [2025] database, each bound to experimentally validated protein targets. These molecules were selected for violating at least one of Lipinski's Rule of Five criteria, reflecting common challenges in early drug discovery. To ensure meaningful optimization, we applied additional quality constraints on molecular weight, lipophilicity, and QED scores, while limiting structural complexity. Each task targets restoring Lipinski compliance as the primary objective and maximizing QED as a secondary goal, providing a rigorous testbed for multi-objective molecular design.

Table 1. Success rate (%) on 16 ChatDrug-200 single-objective and 12 multi-objective small-molecule editing tasks. Δ denotes the required improvement threshold for each property. Evaluation is the hit ratio of the property change. The best results are marked in **bold**.

Property	Δ	Random	PCA	High Variance	GS-Mutate	MoleculeSTM-SMILES	ChatDrug		AutoLead		
							GPT-3.5-turbo	GPT-4o	LLMO	BO	HO
logP-	0	35.33 ± 1.31	33.80 ± 3.63	33.52 ± 3.75	52.00 ± 0.41	61.87 ± 2.67	22.8 ± 0.62	71.7 ± 1.70	88.3 ± 1.26	**89.8 ± 0.76**	87.7 ± 1.53
	0.5	11.04 ± 2.40	10.66 ± 3.24	10.86 ± 2.56	14.67 ± 0.62	49.02 ± 1.84	11.7 ± 0.94	62.2 ± 1.18	**78.5 ± 0.50**	**78.5 ± 0.87**	78.2 ± 1.15
logP+	0	43.36 ± 3.06	39.36 ± 2.55	42.89 ± 2.36	47.50 ± 0.41	52.71 ± 1.67	54.2 ± 0.24	78.8 ± 0.85	**91.5 ± 1.32**	90.7 ± 0.76	90.5 ± 1.00
	0.5	19.75 ± 1.56	15.12 ± 2.93	18.22 ± 3.20	12.50 ± 0.82	30.47 ± 3.26	7.5 ± 0.41	35.2 ± 0.85	**84.7 ± 1.26**	84.5 ± 0.00	83.0 ± 2.29
QED+	0	38.06 ± 2.57	33.99 ± 3.72	36.20 ± 4.34	28.00 ± 0.71	36.52 ± 2.46	18.2 ± 0.62	36.8 ± 1.31	42.0 ± 0.87	49.7 ± 2.31	**59.3 ± 2.08**
	0.1	5.27 ± 0.24	3.97 ± 0.10	4.44 ± 0.58	6.33 ± 0.82	8.81 ± 0.82	4.0 ± 0.41	8.3 ± 0.24	3.7 ± 0.58	8.5 ± 1.32	**11.5 ± 2.50**
QED-	0	36.96 ± 2.25	35.17 ± 2.61	39.99 ± 0.57	71.33 ± 0.85	58.59 ± 1.01	43.8 ± 2.72	58.0 ± 1.08	85.2 ± 2.52	85.5 ± 1.80	**86.5 ± 0.87**
	0.1	6.16 ± 1.87	5.26 ± 0.95	7.56 ± 0.29	27.67 ± 3.79	37.56 ± 1.76	8.7 ± 0.24	10.3 ± 0.85	60.5 ± 1.50	66.0 ± 1.32	**67.3 ± 1.26**
tPSA-	0	25.23 ± 2.13	21.36 ± 0.79	21.98 ± 3.77	22.00 ± 0.82	**57.74 ± 0.60**	6.8 ± 1.03	40.8 ± 2.01	39.0 ± 2.63	41.8 ± 1.26	40.5 ± 1.80
	10	17.41 ± 1.43	14.52 ± 0.80	14.66 ± 2.13	6.17 ± 0.62	**47.51 ± 1.88**	5.0 ± 0.41	28.3 ± 0.24	30.0 ± 2.60	30.0 ± 0.50	27.5 ± 0.50
tPSA+	0	16.79 ± 2.54	15.48 ± 2.40	17.10 ± 1.14	28.83 ± 1.25	34.13 ± 0.59	12.5 ± 0.41	83.8 ± 0.47	91.0 ± 0.54	89.3 ± 0.76	**91.5 ± 0.50**
	10	11.02 ± 0.71	10.62 ± 1.86	12.01 ± 1.01	15.17 ± 1.03	26.48 ± 0.97	11.8 ± 1.31	82.0 ± 1.63	90.5 ± 1.32	**90.8 ± 0.58**	90.3 ± 1.04
HBA+	0	12.64 ± 1.64	10.85 ± 2.29	11.78 ± 0.15	21.17 ± 3.09	54.01 ± 5.26	18.7 ± 1.70	74.8 ± 1.25	89.0 ± 0.38	79.0 ± 1.32	**89.3 ± 0.29**
	1	0.69 ± 0.01	0.90 ± 0.84	0.67 ± 0.01	1.83 ± 0.47	27.33 ± 2.62	1.3 ± 0.24	6.5 ± 0.50	79.2 ± 0.29	78.2 ± 1.15	**80.7 ± 1.04**
HBD+	0	2.97 ± 0.61	3.97 ± 0.55	6.23 ± 0.66	19.50 ± 2.86	28.55 ± 0.76	27.2 ± 2.01	77.8 ± 1.55	74.7 ± 12.77	83.5 ± 1.00	**86.5 ± 0.00**
	1	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	1.33 ± 0.24	7.69 ± 0.56	1.0 ± 0.00	7.2 ± 0.25	**72.5 ± 1.32**	71.2 ± 0.58	71.5 ± 0.50
logP- & HBA+	0–0	9.88 ± 1.03	8.64 ± 2.06	9.09 ± 1.25	14.00 ± 2.48	27.87 ± 3.86	19.8 ± 0.62	66.6 ± 0.00	86.5 ± 1.29	87.7 ± 0.76	**88.3 ± 0.58**
	0.5–1	0.23 ± 0.33	0.45 ± 0.64	0.22 ± 0.31	0.67 ± 0.62	8.80 ± 0.04	1.2 ± 0.24	54.9 ± 0.34	72.5 ± 0.23	72.3 ± 2.02	**75.2 ± 2.27**
logP+ & HBA+	0–0	2.99 ± 0.38	2.00 ± 0.58	2.45 ± 0.67	7.17 ± 0.85	8.55 ± 2.75	1.2 ± 0.47	4.3 ± 0.62	23.0 ± 4.36	30.7 ± 1.26	**33.3 ± 0.29**
	0.5–1	0.45 ± 0.32	0.00 ± 0.00	0.22 ± 0.31	0.17 ± 0.24	2.93 ± 0.30	0.0 ± 0.00	0.2 ± 0.24	1.7 ± 0.29	18.2 ± 0.76	**20.0 ± 0.00**
logP- & HBD+	0–0	2.28 ± 1.15	2.23 ± 1.16	4.44 ± 0.58	13.83 ± 2.95	33.51 ± 4.08	23.0 ± 1.41	79.8 ± 0.62	90.5 ± 0.00	90.3 ± 0.29	**92.0 ± 1.00**
	0.5–1	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	9.98 ± 1.03	0.7 ± 0.24	3.3 ± 0.24	75.8 ± 2.75	75.7 ± 0.29	**76.8 ± 1.76**
logP+ & HBD+	0–0	0.69 ± 0.58	1.96 ± 0.87	1.79 ± 0.66	5.67 ± 0.62	17.03 ± 2.75	3.3 ± 0.47	2.8 ± 0.24	36.2 ± 2.84	42.5 ± 1.80	**51.7 ± 1.61**
	0.5–1	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	2.59 ± 1.14	0.0 ± 0.00	0.0 ± 0.00	0.8 ± 0.58	4.8 ± 1.04	**10.0 ± 4.50**
logP- & tPSA-	0–0	5.06 ± 1.21	3.53 ± 0.38	4.88 ± 2.21	8.17 ± 1.03	**35.69 ± 3.19**	0.8 ± 0.24	10.0 ± 0.71	5.0 ± 0.50	8.0 ± 0.50	8.3 ± 2.47
	0.5–10	1.16 ± 0.68	0.67 ± 0.55	0.66 ± 0.54	0.00 ± 0.00	**19.15 ± 0.73**	0.8 ± 0.24	2.8 ± 0.24	0.2 ± 0.29	1.3 ± 0.58	1.7 ± 0.76
logP- & tPSA+	0–0	12.17 ± 1.05	10.43 ± 2.88	13.08 ± 2.28	19.83 ± 2.46	44.35 ± 0.68	13.7 ± 1.25	75.8 ± 1.25	**92.0 ± 1.73**	91.5 ± 0.50	91.8 ± 1.61
	0.5–10	6.20 ± 0.64	6.23 ± 2.31	6.67 ± 0.53	4.83 ± 0.85	28.67 ± 2.22	12.3 ± 1.25	66.0 ± 0.71	82.0 ± 1.50	82.7 ± 0.29	**83.2 ± 0.58**

2.8.2. Dataset construction

We start with the high-quality HiQBind database, which provides experimentally validated protein–ligand complexes with curated binding affinities. We filter for ligands that:

- have complete property annotations (MW, LogP, HBD, HBA, QED),
- contain at most 50 heavy atoms to ensure manageable complexity,
- violate at least one Lipinski rule (MW $>$ 500, LogP $>$ 5, HBD $>$ 5, or HBA $>$ 10),
- and additionally satisfy quality checks (QED $\geq$ 0.1, MW $\leq$ 800, LogP $\leq$ 8) to maintain feasible optimization targets.

We remove duplicates based on SMILES strings and random sample 1,000 ligands to form LipinskiFix-1000. Each entry is paired with its protein context and annotated with detailed property values, violated rules, and explicit primary (Lipinski compliance) and secondary (QED maximization) optimization objectives.

2.8.3. Molecular Property Optimization Dataset

We evaluate AutoLead on two standard molecular property optimization benchmarks from ChatDrug and DrugAssist that give diverse optimization scenarios.

2.8.4. ChatDrug-200

We use 200 molecules curated from the ChatDrug framework, which systematically investigates conversational drug editing with LLMs. These molecules cover diverse single- and multi-objective optimization tasks, such as adjusting LogP, QED, permeability (mesured by TPSA), and modifying hydrogen bond acceptor/donor counts, under loose and strict improvement thresholds Δ.

2.8.5. DrugAssist-500

We additionally evaluate on 500 molecules curated from the DrugAssist benchmark, designed to test interactive multi-property optimization. Tasks include simultaneous optimization of properties like BBBP, hERG inhibition, solubility, and QED, often requiring target values to lie within specified ranges while preserving structural similarity. Since RDKit can not directly compute BBBP and hERG inhibition, we focus on solubility, QED, and acceptor/donor tasks for optimization under both loose and strict improvement threshold (Δ) constraints.

2.9. Evaluation Metrics

We adopt the evaluation metrics used in Liu et al. [2024], Ye et al. [2025] to assess AutoLead's performance across various optimization scenarios:

Success Rate: Percentage of molecules successfully optimized to meet specified thresholds.

Valid Rate: Percentage of chemically valid molecules.

3. Results and Discussion

This section presents our experimental results. We begin by describing the experimental setup, including the datasets and baseline models. We then report and analyze the performance of our proposed method.

3.1. Experimental Setup

To ensure the reliability of our results, all methods were evaluated over 3 independent runs for each task. We report the mean and standard deviation across these runs.

3.1.1. Specifications for AutoLead

We evaluate three variants of AutoLead to specifically demonstrate how the hybrid strategy improves performance:

- **LLMO (LLM Optimizer)**: An LLM-guided approach that leverages the large language model throughout all iterations to propose candidate molecules based on learned chemical priors.
- **BO (Bayesian Optimizer)**: A special case of the hybrid strategy where BO is used in every round ($z_t \equiv 1$), allowing the surrogate model to fully guide the optimization from start to finish.
- **HO (Hybrid Optimizer)**: A transient strategy that combines LLM exploration and Gaussian Process-based Bayesian Optimization (BO). It initially relies on the LLM to gather diverse molecules and switches to BO once a minimum sequence size ($\tau = 3$ points) is reached, allowing exploitation of the surrogate model Chang et al. [2025].

For both variants, we set the maximum number of optimization iterations T to 10 for each molecule. The process stops early if a molecule satisfying the target property threshold is found, or continues until the iteration limit is reached. All prompt templates used in this work are provided in the Supplementary Data.

3.2. Baselines Methods

The baselines Liu et al. [2023] are based on MegaMolBART Irwin et al. [2022], a pretrained auto-regressive model. Baselines include Random, PCA, High-Variance, GS-Mutate Jensen [2019], and MoleculeSTM Liu et al. [2023] with SMILES and ChatDrug with GPT-3.5 Turbo and GPT-4o Hurst et al. [2024] as the backbone LLMs. For the ChatDrug baselines, we set the number of conversation rounds to 10, and reran their implementation to be precisely aligned with the evaluation metrics, as also noted in Ye et al. [2025]. For the other baselines, we adopted the values reported in Liu et al. [2024].

3.3. Performance on the ChatDrug-200 dataset

Our proposed hybrid optimization method, AutoLead-HO, demonstrates a commanding performance on the ChatDrug-200 benchmark, significantly outperforming both existing state-of-the-art methods and its own ablation variants, AutoLead-LLMO and AutoLead-BO.

In the single-objective setting (Table 1), AutoLead-HO consistently achieves top success rates, especially under the most stringent test conditions. For instance, on challenging tasks such as increasing the hydrogen bond acceptor count (HBA+) or optimizing for QED+, our method significantly outperforms all baselines, including the strong ChatDrug with GPT-4o model. This highlights the robustness of our hybrid strategy in focused optimization scenarios.

In the more demanding 12 multi-objective tasks (in the bottom six rows of Table 1), the superiority of our hybrid approach is further confirmed. The AutoLead framework achieved the best performance on all tasks except for logP- & tPSA-, with AutoLead-HO consistently delivering the top performance in the most complex scenarios. While the MoleculeSTM-SMILES baseline is strong in tasks involving tPSA reduction, our method dominates elsewhere. For instance, when simultaneously decreasing logP and increasing hydrogen bond donors (logP- & HBD+) under strict thresholds ($\Delta = 0.5 - 1$), AutoLead-HO's success rate of **76.8%** is dramatically higher than ChatDrug's **3.3%**. Likewise, for increasing both logP and HBA+ ($\Delta = 0.5 - 1$), our hybrid method achieves a **20.0%** success rate where the previous state-of-the-art only achieves **0.2%**. For valid rate, please Table S3.

3.4. Performance on the DrugAssist-500 dataset

Results on the DrugAssist-500 dataset (Table S2) further confirm the generalizability and effectiveness of our proposed hybrid method, AutoLead-HO, in interactive multi-property optimization scenarios. The AutoLead framework, including its LLMO and BO variants, demonstrates superior performance by achieving the top success rate across most of the tasks. This outperformance is especially pronounced in the more challenging strict settings. For instance, in the complex multi-property task of decreasing logP while increasing HBA+ with a stringent threshold ($\Delta = 1$), our hybrid approach HO achieves a success rate of 72.0%, a dramatic improvement over the 0.8% from ChatDrug. In another difficult task, increasing the number of hydrogen bond donors (HBD+) with a strict threshold of $\Delta = 1$, our BO variant leads with a 70.7% success rate, again showcasing a substantial leap from the 4.2% achieved by ChatDrug with GPT-4o. For valid rate, please refer Table S4.

Table 2. Success rate (%) on the LipinskiFix-1000 dataset. The evaluation metrics is the success rate (percentage of molecules that satisfy both Lipinski's Rule of Five and achieve QED $\geq$ 0.7). The best results are marked in **bold**.

Method	Success Rate (%)
Random	11.50 ± 3.65
PCA	11.90 ± 4.04
High Variance	12.30 ± 4.95
GS-Mutate	1.50 ± 0.23
MoleculeSTM-SMILES	13.90 ± 2.55
ChatDrug (GPT-3.5-turbo)	3.87 ± 0.06
ChatDrug (GPT-4o)	3.83 ± 0.29
AutoLead-LLMO	2.50 ± 0.21
AutoLead-BO	20.30 ± 0.85
AutoLead-HO	**28.33 ± 0.66**

3.5. Performance on the LipinskiFix-1000 Dataset

We evaluate AutoLead on the LipinskiFix-1000 dataset, a realistic benchmark that requires methods to restore Lipinski's Rule of Five compliance while simultaneously improving QED. This task represents a stringent, real-world multi-objective challenge common in early-stage drug discovery.

As summarized in Table 2, our hybrid strategy (AutoLead-HO) demonstrates a significant advantage. It achieves a 28.9% success rate, considerably outperforming previous approaches like MoleculeSTM (13.9%) and ChatDrug (3.87%). In terms of valid rate (Table S5), AutoLead-HO also performs exceptionally well at 93.6%, substantially outperforming methods like MoleculeSTM (49.4%) and ChatDrug (79.3%). The strong performance confirms

the framework's potential for guiding the development of drug-like molecules in realistic lead optimization pipelines.

3.6. Impact of LLM-driven Inverse Mapping

To evaluate the effectiveness of our proposed LLM-driven inverse mapping, we conducted two key analyses.

First, we compared our approach against a conventional deep learning decoder. We replaced the LLM-driven decoder in AutoLead-BO with a Variational Autoencoder (VAE) decoder Polykovskiy et al. [2020], testing latent space dimensionalities of 16, 32, and 128. As shown in Table S6, when evaluated on representative single- and multi-objective tasks from the ChatDrug-200 dataset (QED+ and logP+ & HBD+), all VAE-based variants underperformed compared to our LLM-driven approach. This result highlights the difficulty of using standard decoders to map from a property-defined latent space back to valid, high-quality molecules, validating our novel method.

Second, we sought to confirm that the inverse mapping step itself contributes meaningfully to the optimization process. According to our framework (Algorithm 1), if the LLM-driven decoder fails or generates an invalid SMILES, the system falls back to direct generation via the LLM (G_{LLM}). To isolate the contribution of the decoder, we analyzed the performance of molecules generated by each path on the LipinskiFix-1000 benchmark. The results, presented in Table S7, clearly demonstrate that molecules generated via the LLM inverse mapping achieve a significantly higher valid rate and success rate than those produced by the direct LLM generation fallback. This finding confirms that our LLM-driven decoder is a critical component that effectively translates optimized latent vectors into successful molecular candidates, rather than relying on the fallback mechanism.

3.7. Visualization of Successful Lead Optimizations.

To further illustrate the practical effectiveness of AutoLead, Figure 2 showcases representative examples of successful lead optimizations. These examples demonstrate AutoLead's ability to perform complex edits, transforming initial molecules into more favorable candidates by improving key drug-like properties. Notably, we observe significant increases in QED alongside improvements in Lipinski's Rule of Five compliance. For instance, several transformations show substantial reductions in MW and logP, as well as increases in HBD and HBA counts to fall within desirable ranges. The modifications, such as the strategic introduction of fluorine atoms or subtle structural alterations like ring contractions and functional group removals, are chemically sensible and preserve the core structural motifs of the original molecules. This highlights AutoLead's capability to generate realistic and relevant chemical transformations aligned with medicinal chemistry principles during early-stage lead optimization. We also provide examples of successful molecule refinements across the ChatDrug-200 and DrugAssist-500 datasets in Figure S2.

3.8. Dynamic Usage of Bayesian Optimization in HO.

To understand the contribution of Bayesian Optimization (BO) within our hybrid optimization (HO) strategy, we analyze the proportion of steps where BO was selected during the molecular editing process. The results, summarized in Table S8, reveal that HO dynamically adjusts its reliance on BO based on task difficulty. For simpler property edits like modifying logP+ ($\Delta = 0$), where LLM-based exploration is often sufficient, the BO ratio remains very low (0.17%). In contrast, for tasks requiring more precise, data-driven exploitation, the reliance on BO increases substantially. This is evident in challenging single-objective tasks like QED+ (56.19%) and complex multi-objective cases such as logP+&HBD+ (56.62%). This trend culminates in the LipinskiFix-1000 benchmark, which involves satisfying multiple strict constraints, where BO usage reaches its peak at 64.8%. These findings validate our hybrid design, which effectively balances broad, LLM-driven exploration with focused, BO-driven exploitation according to the complexity of the optimization task.

3.9. Impact of BO Integration and LLM Choice.

As shown in Figure S1, integrating Bayesian Optimization (BO) yields substantial performance gains. The hybrid (HO) strategy consistently outperforms the LLM-only (LLMO) approach, regardless of the LLM backbone, while maintaining a high molecule valid rate above 85%.

The analysis also reveals a crucial synergy between the hybrid method and a more advanced LLM. The performance benefit from upgrading the LLM from GPT-3.5-turbo to GPT-4o is magnified under the HO strategy (+2.1%) compared to the modest gain for the LLMO strategy (+0.6%). This indicates that BO's structured exploration is most effective when paired with a powerful LLM. We hypothesize that such a more capable model serves as a more effective inverse mapping function, $\tilde{\phi}^{-1}$, for decoding optimized latent vectors into valid, high-quality molecular structures.

3.10. A Richer Feature Space is Crucial for Optimization Success.

We also analyze AutoLead-HO by varying the dimensionality of the BO feature space, comparing a comprehensive 10-dimension (10D) space against smaller 2D and 5D subsets. The 10D space included key descriptors like molecular weight, LogP, TPSA, and hydrogen bond counts. As shown in Table S9, the 10D model performs better on average across the ChatDrug-200 and DrugAssist-500 datasets. The advantage of a richer feature space became most apparent on the more challenging LipinskiFix-1000 benchmark. Here, the 10D model's success rate of 28.9% represented an order-of-magnitude improvement over the 2D (2.9%) and 5D (2.3%) models. This demonstrates that while a minimal feature set may suffice for simpler tasks, a comprehensive set of molecular descriptors is critical for navigating complex, multi-objective optimization landscapes and achieving robust performance.

4. Conclusion

In this work, we introduce AutoLead, a novel framework that integrates the chemical reasoning of LLMs with the principled exploration of multi-objective Bayesian optimization to address realistic lead optimization challenges. Unlike prior approaches that rely on narrow objectives or handcrafted features, AutoLead combines the intuitive generative power of LLMs with the uncertainty-aware search of Gaussian Processes to efficiently explore complex molecular spaces.

Across three diverse benchmarks—ChatDrug-200, DrugAssist-500, and the newly introduced LipinskiFix-1000, which simulates a realistic drug-likeness evaluation scenario—AutoLead consistently

outperforms existing methods. Further, our hybrid strategy excels on challenging multi-property tasks, surpassing both LLM-only and BO-only baselines. These results underscore the potential of coupling LLMs with black-box optimization for molecular design, paving the way for more robust and scalable solutions in computational drug discovery.

5. Supplementary data

Submitted a supplementary material to provide additional results.

6. Data availability

The LipinskiFix-1000 data are provided in figshare.

References

R. S. Bohacek, C. McMartin, and W. C. Guida. The art and practice of structure-based drug design: a molecular modeling perspective. *Medicinal research reviews*, 16(1):3–50, 1996.

G. W. Caldwell, Z. Yan, W. Tang, M. Dasgupta, and B. Hasting. Adme optimization and toxicity assessment in early-and late-phase drug discovery. *Current topics in medicinal chemistry*, 9 (11):965–980, 2009.

C.-Y. Chang, M. Azvar, C. E. Okwudire, and R. Al Kontar. Llinbo: Trustworthy llm-in-the-loop bayesian optimization. *CoRR*, 2025.

J. Chu, J. Park, S. Lee, and H. J. Kim. Inversion-based latent bayesian optimization. *Advances in Neural Information Processing Systems*, 37:68258–68286, 2024.

V. Dey, X. Hu, and X. Ning. Large language models for controllable multi-property multi-objective molecule optimization. *arXiv preprint arXiv:2505.23987*, 2025.

P. J. Eddershaw, A. P. Beresford, and M. K. Bayliss. Adme/pk as part of a rational approach to drug discovery. *Drug discovery today*, 5(9):409–414, 2000.

B. Gao, Y. Huang, Y. Liu, W. Xie, W.-Y. Ma, Y.-Q. Zhang, and Y. Lan. Pharmagents: Building a virtual pharma with large language model agents. *arXiv preprint arXiv:2503.22164*, 2025.

A. E. Ghareeb, B. Chang, L. Mitchener, A. Yiu, C. J. Szostkiewicz, J. M. Laurent, M. T. Razzak, A. D. White, M. M. Hinks, and S. G. Rodriques. Robin: A multi-agent system for automating scientific discovery. *arXiv preprint arXiv:2505.13400*, 2025.

R. B. Gramacy. *Surrogates: Gaussian process modeling, design, and optimization for the applied sciences*. Chapman and Hall/CRC, 2020.

A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

R. Irwin, S. Dimitriadis, J. He, and E. J. Bjerrum. Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3(1):015022, 2022.

J. 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.

K. Li, Z. Wu, S. Wang, and W. Hu. Drugpilot: Llm-based parameterized reasoning agent for drug discovery. *arXiv preprint arXiv:2505.13940*, 2025.

C. A. Lipinski. Lead-and drug-like compounds: the rule-of-five revolution. *Drug discovery today: Technologies*, 1(4):337–341, 2004.

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*, 64:4–17, 2012.

S. Liu, W. Nie, C. Wang, J. Lu, Z. Qiao, L. Liu, J. Tang, C. Xiao, and A. Anandkumar. Multi-modal molecule structure–text model for text-based retrieval and editing. *Nature Machine Intelligence*, 5(12):1447–1457, 2023.

S. Liu, J. Wang, Y. Yang, C. Wang, L. Liu, H. Guo, and C. Xiao. Conversational drug editing using retrieval and domain feedback. In *The twelfth international conference on learning representations*, 2024.

D. Polykovskiy, A. Zhebrak, B. Sanchez-Lengeling, S. Golovanov, O. Tatanov, S. Belyaev, R. Kurbanov, A. Artamonov, V. Aladinskiy, M. Veselov, et al. Molecular sets (moses): a benchmarking platform for molecular generation models. *Frontiers in pharmacology*, 11:565644, 2020.

RDKit. : Open-source cheminformatics. `https://www.rdkit.org`.

A. S. Rich, Y. H. Chan, B. Birnbaum, K. Haider, J. Haimson, M. Hale, Y. Han, W. Hickman, K. P. Hoeflich, D. Ortwine, et al. Machine learning adme models in practice: Four guidelines from a successful lead optimization case study. *ACS Medicinal Chemistry Letters*, 15(8):1169–1173, 2024.

Y. Wang, K. Sun, J. Li, X. Guan, O. Zhang, D. Bagni, Y. Zhang, H. A. Carlson, and T. Head-Gordon. A workflow to create a high-quality protein–ligand binding dataset for training, validation, and prediction tasks. *Digital Discovery*, 4(5):1209–1220, 2025.

D. 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.

C. K. Williams and C. E. Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

R. Winter, F. Montanari, A. Steffen, H. Briem, F. Noé, and D.-A. Clevert. Efficient multi-objective molecular optimization in a continuous latent space. *Chemical science*, 10(34):8016–8024, 2019.

G. Ye, X. Cai, H. Lai, X. Wang, J. Huang, L. Wang, W. Liu, and X. Zeng. Drugassist: A large language model for molecule optimization. *Briefings in Bioinformatics*, 26(1):bbae693, 2025.

A. Yong, A. Tripp, L. Hosseini-Gerami, and B. Paige. Bayesian optimization for molecules should be pareto-aware. *arXiv preprint arXiv:2507.13704*, 2025.

J. Yu, Y. Zheng, H. Y. Koh, S. Pan, T. Wang, and H. Wang. Collaborative expert llms guided multi-objective molecular optimization. *arXiv preprint arXiv:2503.03503*, 2025.

Z. Zeng, B. Yin, S. Wang, J. Liu, C. Yang, H. Yao, X. Sun, M. Sun, G. Xie, and Z. Liu. Chatmol: interactive molecular discovery with natural language. *Bioinformatics*, 40(9):btae534, 2024.

D. Zhang, W. Liu, Q. Tan, J. Chen, H. Yan, Y. Yan, J. Li, W. Huang, X. Yue, W. Ouyang, et al. Chemllm: A chemical large language model. *arXiv preprint arXiv:2402.06852*, 2024.

Q. Zhang, K. Ding, T. Lv, X. Wang, Q. Yin, Y. Zhang, J. Yu, Y. Wang, X. Li, Z. Xiang, et al. Scientific large language models: A survey on biological & chemical domains. *ACM Computing Surveys*, 57(6):1–38, 2025.

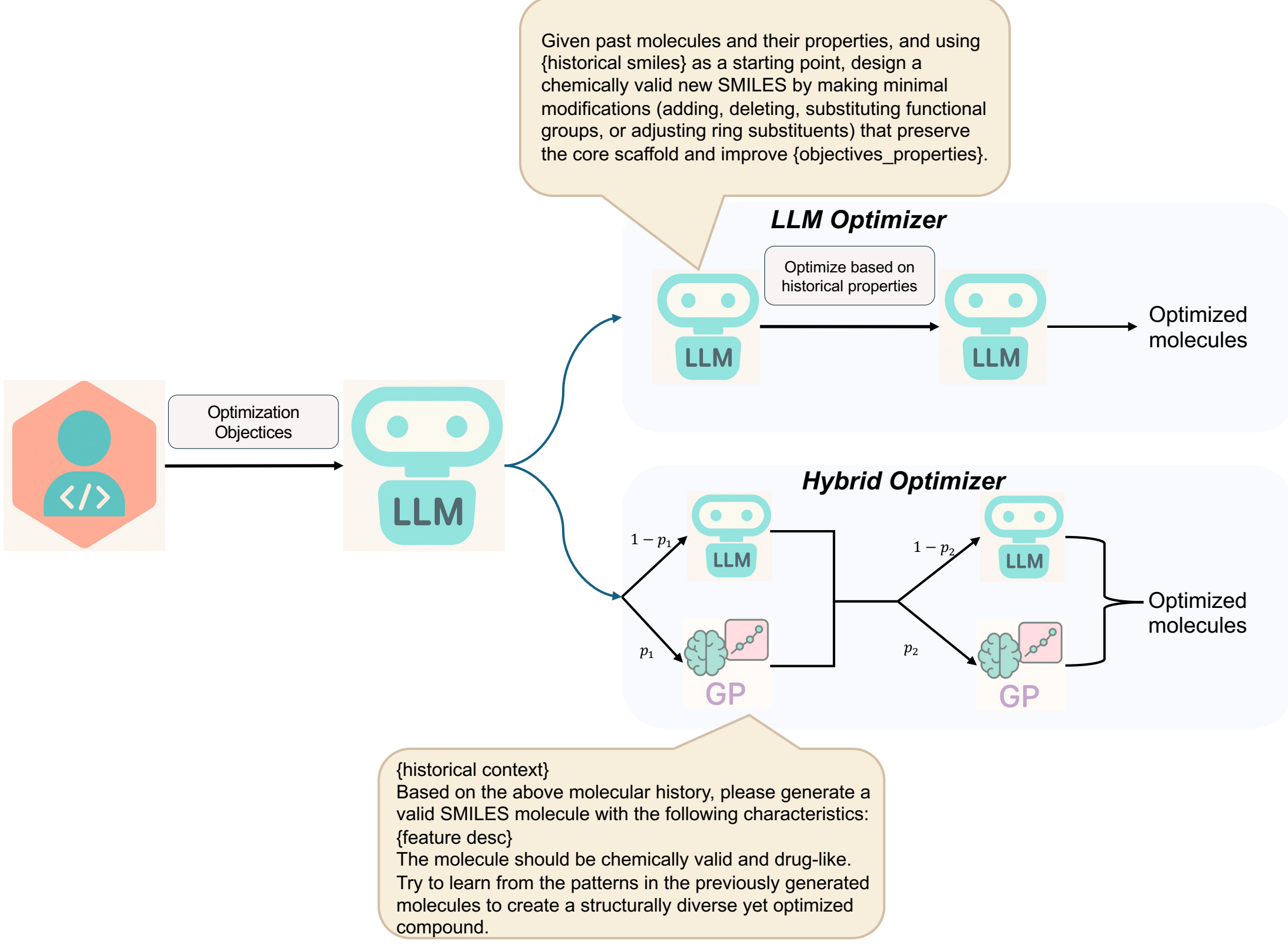

Figure 1. Overview of the AutoLead framework. The system combines an LLM-based optimizer that leverages historical molecular properties and a hybrid optimizer integrating both LLM and GP components. Optimization balances LLM chemical reasoning with uncertainty-aware BO to progressively refine molecules toward multi-objective drug-like criteria.

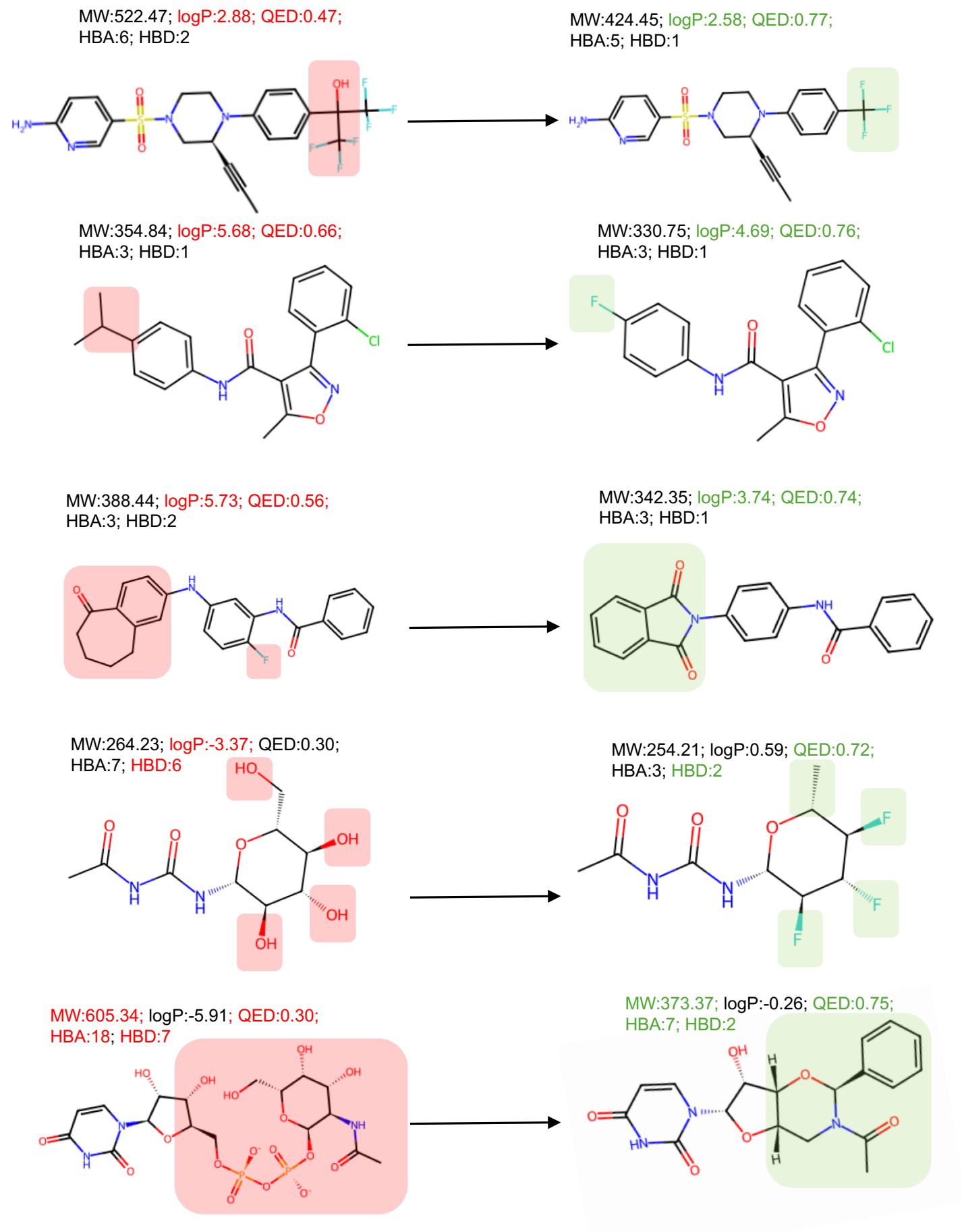

Figure 2. Examples of successful molecule refinement by AutoLead on the LipinskiFix-1000 dataset. For each original (left) and optimized (right) pair, key properties are displayed. Green indicates a favorable change, while red flags the initial property that failed to meet drug-likeness criteria.

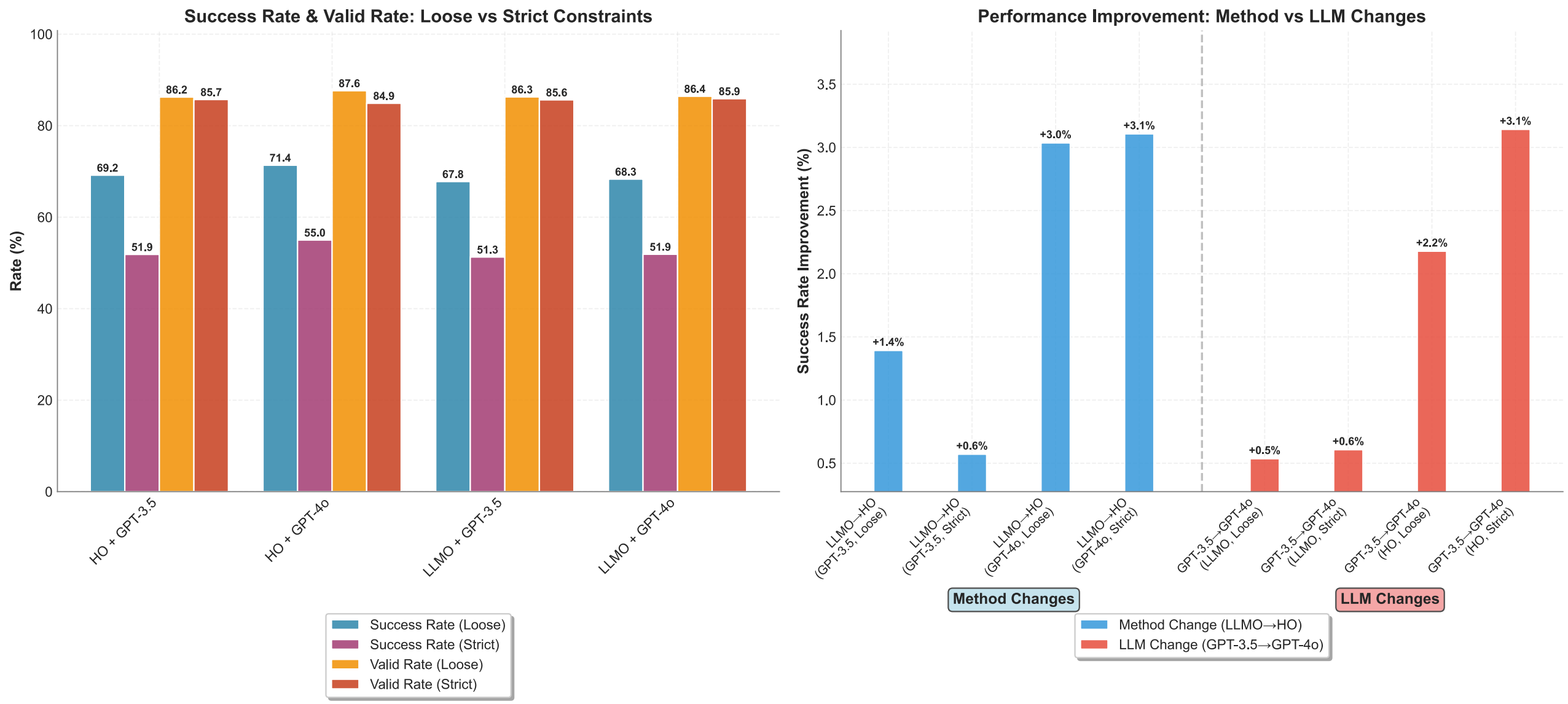

Figure S1. Performance comparison on the ChatDrug-200 dataset.
Left: Success rates and valid rates under both loose and strict constraint settings for different variants of HO and LLMO. **Right:** Success rate improvements attributable to method changes (LLMO → HO) versus LLM backbone changes (GPT-3.5 → GPT-4o). These results highlight the relative contribution of each factor to overall performance gains.

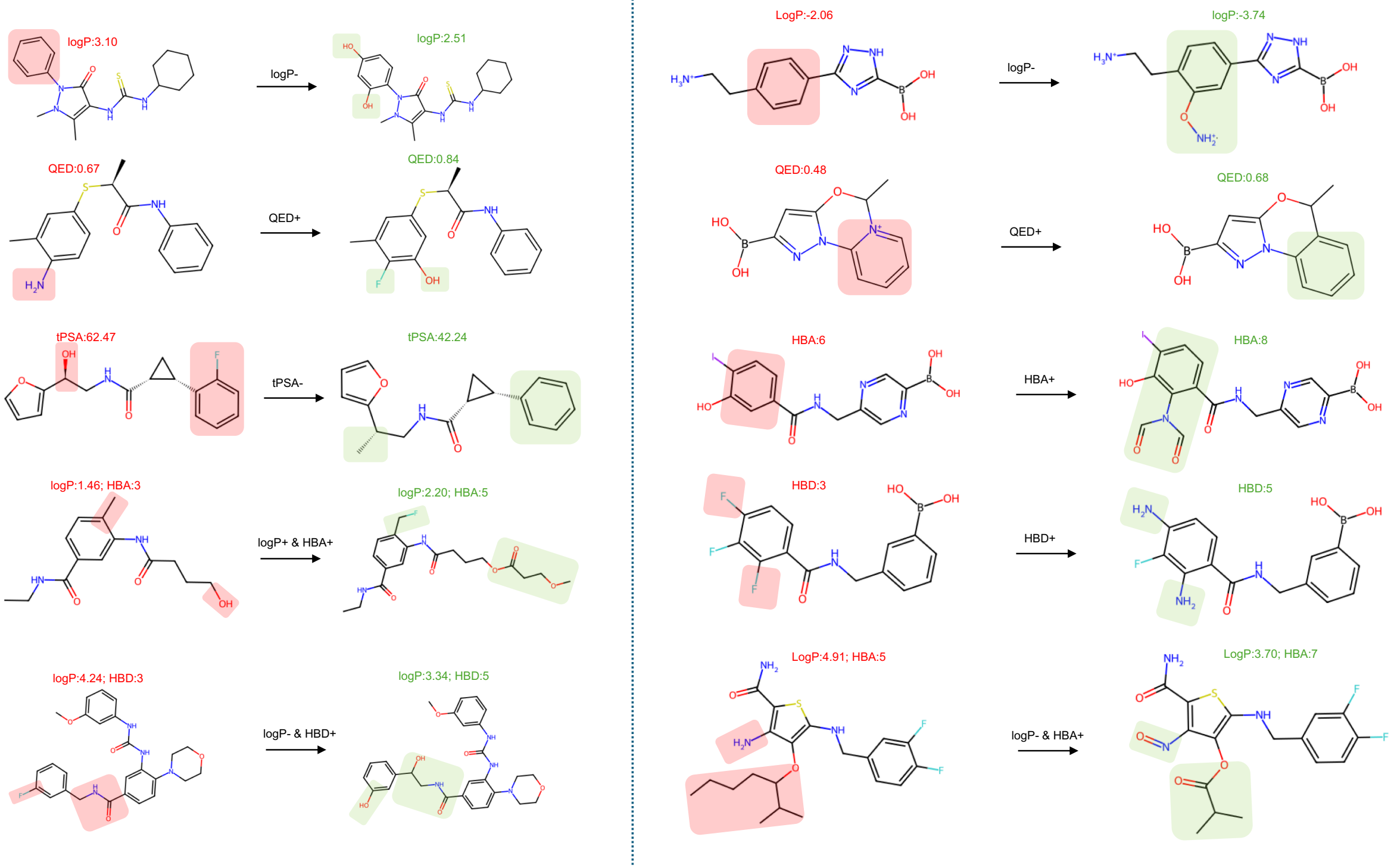

Figure S2. Examples of molecule refinements generated by AutoLead-HO across the ChatDrug-200 and DrugAssist-500 datasets. Each pair shows the original molecule (left) and the optimized molecule (right) with corresponding key property values. Red highlights indicate properties or structural regions in the initial molecule that violated optimization criteria, while green highlights show improvements or favorable modifications introduced by AutoLead-HO.