Sample Efficient Generative Optimization for Molecular Design

Anonymous Authors[1]

Abstract

Discovering optimal molecules, whether in drug discovery, materials design, or catalyst optimization, often requires navigating large chemical spaces with very limited data. Two families of methods have emerged: Bayesian Optimization (BO), which is highly sample efficient but typically operates over fixed, user-defined libraries, and goal-directed generative models, which can explore chemical space freely but often require hundreds or thousands of oracle calls to find promising candidates. This creates a practical tension between sample efficiency and chemical space coverage, yet both are essential for real-world campaigns in which the target region of chemical space is unknown and data is expensive to collect. We introduce Sample Efficient Generative Optimization (SEGO), a framework combining generative modeling with Bayesian optimization for molecular discovery. At each iteration, SEGO uses a surrogate model to focus generation on promising regions of chemical space, then applies BO over the resulting candidates to select the most informative molecules for evaluation. SEGO attains state-of-the-art on the Practical Molecular Optimization benchmark in only one tenth of the oracle calls consumed by other methods, opening the door to optimization campaigns driven by direct experimental feedback.

1. Introduction

Molecular optimization is a central problem in chemistry, with applications ranging from drug discovery to catalyst design (Filella-Merce et al., 2025; Strieth-Kalthoff et al., 2024; Seumer et al., 2023). These tasks require identifying molecules that satisfy multiple, often competing objectives from vast chemical space (Sanchez-Lengeling & Aspuru-Guzik, 2018). Most relevant optimization problems are also data scarce: phenomena such as protein-ligand binding or catalytic activity are difficult to simulate accurately, and experimental measurements—particularly those requiring molecular synthesis—are costly and time-consuming. As a result, the real-world impact of an optimization strategy is defined by the number of evaluations required to discovery optimal molecules (sample efficiency). Low sample efficiency necessitates reliance on inexpensive, low-fidelity *in silico* oracles vulnerable to hacking (Renz et al., 2019; Lehman et al., 2020). Improved sample efficiency would enable the use of higher-fidelity simulations or even direct experimental feedback, thereby increasing the practical impact of molecular optimization methods.

Goal-directed generative models are a class of methods for molecular design that use machine learning to generate candidate molecules from scratch. Diverse model types have been used, including recurrent neural networks (RNNs) (Olivecrona et al., 2017; Loeffler et al., 2024), transformers (Feng et al., 2023; Mazuz et al., 2023), GFlowNets (Bengio et al., 2021) , genetic algorithms (Jensen, 2019), and diffusion or flow-based models (Lee et al., 2023; Jiang et al., 2024; Schneuing et al., 2023). Language models remain among the most sample efficient methods, often using reinforcement learning to guide generation, in the form of Simplified Molecular Input Line Entry System (SMILES) (Weininger, 1988), towards high-scoring regions of chemical space. State-of-the-art generative models still require between hundreds and thousands of oracle evaluations to identify high-quality candidates (Guo et al., 2026; Gao et al., 2022). While this cost may permit the use of moderately expensive simulations such as DFT, it remains prohibitive for workflows involving direct experimental evaluation.

Bayesian Optimization (BO) is an alternative paradigm based on a probabilistic surrogate model and an acquisition function to guide sequential experimentation within a candidate library or a continuous latent space (Frazier, 2018). BO has excellent sample efficiency, and has been used to identify promising candidates in tens to hundreds of evaluations, but its performance depends on the presence of suitable candidates within a library or accurate decoding from a latent space to molecule space (Agarwal et al., 2021; Griffiths et al., 2022). Moreover, BO is highly sensitive to the choice of molecular representation, which can

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

be challenging to design, expensive to compute, and may not generalize across diverse regions of chemical space (Ranković et al., 2025). Recent BO workflows use large language models (LLMs) trained jointly with GP objectives enable the rapid, flexible featurization only from SMILES with excellent performance across diverse chemical tasks, opening the door to on-the-fly application across chemical space (Kristiadi et al., 2024; Ranković et al., 2025; Ranković & Schwaller, 2023).

We propose the Sample Efficient Generative Optimization (SEGO) framework, which unifies generative modeling with Bayesian optimization for sample efficient search in open-ended molecular space. At each iteration, a surrogate-guided generative model produces a library of molecules in promising regions of chemical space. This targeted library is featurized through deep kernel learning based on LLM features (Ranković et al., 2025) and the next most informative experiment is selected with an acquisition function. SEGO further achieves sample efficiency through SMILES augmentation in the surrogate training, and an anchoring mechanism for the surrogate-guided generation. The hybrid architecture improves on both of its parent methods, identifying hit molecules on a toy oracle in an order of magnitude fewer evaluations. SEGO further attains state-of-the-art in the practical molecular optimization benchmark in only a tenth of the oracle budget. These gains could enable prospective campaigns beyond predefined libraries based on direct experimental feedback.

2. Background and related work

2.1. *De Novo* Molecular Design.

De novo molecular design seeks to generate molecules with desired properties from scratch, bypassing the constraints of fixed libraries (Sanchez-Lengeling & Aspuru-Guzik, 2018). Generative approaches span a broad range of architectures, from SMILES-based recurrent networks and transformers to graph-based, diffusion, and flow-based models. Among these, language-based models operating on SMILES representations have proven particularly sample efficient, capable of satisfying even three-dimensional objectives such as docking despite working from one-dimensional string inputs (Gao et al., 2022; Brown et al., 2019). A key advantage of SMILES is its non-injective nature: a single molecule admits many valid string representations, and this redundancy can be leveraged through augmentation to improve generalization in low-data regimes (Guo & Schwaller, 2024a; Arús-Pous et al., 2019). As the field has shifted toward stricter oracle budgets that better reflect real-world constraints, sample efficiency has become a central evaluation criterion, and language-based methods consistently rank among the top performers under these conditions. Recent advances such as combining experience replay with SMILES augmentation have pushed sample efficiency further, establishing strong baselines for generative molecular optimization (Guo et al., 2026).

However, based on the formulation of the loss, these systems do not learn much from negative examples early in the optimization (Loeffler et al., 2024). In challenging optimization scenarios, where early batches yield little to no reward, slow learning causes long lag-times which erodes sample efficiency. This motivates integrating generative models with principled selection strategies that extract signal from all evaluations, even from negative evaluations early in the optimization trajectory.

2.1.1. Bayesian Optimization on Molecules.

Bayesian optimization has been applied to molecular discovery across domains in chemistry. For example, it has been used to identify redox-active molecules from libraries of scaffolds enumerated with substituents and featurized with DFT descriptors (Agarwal et al., 2021), to identify candidate protein binders from an Enamine library using fingerprint representations (Graff et al., 2021), and to materials discovery, where sets of predefined building blocks were evaluated using time-dependent DFT (Strieth-Kalthoff et al., 2024). More broadly, library-based BO has been applied to transition-metal complexes (Janet et al., 2020), metal-organic frameworks (Comlek et al., 2023), with surrogate models trained on small labeled subsets and acquisition functions used to iteratively select candidates for evaluation. These case studies illustrate the strength of BO for optimization in challenging reward landscapes, but also highlight two recurring limitations: optimization is constrained to expert-crafted libraries that cannot propose structures beyond their initial enumeration, and performance depends heavily on molecular representation. Bespoke descriptors such as DFT features can be expensive to compute, while general-purpose fingerprints, though cheap, may not capture the structure-property relationships relevant to a given task.

Recent work has begun to address the representation challenge through language-model-based molecular embeddings (Kristiadi et al., 2024; Ranković & Schwaller, 2023; Ranković et al., 2025). Building on this direction, the Gollum framework fine-tunes language model embeddings using Gaussian Process (GP) marginal likelihood, yielding task-adaptive representations that outperform both fingerprints and bespoke descriptors across diverse benchmarks (Ranković et al., 2025). Crucially, because Gollum operates directly on SMILES strings without requiring additional descriptor computation, it can featurize novel molecules on-the-fly, making it feasible to apply BO over dynamically generated libraries rather than fixed, predefined candidate sets.

2.1.2. Hybrid Generative BO methods.

To move beyond fixed libraries, several works have coupled generative models with BO. A common approach embeds discrete molecular representations into a continuous latent space, performs optimization there, and decodes the resulting vectors back into molecules (Filella-Merce et al., 2025; Gómez-Bombarelli et al., 2018). However, this latent-optimize-then-decode paradigm faces practical challenges: decoded molecules are frequently invalid or redundant, and latent spaces can be high-dimensional and poorly structured, complicating surrogate modeling and uncertainty quantification. Alternative approaches bypass latent embeddings and optimize acquisition functions directly over discrete representations using grammar-constrained search (Moss et al., 2020), graph-based kernels (Oh et al., 2019), or learned generation policies (Swersky et al., 2020) though these methods face their own difficulties in maintaining chemical plausibility and scaling to large batch sizes.

A more recent line of work decouples the generative and optimization stages entirely, first using a generative model to propose candidates and then applying BO to select the most informative subset for evaluation. Dodds et al. (2024) developed a hybrid framework combining REINVENT with a fingerprint-driven, random-forest-based active learning loop, demonstrating improved sample efficiency over baseline REINVENT, though the modest efficiency of the underlying generative model limits practical applicability. Tripp & Hernández-Lobato (2024) used a genetic algorithm as a molecule generation engine and a Tanimoto kernel on molecular fingerprints to achieve excellent results on the PMO benchmark. Muthyala et al. (2025) also proposed a generate-then-optimize system with a new acquisition function for batch selection. While SEGO shares this hybrid format, it introduces surrogate-guided generation and augmented optimization to improve sample efficiency substantially. Further, SEGO builds on frontier methods in both Bayesian optimization and generative design, enabling new levels of sample efficiency not achieved by prior hybrid approaches.

3. SEGO Design

SEGO is a hybrid framework for generative optimization that attains excellent sample efficiency through the intelligent allocation of oracle calls across dynamically generated libraries (Figure 1). Each oracle call drives updates to a generative agent, a GP surrogate model, and the surrogate's learned featurization. SEGO is organized around the following components:

- An anchor generative agent $\pi_{\theta_{agent}}$ updated only on true-oracle feedback.
- An inner-loop agent $\pi_{\theta_{inner}}$, re-instantiated from each BO iteration from the anchor and trained against the GP posterior mean to produce a dynamic candidate library.
- A deep-kernel GP surrogate over LLM embeddings, which featurizes the dynamic library for selection via an acquisition function.

SEGO's sample efficiency rests on two contributions: an anchored surrogate-guided inner loop, enabling surrogate-driven generation while retaining information from true-oracle feedback; and SMILES-augmented surrogate updates, which exploits the string-based featurizer to extract additional signal from each oracle call.

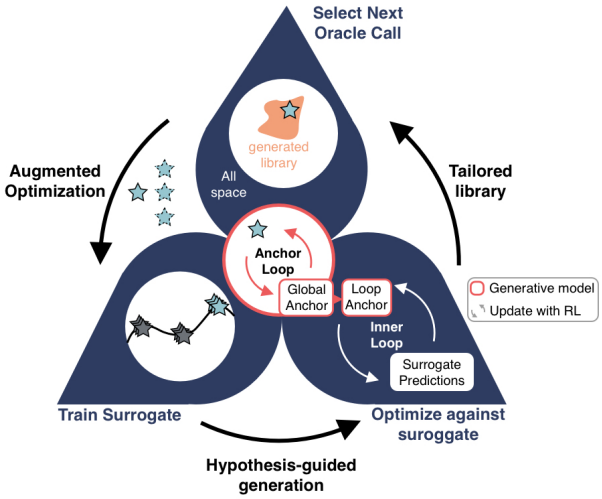

Figure 1. Overview of the Sample Efficient Generative Optimization (SEGO) framework. SEGO unifies generative modeling with Bayesian optimization for sample-efficient search over open-ended molecular space. At each iteration, a surrogate-guided generative model produces a targeted library of molecules in promising regions of chemical space (Guo et al., 2026). This library is featurized through a deep GP (Ranković et al., 2025), and the next most informative experiment is selected via an acquisition function. SMILES augmentation during surrogate training and an anchoring mechanism for surrogate-guided generation further improve sample efficiency.

3.1. Molecular Generation Engine.

As our generative engine, we take Saturn (Guo et al., 2026), a Mamba-based (Gu & Dao, 2024) autoregressive SMILES generator trained with reinforcement learning. A SMILES

string $x = (a_1, ..., a_T)$ is generated as a Markov process,

$$P_\theta(x) = \prod_{t=1}^{T} \pi_{\theta_{agent}}(a_t \mid s_t) \quad (1)$$

where a_t is the token selected at step t, s_t is the token sequence so far, and $\pi_{\theta_{agent}}$ is the Mamba backbone. The aim in RL is to maximize the expected reward (equation **??**). Augmented likelihood (equation 2) is defined, where the prior is a pretrained model with frozen weights, R is a reward function, and σ is a scalar modulating the reward's effect:

$$\log \pi_{\theta_{\text{Augmented}}} = \log \pi_{\theta_{\text{prior}}} + \sigma R(x) \quad (2)$$

Maximizing equation **??** is equivalent (up to a factor) to minimizing the squared difference between the Augmented and Agent likelihoods (equation 3).

$$L(\theta) = \frac{1}{|B|}\left[\sum_{a \in A^*} \left(\log \pi_{\theta_{\text{Augmented}}} - \log \pi_{\theta_{\text{agent}}}\right)^2\right] \quad (3)$$

where B is a batch of SMILES sampled from the agent and A^* denotes the action sequences across all timesteps in the batch. Sample efficiency in Saturn comes from Augmented Memory: a replay buffer of the top-100 rewarded SMILES is maintained, randomized into equivalent SMILES, and used for $N_{augmentation}$ additional update rounds per iteration. Mode collapse is mitigated by purging the buffer of scaffolds with low-diversity samples and assigning zero reward to repeated generations beyond a threshold.

3.2. Bayesian Optimization Engine.

As our BO engine, we take Gollum, a flexible framework based on SMILES that jointly trains LLM embeddings and GP hyperparameters (Ranković et al., 2025). A learned feature transformation is composed with the base kernel:

$$k_{\theta,\phi}(\mathbf{x}, \mathbf{x}') = k_\theta(g_\phi(\mathbf{x}), g_\phi(\mathbf{x}')) \quad (4)$$

where g_ϕ is a parameterized feature extractor with parameters ϕ, trained jointly with the GP hyperparameters through the marginal likelihood. Though multiple ways to constructing g_ϕ were proposed in the original paper, we take the projection layer variant: $g_\phi(\mathbf{x}) = \mathbf{P}\text{LLM}(t)$, where $\mathbf{P} \in \mathbb{R}^{m \times d}$ is a learned linear projection followed by ELU activation. The GP hyperparameters $\theta = (\ell, \sigma^2, \sigma_n^2, c)$ and feature parameters θ are jointly optimized by maximizing the log marginal likelihood:

$$\begin{aligned}\mathcal{L}(\theta, \phi) &= \log p(\mathbf{y} \mid \mathbf{X}, \theta, \phi) \\ &= -\frac{1}{2}\left(\mathbf{y}^\top \mathbf{K}_{\theta,\phi}^{-1}\mathbf{y} + \log|\mathbf{K}_{\theta,\phi}| + n \log 2\pi\right) \quad (5)\end{aligned}$$

Gollum is particularly well suited as the BO engine in SEGO because it operates directly on SMILES strings, bypassing the need for hand-crafted molecular descriptors. Through joint optimization of the LLM and GP via the marginal likelihood, Gollum learns to smooth the representation space so that the GP can generalize effectively from very few observations.

3.3. SEGO operation

At iteration i, SEGO maintains the anchor agent $\pi_{\theta_{\text{agent}}}^{(i)}$ and a GP $\mathcal{GP}^{(i)}$ fit to the oracle-evaluated set $\mathcal{D}^{(i)} = \{(x_j, y_j)\}_{j \le i}$. The loop is initialized by sampling L molecules from $\pi_{\theta_{\text{prior}}}$, scoring them under the acquisition function, and selecting n_{init} for oracle evaluation; these seed $\mathcal{D}^{(0)}$ and provide the first updates to $\pi_{\theta_{\text{agent}}}$ and $\mathcal{GP}$. Each iteration:

1. Initializes $\pi_{\theta_{\text{inner}}}$ from $\pi_{\theta_{\text{agent}}}^{(i)}$.
2. Trains $\pi_{\theta_{\text{inner}}}$ for K_{inner} Saturn updates against $R_{\text{surr}}(x) = \mu^{(i)}(x)$, the GP posterior mean.
3. Samples a candidate library $\mathcal{C}^{(i)}$ of size L from $\pi_{\theta_{\text{inner}}}$, then *discards* $\pi_{\theta_{\text{inner}}}$.
4. Selects $x_i^\star = \arg\max_{x \in \mathcal{C}^{(i)}} \alpha(x)$ under acquisition function α (we use expected improvement).
5. Queries the oracle to obtain $y_i^\star = f_{\text{oracle}}(x_i^\star)$.
6. Constructs an augmented set $\mathcal{A}_i = \{(x_{i,k}^\star, y_i^\star)\}_{k=1}^{N_{\text{aug}}}$ from N_{aug} randomized-SMILES variants of $x_i^\star$, updates $\mathcal{D}^{(i+1)} \leftarrow \mathcal{D}^{(i)} \cup \mathcal{A}_i$, refits (θ, ϕ) via Eq. (5), and updates $\pi_{\theta_{\text{agent}}}$ by one round of Saturn updates (Eq. (3)) with reward $y_i^\star$.

4. Experiments

4.1. SEGO architecture ablation experiments

We explore the impact of key hyperparameters in SEGO: the role of the inner loop in sample efficiency, the anchor's role in training stability, and augmented optimization.

Experimental Setup. Following Guo & Schwaller (2024b), we optimize molecules according to a multi-property optimization (MPO) objective: molecular weight (MW) <350 Da, number of rings ≥ 2, and maximize the topological polar surface area (tPSA). To satisfy this objective, molecules must have rings saturated with heteroatoms. Such saturated molecules are dissimilar from training data, so this experiment also evaluates out-of-distribution optimization. We use an extremely stringent oracle budget of 100 calls, 10% of the already stringent budget employed in (Guo et al., 2026), which better reflects realistic budgets for experimental campaigns. All experiments were run across 10 seeds (0-9 inclusive) using a prior pretrained on the ChEMBL 33 dataset (Gaulton et al., 2012).

Table 1. Sample efficiency and diversity metrics for inner loop initialization strategies in SEGO. OB (oracle burden) is the number of calls required to generate N unique molecules, and yield is the number of unique molecules generated. IntDiv is internal diversity, Scaffolds is the number of Bemis-Murcko scaffolds, and #Circles measures chemical space coverage. Metrics are computed at a threshold of 0.7 with a budget of 100 oracle calls. The mean and standard deviation across 10 seeds (0-9 inclusive) are reported. Numbers in parentheses represent number of runs to find any hits out of ten.

INITIALIZATION	YIELD (↑)	OB1(↓)	OB5(↓)	INTDIV (↑)	SCAFFOLDS (↑)	CIRCLES (↑)
PRIOR	0±0	40±50 (3)	-	-	0±0	0±0
CONTINUOUS	47±19	40±27	38±17 (9)	0.774±0.027	47±19	7±3
ANCHOR	57±22	23±6 (9)	30±7 (9)	0.568±0.081	52±21	2±2

Table 2. Sample efficiency and diversity metrics for inner loop iterations. The first entry 0 match augmentation sets the anchor augmentation rounds to 210 to mirror the number of updates that would occur in the inner loop given augmented memory. OB (oracle burden) is the number of calls required to generate N unique molecules, and yield is the number of unique molecules generated. IntDiv is internal diversity, Scaffolds is the number of Bemis-Murcko scaffolds, and #Circles measures chemical space coverage. Metrics are computed at a threshold of 0.7 with a budget of 100 oracle calls. The mean and standard deviation across 10 seeds (0-9 inclusive) are reported. Numbers in parentheses represent number of runs to find any hits out of ten.

INNER LOOP ITERATIONS	YIELD (↑)	OB1(↓)	OB5(↓)	INTDIV (↑)	SCAFFOLDS (↑)	CIRCLES (↑)
0 MATCH AUGMENTATION	24±31	34±25 (4)	42±25 (3)	0.423±0.052	12±17	1±1
0	51±24	40±29	39±21 (9)	0.519±0.084	37±23	1±1
5	50±25	33±18 (9)	39±18 (9)	0.580±0.087	43±22	2±2
10	60±14	31±15	37±16	0.582±0.092	54±13	3±1
20	66±9	23±9	30±9	0.608±0.086	60±9	4±2

Metrics. As our sample efficiency metrics, we take **Yield**, the number of *unique* molecules generated above a reward threshold and **Oracle Burden** (OB), the number of oracle calls required to generate *N* molecules above a reward thresholds. As our thresholds, we take 0.7 as molecules start to possess saturated heteroatom rings. We also compute diversity metrics, as there is a tradeoff in most generative methods between yield and diversity. We compute **IntDiv** the internal diversity (Polykovskiy et al., 2020), **Scaffolds** the number of Bemis-Murcko scaffolds (Bemis & Murcko, 1996), and **#Circles** (Xie et al., 2023) which measures chemical space coverage.

Inner loop anchor strategies. We explore three strategies for instantiating the inner loop, in which the RL agent receives reward from the GP predicted mean. The default strategy initializes each inner loop from an anchor, an RL agent previously trained on true oracle rewards, giving the inner loop a warm start in a promising region of chemical space. To test whether this warm start is necessary, we ablate with two alternatives: a prior strategy, which re-initializes from the pretrained prior at each iteration, testing whether the inner loop can learn to generate high-scoring molecules without any oracle-informed starting point; and a continuous strategy, which retains the inner loop agent across iterations, testing whether it is sufficiently flexible to track the changing GP landscape throughout the campaign.

The prior initialization strategy fails to produce hits in most cases (Table 1). This is unsurprising: at early iterations the GP has been trained on as few as 10 molecules, so its reward landscape is poorly calibrated across chemical space, providing too faint a signal for the agent to learn from scratch. The continuous strategy yields competitive hit counts, but at the cost of reliability. Because the agent is never reset, it tends to commit to a narrow region of chemical space; if that region contains hits, oracle burden is low, but if not, the agent struggles to transition elsewhere. This is reflected in the high variance of OB1 across seeds. The continuous strategy does produce greater diversity by all metrics, likely because the persistent agent drifts through different regions of chemical space across iterations as the GP landscape evolves, whereas the anchor resets to the same starting point each iteration, biasing exploration toward a narrower neighborhood. In the context of prospective discovery, however, the additional diversity likely does not outweigh the unreliable oracle burden, and we adopt the anchor strategy as the default.

Surrogate-guided inner loop. Given the strong performance of the anchor strategy, we next ablate the importance of the inner loop itself by testing variants with different numbers of inner loop rounds, including a variant with no inner loop at all. To isolate the effect of surrogate-informed training from the additional augmented memory updates that the inner loop provides, we include a further control: a variant with zero inner loops but with *N* anchor augmentation rounds set to 210, matching the total number of updates the 20-step inner loop would receive (20 inner loop steps $\times$ 10 inner loop augmentations, plus 10 outer loop augmentations). Sample efficiency improves and variance across seeds decreases as the number of inner loop iterations in-

Table 3. Sample efficiency and diversity metrics for different augmentation factors of SEGO. OB (oracle burden) is the number of calls required to generate N unique molecules, and yield is the number of unique molecules generated. IntDiv is internal diversity, Scaffolds is the number of Bemis-Murcko scaffolds, and #Circles measures chemical space coverage. Metrics are computed at a threshold of 0.7 with a budget of 100 oracle calls. The mean and standard deviation across 10 seeds (0-9 inclusive) are reported. Bolded values are statistically significant compared to their counterparts at augmentation factor 1 with a Welch's two-sided t-test at a significance level of 0.01. Numbers in parentheses in the yield column represent number of runs to find any hits out of ten.

AUGMENTATION FACTOR	YIELD (↑)	OB1(↓)	OB5(↓)	INTDIV (↑)	SCAFFOLDS (↑)	#CIRCLES (↑)
1	33±16	45±22	61±17	0.619±0.063	32±15	3±1
5	**65±6**	25±7	**32±6**	0.601±0.069	**62±8**	3±2
10	**67±6**	22±7	**29±6**	0.554±0.087	**62±6**	2±1
20	55±21	25±10 (9)	**31±9** (9)	0.630±0.070	50±20	4±3

creases, and diversity follows suit, reflecting the stronger optimization (Table 2). The augmentation-matched control, despite receiving the same number of gradient updates from the replay buffer, suffers in yield, oracle burden, and the number of seeds that successfully identify a hit, confirming that the benefit of the inner loop stems from training on GP-predicted rewards rather than from additional augmented memory updates alone. Though the difference between 0 and 20 inner loops does not reach statistical significance for most metrics, the consistent trends across all metrics suggest that the inner loop improves both the reliability and efficiency of SEGO.

Augmented Optimization. Inspired by Augmented Memory, in which a generative model is updated according to N augmentation rounds of randomized, high-reward SMILES, we develop augmented optimization, in which library SMILES are augmented N-fold and the GP is trained on all N representations for each molecule. Because the Gollum framework learns its latent space through implicit contrastive learning — reorganizing so that similar molecules have more similar representations — we hypothesize that providing many representations of the same molecule will encourage the GP to align them, improving predictive accuracy.

Augmented optimization improves both oracle burden and yield by a statistically significant margin (Table 3), and further improves diversity by all metrics. The effect saturates at 5-fold augmentation; no statistically significant difference was observed between 5-, 10-, and 20-fold augmentation. To investigate the mechanism behind this improvement, we examined the impact of augmentation on both representational alignment and predictive accuracy. At each iteration we sampled 500 molecules from the prior and from the anchor and recorded the GP mean prediction across 10 augmentations of each molecule. SMILES augmentation narrows the range of predictions for different representations of the same molecule, and this alignment improves with additional augmentation rounds (Figure A1). Crucially, the effect is observed for the prior as well as the anchor, indicating that augmentation improves global representational alignment rather than only aligning the narrow region of chemical space being explored by the oracle. This corresponds to gains in predictive accuracy — measured by the correlation between mean predicted values and true oracle values — particularly early on, when the GP has very little training data. (Figure A1).

Isomer Collapse. The previous experiments showcase the powerful optimization behavior of SEGO, though as is typical of strong optimizers, this comes at the cost of diversity. In practice, low oracle burden partially mitigates this concern: when hits are found within tens of oracle calls, practitioners typically pool results across multiple short campaigns rather than relying on a single run to cover chemical space. It is still of note, however, the Bemis-Murcko scaffold metric can be misleading for this oracle, which rewards saturated rings: dense saturation means that molecules differing only in heteroatom placement or functionalization are assigned distinct scaffolds, inflating apparent diversity. To capture this, we introduce the metric Isomers, defined as the number of unique molecular formulas among hits above a threshold of 0.7. Despite generating 67 (±6) unique molecules at this threshold, SEGO produces only 17 (±8) unique isomers, revealing a form of mode collapse in which the agent repeatedly generates constitutional isomers of the same formula.

To mitigate this, we introduce a diversity filter in the inner loop that tracks molecular formulas rather than Bemis-Murcko scaffolds. The anchor and inner loop maintain separate filters, and the inner loop filter is reset at each outer loop iteration. We set a stringent isomer threshold of 2, meaning the inner loop penalizes a molecular formula after it has been seen twice and the outer loop penalizes formulas that have been selected twice. The filtered version produces lower overall yield but generates significantly more unique isomers, alleviating the repeated generation of molecules that differ only by a stereocenter or functional group placement (Table A1). While such isomer exploration may itself be valuable in some campaigns that depend on the spatial arrangement of functionalization, the diversity filter provides explicit control over this tradeoff.

Table 4. Sample efficiency and diversity metrics comparing SEGO to its engines, Saturn (generative backbone) and Gollum (BO). OB (oracle burden) is the number of calls required to generate N unique molecules, and yield is the number of unique molecules generated. IntDiv is internal diversity, Scaffolds is the number of Bemis-Murcko scaffolds, and #Circles measures chemical space coverage. Metrics are computed at a threshold of 0.7 with a budget of 100 oracle calls. The mean and standard deviation across 10 seeds (0-9 inclusive) are reported. Numbers in parentheses in the yield column represent number of runs to find any hits out of ten.

METHOD	YIELD (↑)	OB1(↓)	OB5(↓)	INTDIV (↑)	SCAFFOLDS (↑)	CIRCLES(↑)
CHEMBL 33	0±0	-	-	-	-	-
SATURN	-	-	-	-	-	-
GOLLUM-1X	0±1	18±1 (2)	-	-	0±1	0±1
GOLLUM-10X	0±0	60±29 (2)	-	-	0±0	0±0
SEGO	59±11	29±14	35±12	0.601±0.069	56±11	37±7

Hybrid versus individual approaches. SEGO is a hybrid architecture that enables sample-efficient search over open-ended chemical space. We compare SEGO to the individual generative and BO engines it is built from to showcase the benefits of the hybrid approach. To benchmark Gollum, we take a random sample of 500 molecules from ChEMBL 33 (Gaulton et al., 2012), the same database used to train the prior in Saturn and in SEGO, to mimic a real-world prospective campaign in which a desirable library is not known. Given the benefits of augmented optimization observed earlier, we also test Gollum with 10-fold SMILES augmentation (Gollum-10x). For SEGO, we set the number of inner loop iterations to 20, augmented optimization to 10, and use the anchored framework as well as the Isomer diversity filter.

Saturn fails to produce any hits at this strict oracle budget. With the recommended batch size of 16 and augmentation rounds of 10, it undergoes only 6 rounds of RL updates, which is insufficient to shift the distribution toward hit molecules. Gollum, on the other hand, fails because the randomly sampled library simply does not contain hits. Augmentation in pure Gollum did not improve metrics as there were too few hits to calculate statistics.

4.2. PMO benchmark

Experimental Setup. We evaluate SEGO on the Practical Molecular Optimization (PMO) benchmark, a suite of 23 oracles spanning model-based scoring (e.g., DRD2, JNK3, GSK3β), similarity-based objectives (e.g., Albuterol, Mestranol, Celecoxib rediscovery), and physicochemical and multi-property composites (e.g., QED, isomer matching, scaffold hopping) (Gao et al., 2022). PMO reports the area under the curve (AUC) of the top-10 average performance as a function of cumulative oracle queries, favoring methods that find high values in fewer calls. The PMO takes a relaxed budget of 10,000. For increased relevance to expensive oracles, following (Nguyen & Grover, 2025), we instead report results for baselines on the PMO when restricted to an oracle budget of 1,000 (PMO-1K). For SEGO, we restrict oracle budget to just 100 oracle calls, after which the top-1 and top-10 scores are held fixed for computing metrics at the 1K horizon (Krüger et al., 2026) to showcase sample efficiency. We report both top-1 AUC (consistent with 1% of oracle calls) and top-10 AUC, aggregated across 3 seeds (0–2 inclusive). We compare against published PMO leaderboard methods including REINVENT (Loeffler et al., 2024), Augmented Memory (Guo & Schwaller, 2024a), Genetic GFN (Kim et al., 2024), MOLLEO (Wang et al., 2025), Graph GA (Jensen, 2019), and GP BO (Tripp & Hernández-Lobato, 2024).

Table 5. Performance of SEGO and baselines on 23 tasks in the PMO-1K. A higher score is better. We report the mean and standard deviation of 3 seeds. Baseline results are taken from prior work (Nguyen & Grover, 2025) . SEGO only proposes molecules during the first 100 oracle calls, after which the top-1 or top-10 score is held fixed, while baselines continue optimizing for the full 1,000 oracle calls.

METHOD	SUM OF AUC (↑)
REINVENT	10.68
AUGMENTED MEMORY	10.81
GRAPH GA	10.90
GP BO	11.27
GENETIC GFN	11.56
MOLLEO	11.65
LICO	11.71
SEGO-TOP10 (OURS)	11.16
SEGO-TOP1 (OURS)	11.86

At the top-1 threshold, which mirrors the baselines by evaluating the top 1% of the oracle budget, SEGO achieves state-of-the-art performance **despite consuming only a tenth of the oracle budget available to other methods** (Table 5). SEGO attains competitive performance at the top-10 threshold which particularly challenging because top-10 represents 10% of the total available budget. We note that the narrow gap between top-1 and top-10 performance could reflect diversity collapse, where the model generates many near-identical high-scoring candidates; however, diversity metrics are not reported for the baselines, precluding a direct comparison (Nguyen & Grover, 2025). Extended results for individual oracles are reported in the appendix (A2). Together, these results demonstrate SEGO's strong opti-

mization capability under tight budget constraints.

The most closely related method in this benchmark is GP BO (Tripp & Hernández-Lobato, 2024), which uses Bayesian optimization with count-based molecular fingerprints to select candidates from a library generated by a genetic algorithm. SEGO outperforms GP BO at a fraction of the oracle budget, highlighting the strength of its underlying generative and optimization components. LICO (Nguyen & Grover, 2025) follows a similar framework but replaces the GP surrogate with an LLM. SEGO surpasses LICO with greater sample efficiency, though LICO's strong results suggest that substituting SEGO's GP with an LLM is a promising direction for future work. Another language-model approach is MOLLEO (Wang et al., 2025), which leverages an LLM to perform mutation and crossover operations within an evolutionary algorithm. While MOLLEO achieves competitive performance, it relies on detailed textual task descriptions provided to the LLM, raising concerns about data leakage and limiting its applicability to prospective campaigns where such descriptions may not be available.

5. Discussion

We introduce SEGO, a hybrid generative optimizer for sample-efficient search in open-ended chemical space. Unlike pure generative methods that rely on slow updates to shift the distribution, and conventional BO methods that operate on fixed libraries, SEGO leverages the principled exploration of BO across dynamically generated libraries. This design yields gains in hit-finding and sample efficiency compared to the individual parent methods on a toy oracle. SEGO also achieves excellent performance on the PMO benchmark, reaching state-of-the-art top 1% performance in only 100 oracle calls—one tenth of the oracle budget of PMO-1k.

Our ablations show that our design decisions—namely, guiding an inner loop based on surrogate predictions and anchoring that inner loop—contribute to the sample efficiency of SEGO. Further, data augmentation, known to be beneficial for sample efficiency in generative models, can improve the predictive performance of BO as well and improve representational alignment between diverse SMILES.

However, excellent performance on the PMO benchmark may not reflect optimization behavior in the complex reward landscapes found in real-world campaigns, where competing objectives like synthetic accessibility, solubility, and activity must be navigated simultaneously. Further work should assess SEGO on challenging oracles including physics-based simulations relevant to drug-discovery and catalysis as well as stringent synthesis constraints before considering deployment to wet-lab prospective campaigns (Lee et al., 2024). Further work should also investigate the mechanism behind repeat isomer generation in SEGO. We alleviate repeat isomer generation through a diversity filter (Blaschke et al., 2020), but further studies on isomer representation within the augmented GP as well as on isomer collapse in the inner loop could help target the issue at the source. Finally, practical high-throughput campaigns require multiobjective optimization over Pareto fronts and batched acquisition (Sin et al., 2025) which could be introduced into SEGO via alternative acquisition functions (Muthyala et al., 2025). Despite these limitations, the sample efficiency and hit-finding ability demonstrated here are a promising step towards generative molecular design driven by direct experimental feedback.

Software and Data

If a paper is accepted, we will publish software with the camera-ready version.

Impact Statement

As AI-driven methods become increasingly integrated into chemical discovery, it is important to acknowledge the dual-use risks inherent in molecular generation, which become more relevant as engines become more powerful. The risks of SEGO are in line with existing performant molecular design engines. Such risks can be mitigated through controlled access to oracles and established safeguards at the institutional level.

References

Agarwal, G., Doan, H. A., Robertson, L. A., Zhang, L., and Assary, R. S. Discovery of energy storage molecular materials using quantum chemistry-guided multiobjective bayesian optimization. *Chemistry of Materials*, 33(20): 8133–8144, 2021.

Arús-Pous, J., Johansson, S. V., Prykhodko, O., Bjerrum, E. J., Tyrchan, C., Reymond, J.-L., Chen, H., and Engkvist, O. Randomized smiles strings improve the quality of molecular generative models. *Journal of cheminformatics*, 11:1–13, 2019.

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

Bengio, E., Jain, M., Korablyov, M., Precup, D., and Bengio, Y. Flow network based generative models for non-iterative diverse candidate generation. *Advances in Neural Information Processing Systems*, 34:27381–27394, 2021.

Blaschke, T., Engkvist, O., Bajorath, J., and Chen, H. Memory-assisted reinforcement learning for diverse

molecular de novo design. *Journal of cheminformatics*, 12(1):68, 2020.

Brown, N., Fiscato, M., Segler, M. H., and Vaucher, A. C. Guacamol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling*, 59(3):1096–1108, 2019.

Comlek, Y., Pham, T. D., Snurr, R. Q., and Chen, W. Rapid design of top-performing metal-organic frameworks with qualitative representations of building blocks. *npj Computational Materials*, 9(1):170, 2023.

Dodds, M., Guo, J., Löhr, T., Tibo, A., Engkvist, O., and Janet, J. P. Sample efficient reinforcement learning with active learning for molecular design. *Chemical Science*, 15(11):4146–4160, 2024.

Feng, T., Xu, P., Fu, T., Laghuvarapu, S., and Sun, J. Molecular de novo design through transformer-based reinforcement learning. *arXiv preprint arXiv:2310.05365*, 2023.

Filella-Merce, I., Molina, A., Díaz, L., Orzechowski, M., Berchiche, Y. A., Zhu, Y. M., Vilalta-Mor, J., Malo, L., Yekkirala, A. S., Ray, S., et al. Optimizing drug design by merging generative ai with a physics-based active learning framework. *Communications Chemistry*, 8(1): 238, 2025.

Frazier, P. I. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

Gao, W., Fu, T., Sun, J., and Coley, C. Sample efficiency matters: a benchmark for practical molecular optimization. *Advances in neural information processing systems*, 35:21342–21357, 2022.

Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., Light, Y., McGlinchey, S., Michalovich, D., Al-Lazikani, B., et al. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(D1):D1100–D1107, 2012.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

Graff, D. E., Shakhnovich, E. I., and Coley, C. W. Accelerating high-throughput virtual screening through molecular pool-based active learning. *Chemical science*, 12(22): 7866–7881, 2021.

Griffiths, R.-R., Greenfield, J. L., Thawani, A. R., Jamasb, A. R., Moss, H. B., Bourached, A., Jones, P., McCorkindale, W., Aldrick, A. A., Fuchter, M. J., et al. Data-driven discovery of molecular photoswitches with multioutput gaussian processes. *Chemical Science*, 13(45):13541–13551, 2022.

Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. In *1st Conference On Language Modeling*, 2024.

Guo, J. and Schwaller, P. Augmented memory: Sample-efficient generative molecular design with reinforcement learning. *JACS Au*, 2024a.

Guo, J. and Schwaller, P. Augmented memory: Sample-efficient generative molecular design with reinforcement learning. *Jacs Au*, 2024b.

Guo, J., Chen, J., Gx-Chen, A., and Schwaller, P. Sample-efficient generative molecular design using memory manipulation. *Nature Machine Intelligence*, pp. 1–12, 2026.

Janet, J. P., Ramesh, S., Duan, C., and Kulik, H. J. Accurate multiobjective design in a space of millions of transition metal complexes with neural-network-driven efficient global optimization. *ACS central science*, 6(4): 513–524, 2020.

Jensen, J. H. 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.

Jiang, Y., Zhang, G., You, J., Zhang, H., Yao, R., Xie, H., Zhang, L., Xia, Z., Dai, M., Wu, Y., et al. Pocketflow is a data-and-knowledge-driven structure-based molecular generative model. *Nature Machine Intelligence*, 6(3): 326–337, 2024.

Kim, H., Kim, M., Choi, S., and Park, J. Genetic-guided gflownets for sample efficient molecular optimization. *Advances in Neural Information Processing Systems*, 37: 42618–42648, 2024.

Kristiadi, A., Strieth-Kalthoff, F., Skreta, M., Poupart, P., Aspuru-Guzik, A., and Pleiss, G. A sober look at llms for material discovery: Are they actually good for bayesian optimization over molecules? *arXiv preprint arXiv:2402.05015*, 2024.

Krüger, F. P., Hunklinger, A., Wolny, A., Adler, T. J., Tetko, I., and Villalba, S. D. Seismo: Increasing sample efficiency in molecular optimization with a trajectory-aware llm agent. *arXiv preprint arXiv:2602.00663*, 2026.

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

Lee, S., Kreis, K., Veccham, S. P., Liu, M., Reidenbach, D., Paliwal, S., Vahdat, A., and Nie, W. Molecule generation with fragment retrieval augmentation. *Advances in Neural Information Processing Systems*, 37:132463–132490, 2024.

Lehman, J., Clune, J., Misevic, D., Adami, C., Altenberg, L., Beaulieu, J., Bentley, P. J., Bernard, S., Beslon, G., Bryson, D. M., et al. The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities. *Artificial life*, 26(2):274–306, 2020.

Loeffler, H. H., He, J., Tibo, A., Janet, J. P., Voronov, A., Mervin, L. H., and Engkvist, O. Reinvent 4: Modern ai–driven generative molecule design. *Journal of Cheminformatics*, 16(1):20, 2024.

Mazuz, E., Shtar, G., Shapira, B., and Rokach, L. Molecule generation using transformers and policy gradient reinforcement learning. *Scientific Reports*, 13(1):8799, 2023.

Moss, H., Leslie, D., Beck, D., Gonzalez, J., and Rayson, P. Boss: Bayesian optimization over string spaces. *Advances in neural information processing systems*, 33: 15476–15486, 2020.

Muthyala, M. R., Sorourifar, F., Tan, T., Peng, Y., and Paulson, J. A. Generative multiobjective bayesian optimization with scalable batch evaluations for sample-efficient de novo molecular design. *Industrial & Engineering Chemistry Research*, 65(1):628–642, 2025.

Nguyen, T. and Grover, A. Lico: Large language models for in-context molecular optimization. In *International Conference on Learning Representations*, volume 2025, pp. 53576–53597, 2025.

Oh, C., Tomczak, J., Gavves, E., and Welling, M. Combinatorial bayesian optimization using the graph cartesian product. *Advances in Neural Information Processing Systems*, 32, 2019.

Olivecrona, M., Blaschke, T., Engkvist, O., and Chen, H. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9:1–14, 2017.

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

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21 (140):1–67, 2020.

Ranković, B. and Schwaller, P. Bochemian: Large language model embeddings for bayesian optimization of chemical reactions. In *NeurIPS 2023 Workshop on Adaptive Experimental Design and Active Learning in the Real World*, 2023.

Ranković, B., Griffiths, R.-R., and Schwaller, P. Large language models as uncertainty-calibrated optimizers for experimental discovery. *arXiv preprint arXiv:2504.06265*, 2025.

Renz, P., Van Rompaey, D., Wegner, J. K., Hochreiter, S., and Klambauer, G. On failure modes in molecule generation and optimization. *Drug Discovery Today: Technologies*, 32:55–63, 2019.

Sanchez-Lengeling, B. and Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, 2018.

Schneuing, A., Du, Y., Harris, C., Didi, K., Jamasb, A., Igashov, I., Du, W., Gomes, C., Welling, M., Blundell, T., et al. Flexible structure-based design of small molecules with equivariant diffusion models. In *PROTEIN SCIENCE*, volume 32. WILEY 111 RIVER ST, HOBOKEN 07030-5774, NJ USA, 2023.

Seeger, M. Gaussian processes for machine learning. *International journal of neural systems*, 14(02):69–106, 2004.

Seumer, J., Kirschner Solberg Hansen, J., Brøndsted Nielsen, M., and Jensen, J. H. Computational evolution of new catalysts for the morita–baylis–hillman reaction. *Angewandte Chemie International Edition*, 62 (18):e202218565, 2023.

Sin, J. W., Chau, S. L., Burwood, R. P., Püntener, K., Bigler, R., and Schwaller, P. Highly parallel optimisation of chemical reactions through automation and machine intelligence. *Nature Communications*, 16(1):6464, 2025.

Strieth-Kalthoff, F., Hao, H., Rathore, V., Derasp, J., Gaudin, T., Angello, N. H., Seifrid, M., Trushina, E., Guy, M., Liu, J., et al. Delocalized, asynchronous, closed-loop discovery of organic laser emitters. *Science*, 384(6697): eadk9227, 2024.

Swersky, K., Rubanova, Y., Dohan, D., and Murphy, K. Amortized bayesian optimization over discrete spaces. In *Conference on Uncertainty in Artificial Intelligence*, pp. 769–778. PMLR, 2020.

Tripp, A. and Hernández-Lobato, J. M. Diagnosing and fixing common problems in bayesian optimization for molecule design. *arXiv preprint arXiv:2406.07709*, 2024.

Wang, H., Skreta, M., Ser, C.-T., Gao, W., Kong, L., Strieth-Kalthoff, F., Duan, C., Zhuang, Y., Yu, Y., Zhu, Y., et al. Efficient evolutionary search over chemical space with large language models. In *International Conference on Learning Representations*, volume 2025, pp. 51694–51727, 2025.

Weininger, D. 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.

Xie, Y., Xu, Z., Ma, J., and Mei, Q. How much space has been explored? measuring the chemical space covered by databases and machine-generated molecules. In *Proc. 11th International Conference on Learning Representations*, 2023.

A. Appendix.

A.1. Saturn Hyperparameters.

Saturn was used with default hyperparameters as recommended in the paper (Guo et al., 2026). The inner loop used 10 augmentation rounds and a batch size of 64, selected to be a good balance between sample efficiency and stability in the surrogate-guided approach which does not consume oracle calls. The outer loop used 10 augmentation rounds based on a batch size of 1, selected by the BO loop.

A.2. Gollum Hyperparameters.

Gollum was used with default hyperparameters as recommended in the paper (Ranković et al., 2025). We used the T5 language model (Raffel et al., 2020) as our base featurizer in the PLLM setting, using a gaussian process (Seeger, 2004) as our surrogate model and a Matérn-5/2 kernel. We use the expected improvement acquisition function. For other implementation details, refer to Ranković et al. (2025).

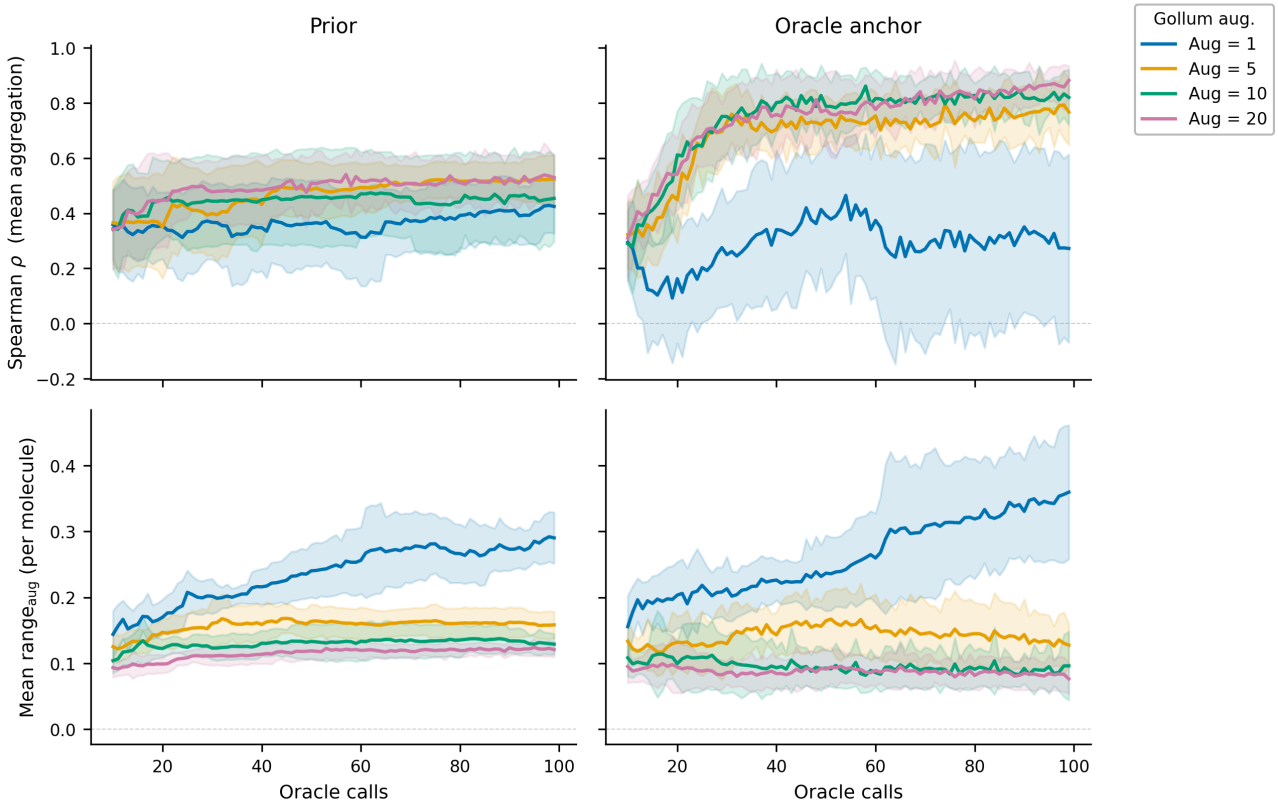

Figure A1. Effect of SMILES augmentation on GP surrogate quality (predictive accuracy and representational alignment) for molecules sampled from the prior and the oracle anchor across oracle calls. **Top:** Spearman rank correlation () between the GP's predicted rewards (mean over 10 augmented SMILES representations) and true oracle rewards, averaged over 10 seeds (shading: ±1 std). **Bottom:** Mean within-molecule prediction range (max min GP prediction across 10 augmentations per molecule), measuring consistency of the GP featurizer. Without augmentation during training (aug = 1), the GP achieves poor rank correlation on oracle-anchor molecules and produces highly inconsistent predictions across SMILES representations of the same molecule. Increasing to aug 5 substantially reduces within-molecule variance and improves rank correlation, particularly on the oracle-anchor set. Further increases to aug = 10 or 20 yield diminishing returns. Notably, augmentation improves representational alignment even for molecules sampled from the prior, diverse chemical space far from the oracle-explored region, though the gains in predictive accuracy there are more modest.

Table A1. Sample efficiency and diversity metrics for SEGO with and without the isomer filter. OB (oracle burden) is the number of calls required to generate N unique molecules, and yield is the number of unique molecules generated. IntDiv is internal diversity, Scaffolds is the number of Bemis-Murcko scaffolds, and #Circles measures chemical space coverage. Metrics are computed at a threshold of 0.7 with an budget of 100 oracle calls. The mean and standard deviation across 10 seeds (0-9 inclusive) are reported. Bolded values are statistically significant compared to their counterparts at augmentation factor 1 with a Welch's two-sided t-test at a significance level of 0.01. Numbers in parentheses in the yield column represent number of runs to find any hits out of ten.

Diversity Filter	Yield (↑)	OB1(↓)	OB5(↓)	IntDiv (↑)	Scaffolds (↑)	Isomers (↑)	Circles (↑)
Scaffold	67±16	22±7	29±6	0.554±0.087	62±6	17±8	2±1
Scaffold + Isomer	59±11	29±14	35±12	0.601±0.069	56±11	37±7	5±4

Table A2. Performance of SEGO and baselines on 23 tasks in the PMO-1k. A higher score is better. We report the mean and standard deviation of 3 seeds. SEGO only proposes molecules during the first 100 oracle calls, after which the top-1 or top-10 score is held fixed. We report the top-1 and top-10 AUC score across 100 oracle calls, 1k oracle calls, and 10k oracle calls.

Budget	100		1k		10k	
Oracle	top-1	top-10	top-1	top-10	top-1	top-10
albuterol_similarity	0.492±0.114	0.437±0.076	0.568±0.139	0.537±0.113	0.576±0.142	0.547±0.117
amlodipine_mpo	0.475±0.027	0.429±0.028	0.524±0.040	0.509±0.036	0.529±0.041	0.517±0.037
celecoxib_rediscovery	0.350±0.032	0.312±0.026	0.405±0.051	0.385±0.045	0.410±0.053	0.392±0.047
deco_hop	0.592±0.012	0.582±0.013	0.605±0.013	0.602±0.014	0.606±0.013	0.604±0.014
drd2	0.400±0.383	0.315±0.376	0.718±0.317	0.605±0.413	0.749±0.315	0.634±0.423
fexofenadine_mpo	0.647±0.024	0.597±0.026	0.717±0.017	0.703±0.019	0.724±0.017	0.713±0.019
gsk3b	0.390±0.065	0.309±0.086	0.603±0.047	0.577±0.029	0.624±0.056	0.604±0.039
isomers_c7h8n2o2	0.713±0.077	0.586±0.089	0.956±0.020	0.896±0.017	0.981±0.015	0.927±0.010
isomers_c9h10n2o2pf2cl	0.463±0.122	0.375±0.139	0.699±0.056	0.624±0.107	0.723±0.052	0.649±0.106
jnk3	0.065±0.034	0.032±0.017	0.108±0.017	0.054±0.005	0.110±0.017	0.055±0.005
median1	0.245±0.026	0.205±0.024	0.304±0.017	0.269±0.007	0.310±0.018	0.276±0.006
median2	0.200±0.013	0.185±0.015	0.234±0.019	0.226±0.020	0.237±0.019	0.230±0.021
mestranol_similarity	0.433±0.060	0.383±0.059	0.508±0.064	0.484±0.064	0.516±0.065	0.494±0.065
osimertinib_mpo	0.725±0.032	0.665±0.029	0.784±0.044	0.769±0.042	0.790±0.045	0.780±0.043
perindopril_mpo	0.391±0.006	0.355±0.007	0.454±0.027	0.440±0.020	0.460±0.029	0.448±0.022
qed	0.916±0.025	0.887±0.028	0.938±0.012	0.932±0.016	0.940±0.011	0.937±0.015
ranolazine_mpo	0.600±0.077	0.541±0.071	0.709±0.076	0.690±0.076	0.719±0.076	0.705±0.077
scaffold_hop	0.485±0.005	0.469±0.005	0.513±0.013	0.505±0.011	0.515±0.014	0.508±0.012
sitagliptin_mpo	0.219±0.080	0.141±0.105	0.299±0.083	0.215±0.156	0.307±0.084	0.222±0.161
thiothixene_rediscovery	0.336±0.033	0.307±0.028	0.428±0.014	0.403±0.029	0.437±0.012	0.412±0.029
troglitazone_rediscovery	0.256±0.071	0.230±0.064	0.305±0.085	0.289±0.091	0.310±0.087	0.295±0.094
valsartan_smarts	0.000±0.000	0.000±0.000	0.000±0.000	0.000±0.000	0.000±0.000	0.000±0.000
zaleplon_mpo	0.396±0.049	0.327±0.047	0.484±0.037	0.447±0.028	0.492±0.036	0.459±0.026
Sum	9.788	8.669	11.862	11.161	12.067	11.410