Accelerating Molecular Design: A Framework for High-Dimensional Bayesian Optimization in Chemical Solubility Prediction

CHBE 413/CHEM 452, Group 5
Chengxi Jiang, Tiffany Guo, Javier Carreon

Abstract

In this study, Bayesian optimization and methods designed for data of higher dimensions (HEBO, SAASBO, and TurBO) in tandem with a Gaussian Process model were utilized to optimize molecules within a dataset for aqueous solubility from the Autonomous Energy Materials Discovery research organization. The dataset was cleaned to remove missing values, duplicates, and outliers. Standard Bayesian optimization was shown to converge to the best performance at 15 iterations. The three methods all converged to a final iteration with similar evaluations of RMSE. For obtaining 90% convergence, TurBO was shown to be the fastest at 4 iterations, but for higher convergences such as 95%, HEBO was faster at 7 iterations. It was also determined that out of all the features of each molecule in the dataset, MolLogP was the most important for optimization. Overall, it was determined that high-dimensional Bayesian optimization was very effective for training a Gaussian Process model on the dataset used.

Introduction

The pursuit of optimal molecular compounds is fundamentally constrained by the molecular design problem, a challenge characterized by the immense complexity of chemical space. The number of plausible, stable molecules is inconceivably vast, creating a functional curse of dimensionality where the volume of the potential search space grows exponentially with the number of variables used to define a molecule. This exponential scaling renders any exhaustive combinatorial search or simplistic sampling strategy computationally intractable, effectively transforming the search for a high-performing molecule into a quest for a singular point in an effectively infinite and topologically complex landscape.

In response to this fundamental limitation, Bayesian Optimization (BO) has emerged as a premier, sample-efficient framework for navigating such complex and expensive-to-evaluate black-box functions. Its efficacy stems from a principled sequential decision-making process that intelligently navigates the trade-off between exploration and exploitation. BO operates by constructing a probabilistic surrogate model, which uses all prior observations to approximate the unknown objective function while quantifying prediction uncertainty. This model then informs an acquisition function, which uses these probabilistic estimates to select the single most informative point for the next evaluation. BO systematically reduces global uncertainty while converging on high-performance regions by iteratively updating its belief with each new data point, minimizing the number of costly function evaluations required to arrive at a near-optimal solution.

However, the application of standard BO to molecular design encounters a significant impediment rooted in the field's fundamental data representation. To be processed computationally, molecules must be translated from their structural form into a quantitative numerical format through molecular descriptors. The need for comprehensiveness necessitates a high dimensionality. Standard implementations often use vectors with a thousand or more bits to ensure a unique and informative representation for a diverse set of molecules. Consequently, even a relatively simple fingerprint projects the optimization problem into a search space of several hundred to several thousand dimensions, fundamentally altering the nature of the optimization challenge.

This high-dimensional space is inherently hostile to the standard BO framework, which typically relies on Gaussian Process (GP) surrogate models. The effectiveness of a GP depends on its ability to measure similarity between data points using a kernel function. In a high-dimensional descriptor space, data points become exceedingly sparse, and a phenomenon known as distance concentration occurs, where most pairwise distances become statistically similar. This renders the kernel unable to distinguish meaningful structure, causing the GP's predictions to lose accuracy and its uncertainty estimates to become less informative. Consequently, the acquisition function can no longer guide the search intelligently, and the optimization's performance degrades, often reducing to an inefficient, random exploration of the chemical landscape. This fundamental breakdown creates a clear need for algorithms specifically designed to maintain efficiency in high dimensions. To address this critical bottleneck, this study presents a comprehensive benchmark evaluating the performance of specialized High-Dimensional Bayesian Optimization (HD-BO) methods for the task of molecular solubility optimization.

Methods

The standard Bayesian Optimization method usually operates in a sequential, iterative loop guided by a Gaussian Process surrogate model. The optimization begins with a few initial observations to fit the initial GP model, which provides a predictive mean and uncertainty across the search space. In each iteration, the GP will be refitted to the accumulated data. After that, the Expected Improvement acquisition function is calculated from scratch using mean, uncertainty, and the best observed value so far. Then, maximizing EI, the system ensures that the new data point will provide the most useful information to improve the model and get closer to the true optimum. Finally, the true objective function is evaluated, and a new observation is added to the training dataset, effectively shrinking the GP uncertainty in that region for the subsequent iteration.

The following methods will be applied to a solubility dataset named Aqueous Solubility Data Curation (AqSolDB) created by the Autonomous Energy Materials Discovery (AMD) research group (Sorkun, Khetan, & Er, 2019). The dataset consists of aqueous solubility values of 9,982 unique compounds curated from 9 different publicly available aqueous solubility datasets. In addition to curated experimental solubility values, AqSolDB also contains some relevant topological and physico-chemical 2D descriptors calculated by RDKit and validated molecular representations of each of the compounds(Sorkun et al., 2019).The solubility dataset has 17 features, and uses solubility as the target. Although using standard Bayesian Optimization is a classical method to predict chemical solubility, normal Bayesian Optimization struggles when dimensions become more. The curve of the best solubility found and improvement per iteration will appear to be flat and show no improvement after the initial few steps. That's because Bayesian Optimization is searching a continuous domain that a poorly chosen true function surrogate has constrained. Although the high dimensionality of the feature space is not the root cause of the BO's failure, it exacerbated the issue. Therefore, applying a high-dimensional Bayesian Optimization method may be a better way to optimize solubility based on the dataset.

HEBO, SAASBO, and TuRBO are all high-dimensional Bayesian Optimisation methods; TuRBO is the most popular one among them. HEBO, which is a Heteroscedastic and Evolutionary Bayesian Optimisation method, is a solver designed to tackle the significant performance challenges caused by heteroscedasticity and non-stationarity and is often found in black-box optimization hyperparameter tuning tasks. HEBO applies non-linear input and output warping to solve issues. More specifically, it uses output transformations to handle heteroscedasticity and input warping to correct for non-stationarity. It further enhances performance by employing multi-objective acquisition ensembles to improve queried configurations(Cowen-Rivers et al., 2020c) . SAASBO, or Sparse Axis-Aligned Subspace Bayesian Optimization method, addresses the problems of dimensionality by defining Gaussian process surrogate models on sparse axis-aligned subspaces. SAASBO uses a sparsity-inducing SAAS function prior over the kernel hyperparameters, which are length scales, concentrating irrelevant dimensions near zero, which is a form of automatic relevance determination. This prior structure adaptively compromises between flexibility and parsimony, helping the model quickly identify the low-dimensional subspace relevant to the unknown objective function(Eriksson & Jankowiak, 2021). TuRBO, which is the Trust Region Bayesian Optimization method, is a global optimization technique designed to solve high-dimensional problems through the local

probabilistic approach. Instead of relying on a single global surrogate model that often struggles with function heterogeneity and over-exploration, TuRBO maintains a collection of independent local Gaussian Process models. Each local GP model operates within a constrained hyperrectangle or trust region, where the model is believed to be accurate. Global optimization is achieved by managing multiple simultaneous local optimization runs and utilizing a strategy to efficiently allocate samples to the most promising trust regions(Eriksson et al., 2019).

Here, a simple version of HEBO will be used. This method starts with employing QuantileTransformer as a non-linear input warping method to map the hyperparameter space to a normalized Gaussian distribution. It can improve the fit quality of the Gaussian Process. HEBO also applies PowerTransformer to the RMSE values, which helps stabilize the variance, in other words, addressing heteroscedasticity before the GP models the objective. Then, a Gaussian Process Regressor will be fitted to this doubly-warped data to serve as the probabilistic surrogate model to estimate the mean and uncertainty of the objective function. Next, employing evolutionary mutation around the best points, alongside random sampling to generate candidates for the Expected Improvement (EI) calculation. Evaluating the top-scoring hyperparameter set through the XGBoost cross-validation function, and adding the new data to the next iteration model. This combination of warping inputs, GP modeling, and diverse candidate generation mimics the HEBO framework. Here, Google Colab can not support BoTorch running, so the report code uses a simpler version of the SAASBO method to imitate the dimensionality-sparsifying behavior of SAASBO in the paper. It begins with a warmup phase that evaluates many random parameter vectors to build an initial dataset. And then use LASSO regression, which is familiar, to identify the most influential hyperparameters, those with non-zero (or large-magnitude) coefficients. Optimization is then restricted to this reduced subspace by running the GP-EI optimizer only over the selected dimensions. Here, the GP-EI optimizer code uses what has been in smaller HEBO previously. The remaining inactive parameters are filled in using the best-performing warmup sample. This version mimics SAASBO's idea of shrinking irrelevant dimensions while focusing Bayesian Optimization effort on the active ones. Here, TuRBO begins with the search by defining a trust region in the normalized hyperparameter space, centered around the best starting point. In each iteration, a Gaussian Process model is fitted to all accumulated data points, providing an objective function for the global model. Candidate points of TuRBO are sampled uniformly through Sobol sequences. Candidate points are also only from within the current, confined trust region defined by the center and length. Then, these local candidates will be ranked using the Expected Improvement acquisition function. Applying the Expected Improvement acquisition function will ensure the chosen points balance exploitation and exploration strictly within the local area. Using the XGBoost cross-validation function to evaluate the highest-EI candidates in the local. If this local yields an improvement, the trust region will expand, and if the batch local fails, the trust region will shrink. With the above mechanism, the algorithm adapts to the objective function's local smoothness. Finally, the center of the trust region will be moved to the globally best point found so far, concentrating the search on the most promising area. The model is always global, and the search is always local. Using the global model to inform the local search mimics the performance of the TuRBO framework.

Results and Discussion

As stated previously, traditional Bayesian Optimization methods in a continuous domain begins to fail when the number of features exceeds 10. The goal was to verify this claim by performing a Bayesian Optimization with the Gaussian GP in a discrete domain and then a continuous domain. If the optimization failed or degraded in the continuous domain, it would be justification for using high dimensional bayesian optimization methods such as HEBO, SAASBO, and TuRBO. Afterwards, the performance of each of these methods was analyzed and compared with the ultimate goal of determining which of the three most effectively minimized solubility RMSE.

Data & Problem Formulation

The chosen solubility dataset first underwent standard preprocessing with data cleaning by removing missing values, outliers, and duplicate values resulting a total amount of 8,673 molecules from the initial 9,982. All variables were standardized to ensure comparability between features. Correlations between variables are visualized with correlation heatmap (Figure 1).

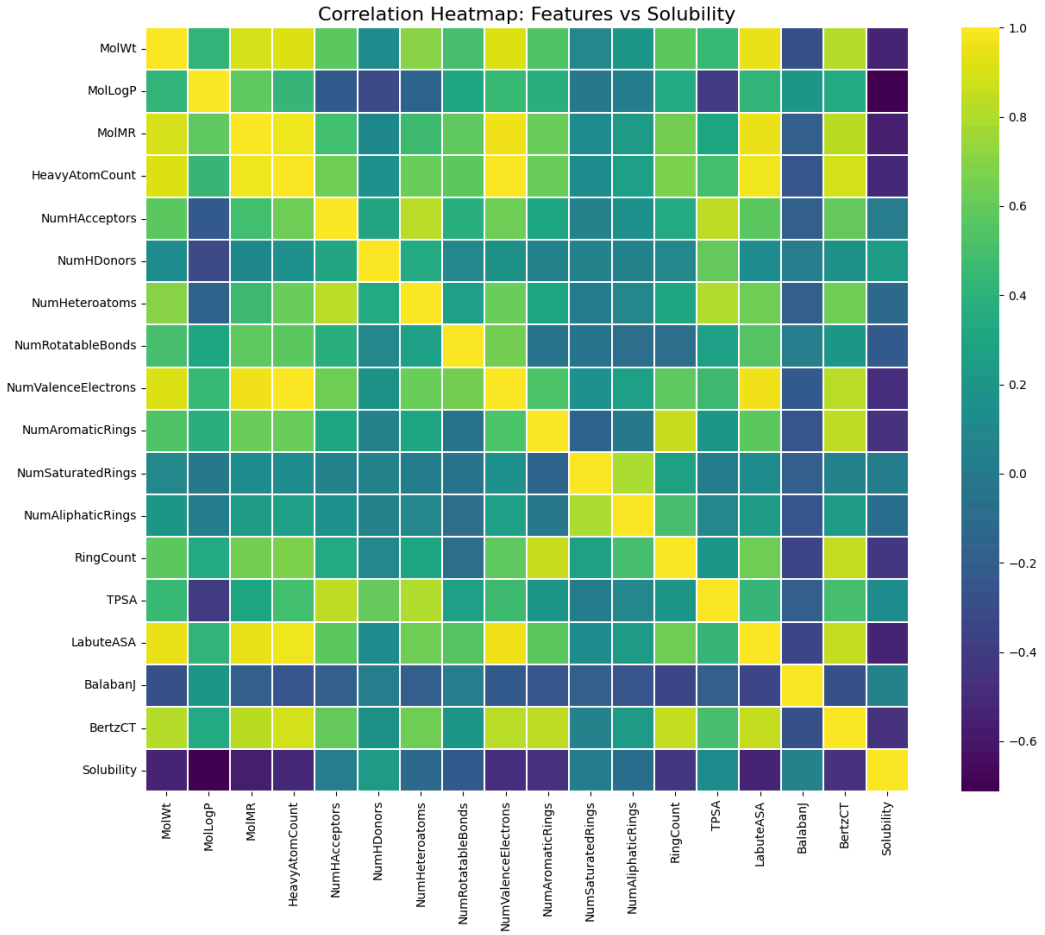

Figure 1. Molecular Features vs. Solubility Correlation Heatmap

From initial correlation analysis, nine variables exhibited correlations above 0.5 indicating a moderate relationship. The highest correlation between feature variables was approximately 0.4 between the topological polar surface area (TPSA) and MolLogP. Since there wasn't a strong correlation between feature variables identified, issues with multicollinearity during gaussian process modeling were unlikely.

Standard Bayesian Optimization Setup

Using the scaled solubility dataset, Bayesian Optimization (BO) was performed to identify the molecule with the highest solubility. BO was performed by sampling observations from the dataset rather than incorporating new experimental data. The Gaussian Process (GP) model was configured using RBF kernel and White kernel to smooth and reduce noise. The GP was trained on 10 initial observations where the GP predicted the mean and uncertainty of all unobserved molecules. Using those values the expected improvement (EI) was calculated. From the EI the observed molecule with the highest expected improvement was selected and added to the observed dataset. The EI for maximization would be evaluated as:

$$EI(x) = (\mu_f(x) - f_{best})\Phi(Z) + \sigma_f(x)\phi(Z)$$

Where $Z \equiv (\mu_f(x) - f_{best})/\sigma_f(x)$ and where Φ and ϕ are the CDF and PDF of standard normal. $\mu_f(x)$ was the predicted mean, $\sigma_f(x)$ was the predicted standard deviation, and f_{best} was the best observed value so far.

This process was repeated for 20 iterations. The corresponding convergence curves were plotted to examine how many iterations were needed to reach the best-observed solubility. A 2D PCA projection helped visualize the solubility distribution. The gaussian process surrogate prediction was plotted against true solubility values. The datapoints do not fall perfectly on the diagonal indicating that the predictions deviate from the ground truth. It appeared that the gaussian process model provided more accurate predictions at higher solubilities but struggle at lower values with the predicted solubilities being higher than the true solubilities. Although the discrete domain aligns with the fact that molecules are inherently discrete, comparing standard BO to high-dimension BO methods required evaluating a continuous domain BO as well.

When performing the BO in a continuous domain instead of a selection of a set of initial observations, random vectors are sampled. Those vectors did not necessarily correspond to real molecules and caused errors since those values may not represent feasible structures in a chemical context. To remedy this, values from real molecules were chosen for intialization and k-nearest neighbors (KNN) to ensure evaluations corresponded to valid structures. For each iteration with a total of 20 iterations, a new surrogate gaussian process was fitted where the mean and uncertainty was used to calculate the EI. The implementation with these changes was shown in Figure 2 with a convergence plot with the best solubility for each iteration.

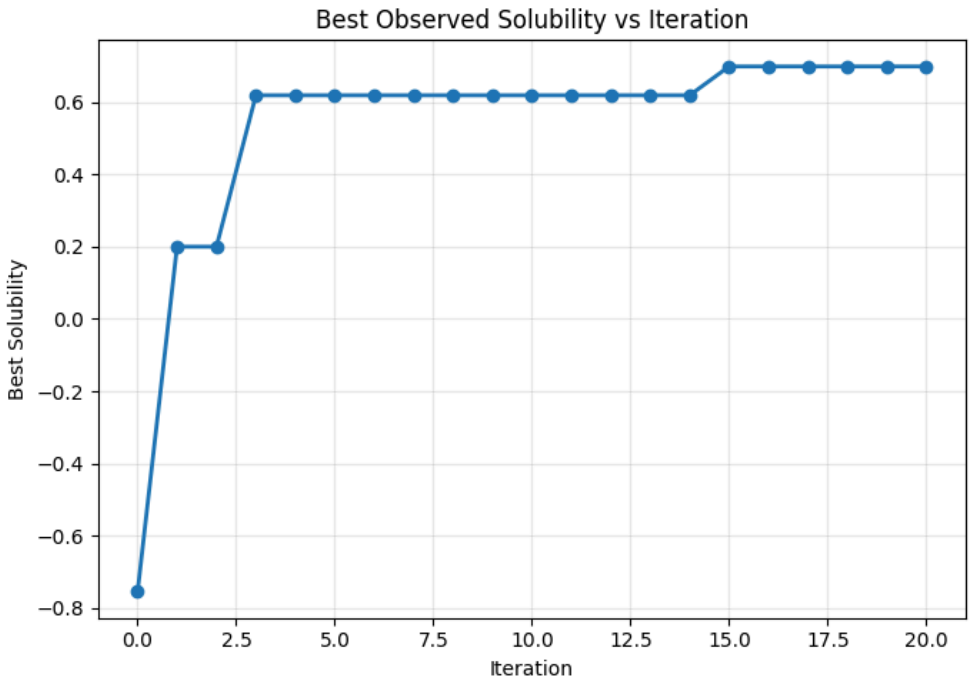

Figure 2. Convergence Plot for BO Continuous Domain Implementation

From this implementation, the BO reached the best-observed solubility at 15 iterations out of the 20 with rapid initial improvement at 3 iterations (Figure 2). The steep early improvement proves that the BO is highly efficient at finding the high-solubility candidates.

High-dimension Solubility Optimization

An alternative approach to the continuous BO implementation was to applu high-dimensional BO methods like HEBO, SAASBO, and TURBO to minimize of RMSE which serves as an indicator for predictive accuracy. In this case, to use EI which was a maximization acquisition function, -RSME was used corresponding to minimizing RMSE.

For all three methods, the data was split using train-test split with a 20% test size to replicate the effect of collecting experimental data. XGBoost served as the predictive model that was tuned and evaluated. All of the implementations operate in a normalized [0,1]^d space.

For HEBO and SAASBO, simplified versions of the original algorithms were implemented as described in the methods. For simple-HEBO the GP surrogate model utilize the Matern kernel combined with a WhiteKernel. This method was expected to perform sufficiently with a moderate number of dimensions (17 dimensions). The simplified SAASBO compared to the full SAASBO utilizes LASSO to select important dimensions where the only the important dimensions are utilized for reduced search space.

Unlike HEBO and SAASBO, TuRBO restricts its search to a local trust region rather than the entire domain like HEBO and SAASBO. This specific implementation used one trust region

which expands and shrinks depending on success factors tied to -RMSE. The use of one trust domain should prove to be sufficient since there was only a moderate number of dimensions.

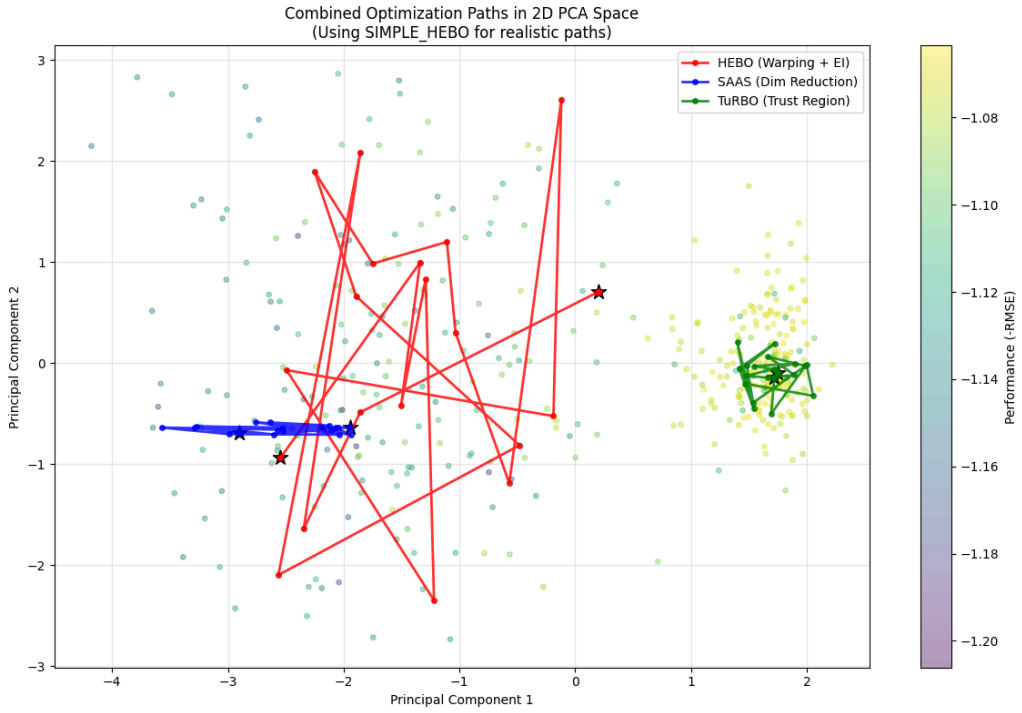

Figure 3. Optimization paths for Simple HEBO, SSASBO, and TuRBO

Optimization paths shown in a 2D PCA space illustrated how the different optimization methods explore and converge (Figure 3). HEBO explores broadly over a large region of space with an erratic and zig zag path and does not appear to effectively converge. This behavior was characteristic of HEBO as the method encourages exploration and samples across a large area. The lack of convergence may be due to the lack of iterations.

Unlike HEBO, SAASBO and TuRBO remain in compact regions of the PCA space and eventually converge. For SAASBO, mainly search across PC1, reflecting the behavior of SAASBO with its quick initial dimensional reduction. This restricts its search to a narrow subspace shown as a narrow band. In terms of TuRBO the search region remains localized as the trust region rarely expands. This demonstrates that improvements in -RMSE were relatively small and sporadic keeping the region highly local. Overall, these patterns align with the behavior of each method and provide insight on how each algorithm allocates exploration vs. exploration.

In terms of actual performance of these HD-BO methods, the accuracy and efficiency was visualized through the convergence plots of the validation RMSE and solubility proxy. Final test performance comparison and along with exploration efficiency serve to add understanding overall model performance on new data.

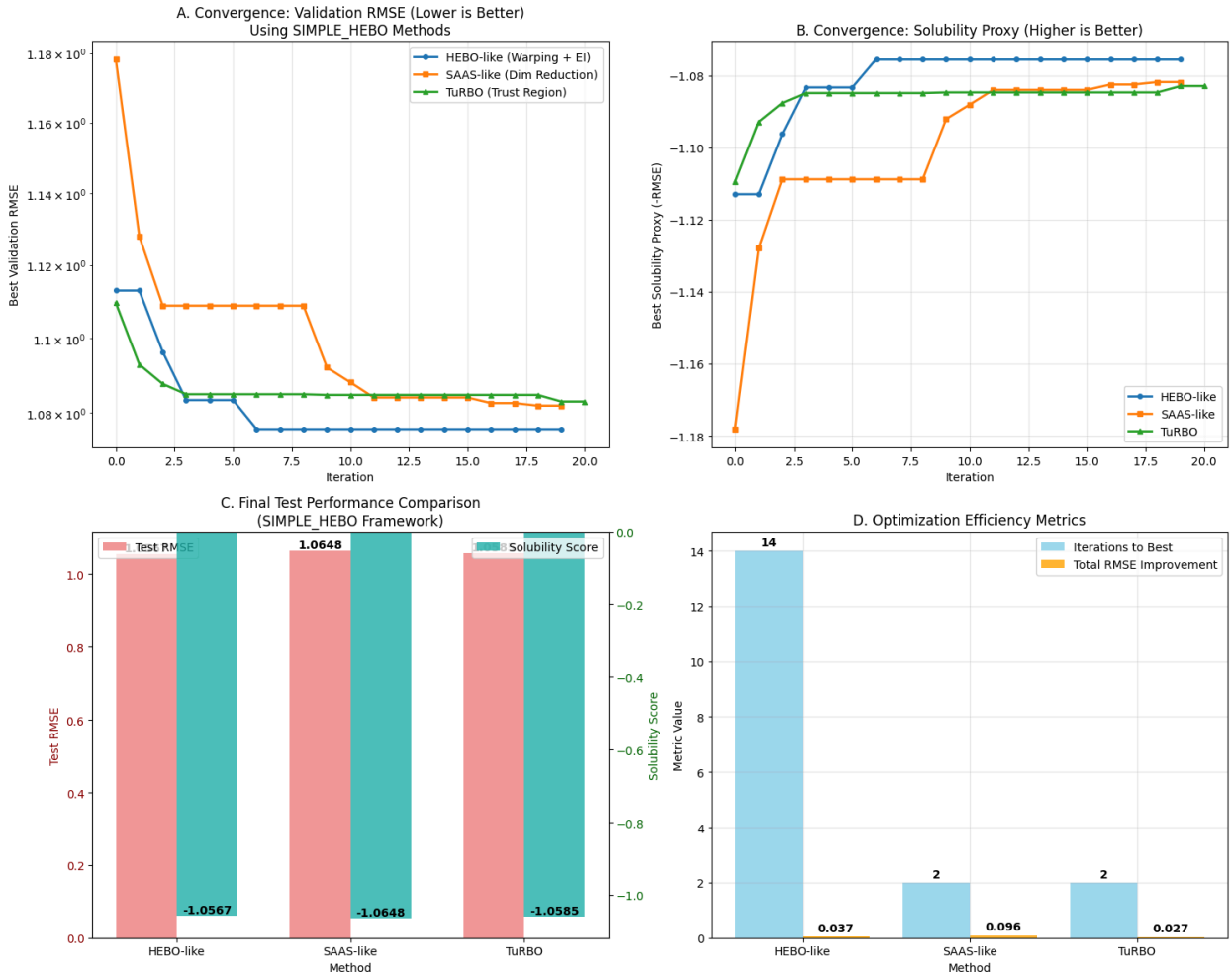

Figure 4. Comprehensive Method Comparison Between HEBO, SAASO, and TURBO

These visualizations shown in Figure 4A and 4B indicate that the HEBO-like method or simple-HEBO reached the lowest validation RMSE and best solubility proxy more rapidly than other methods. TuRBO performed second best where simple-HEBO and TuRBO both started at similar RMSE values before the first iteration but simple-HEBO saw a greater RMSE reduction. For SAAS-like BO it started at a higher RMSE value and declined steeply where with more iterations it reaches similar values to TuRBO.

In terms of final test performance, the HEBO-like has the lowest RMSE of 1.0567 but only by a small margin to the second lowest of 1.0585 (Figure 4C). Overall, all methods were within 0.01 RMSE if one another, indicating performance differences were not very significant.

For optimization efficiency the HEBO-like BO required 14 iterations compared to two for SAAS-like and TuRBO. This was expected due to the characteristic of HEBO to explore widely before improvement. SAAS-like also demonstrated the largest total RMSE improvement suggesting its effectiveness and fast convergence. This would be an important consideration if speed was prioritized of whether the high iteration cost of HEBO-like implementation outweighed the minimal performance increase.

Table 1. *Paired t-tests and Wilcoxon signed-rank tests and their corresponding p-values*

Pair	t-stat	p-value(t-stat)	W-stat	p-value(W-stat)
HEBO vs. TuRBO	0.76	0.44	749562.5	0.87
SAAS vs. TuRBO	-0.95	0.34	752450.5	0.98
HEBO vs. SAAS	1.38	0.17	752976	0.99

Pairedt-tests and Wilcoxon signed-ranked tests were used to evaluate the differences between RMSE values and their corresponding directions respectively(Demˇsar, 2006). As shown in Table 1, there appears to be no significant value and directional difference between the RMSE for each pair of methods. All p-values were well above standard significance thresholds of 0.05-0.1. This confirms that there were no statistical differences between final RMSE outcomes.

Table 2. *Convergence speed analysis for each method*

Method	90% Convergence	95% Convergence	Improvement
HEBO-like	7	7	0.0374
SAAS-like	11	12	0.0963
TuRBO	4	20	0.0266

Table 2 summarizes the convergence speeds of the three methods along with their overall improvement. The convergence speed of each method shows that the HEBO-like BO reaches convergence from 90% to 95% by the 7th iteration indicating fast late stage improvement. TuRBO shows rapid initial improvement reaching 90% convergence at the 4th iteration but slower late-stage improvement requiring 20 iterations to reach 95% convergence. SAAS-like BO showed a result inbetween TuRBO and HEBO-like reaching 90% to 95% convergence wih 11-12 iterations. These results highlight that TuRBO would be more ideal rapid moderate convergences and HEBO-like would be more efficient for reaching higher convergences.

Feature Importance Analysis

Since all methods produced similar RMSE values, it was also important to understand the effectiveness of the top descriptors in differentiating between high and low solubility along with how feature importance varies between method.

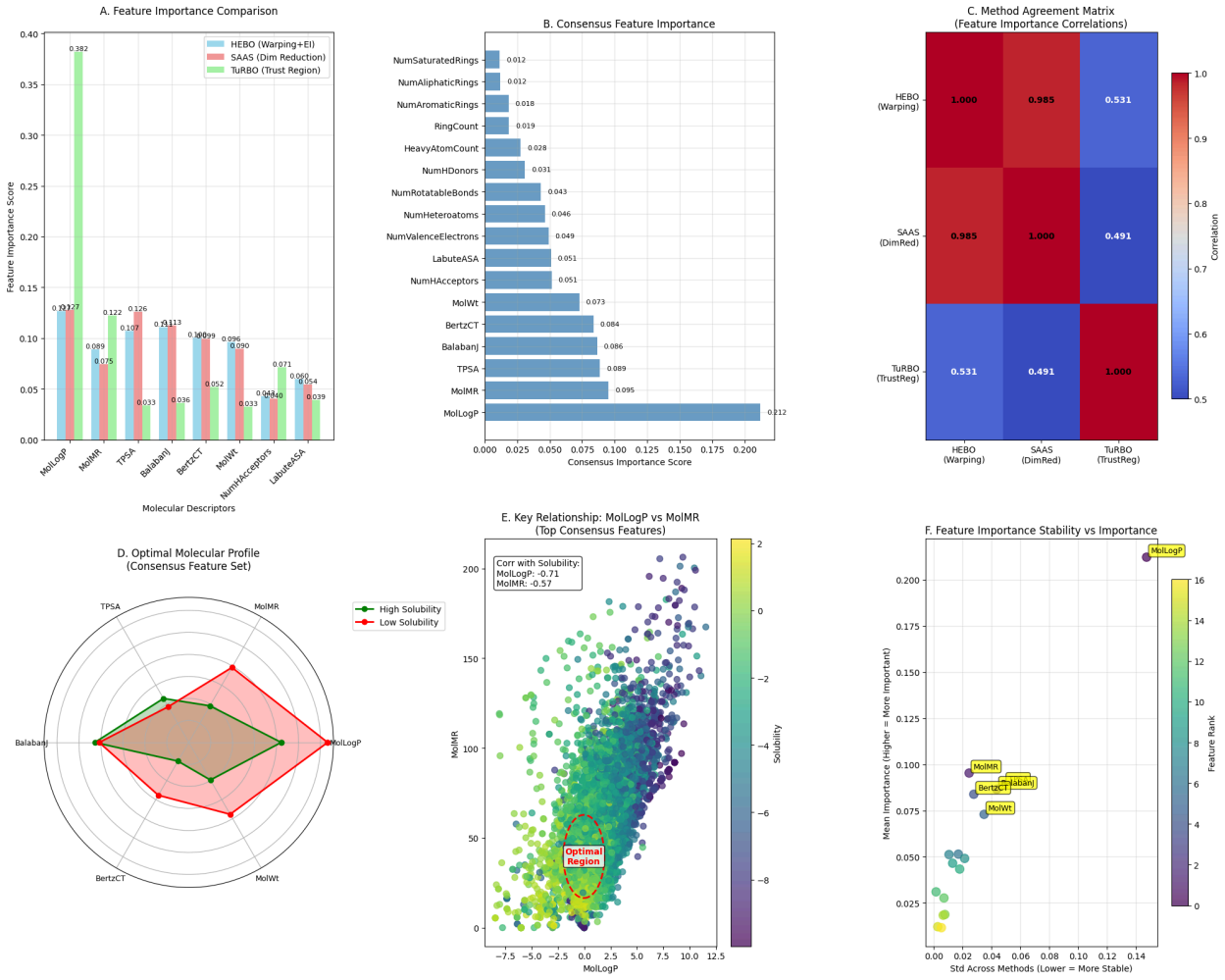

Figure 5. Comprehensive Descriptor Analysis of HEBO, SAASBO, and TuRBO

In Figure 5D, the lack of overlap between the radial area for MolMR, MolLogP, MolWt, and BertzCT indicate that these variables were able to clearly difference between high and low solubilities. The top two consensus features of MolLogP and MolMR show a strong correlation with solubility when plotted, reinforcing their importance (Figure 5E).

Feature importance varied with method as TuRBO placed the strongest emphasis on MolLogP compated to other methods (Figure 5A). Otherwise HEBO and SAASBO showed more distributed importance profiles. This was reflected in the method agreement matrix where TuRBO's feature importance correlations with HEBO and SAASBO are around 0.5 whereas HEBO with SAAS are high around 0.95 (Figure 5C).

An important issue to consider when using MolLogP as a feature variable was the high standard deviation it displays. In Figure 5F, it had the highest standard deviation out of all the top important feature variables around 0.14. This may be due to outlier values of MolLogP apparent in Figure 5E which would lead to some deviations in the feature importance. In addition, it

would be important to note that the feature importance of MolLogP was strongly skewed due to the strong feature importance TuRBO placed on the variable.

MolLogP which was a molecular descriptor that quantifies a molecules lipophilicity would be a strong indicator for solubility as solubility depends heavily on polarity. However, other factors such as ionization state along with experimental conditions like temperature, pressure, and state could contribute to measured solubility values. Like MolLogP, MolMR or molecular molar refractivity also serves as an indicator to polarity along with molecular weight.

Conclusion and Recommendations

In conclusion, the HD-BO methods seemed to perform well enough for finding a minimal RMSE value, as all of their final iterations have had very similar evaluations. Therefore, to recommend a method is more of a question of speed and optimization. It was established that of the three for reaching 90% convergence, TurBO is the best choice to go with, but for higher convergences of 95% and above, the HEBO-like method is the better option. Each of the methods share similar importance for the features in a molecule, favoring MolLogP the most.

Overall, HD-BO provides a powerful and sample-efficient framework for molecular property optimization, dramatically outperforming conventional methods. It can be used to find an optimal molecule geared for high or low solubilities. For future work, it can perhaps be used to optimize for another objective simultaneously, such as toxicity of a molecule. Another thing to be considered is constraints being put in place for how feasible an optimal compound chosen by a process is, as how KNN was used for standard BO for this study. Alternatively, a graph neural network which has much more of a direct approach on the structure of a molecule can be used instead of a GP model used in this study.

Group Contribution

Method & Code: Chengxi Jiang
Result & Discussion: Tiffany Guo
Abstract& Introduction & Conclusion : Javier Carreon

Reference

Cowen-Rivers, A., I., Lyu, W., Tutunov, R., Wang, Z., Grosnit, A., Griffiths, R. R., Maraval, A. M., Jianye, H., Wang, J., Peters, J., & Ammar, H. B. (2020c, December 7). HEBO pushing the limits of Sample-Efficient Hyperparameter Optimisation. arXiv.org. https://arxiv.org/abs/2012.03826

Demˇsar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. In Dale Schuurmans (Ed.), Journal of Machine Learning Research (Vol. 7, pp. 1–30) [Journal-article]. https://www.jmlr.org/papers/volume7/demsar06a/demsar06a.pdf

Eriksson, D., & Jankowiak, M. (2021, February 27). High-Dimensional Bayesian Optimization with Sparse Axis-Aligned Subspaces. arXiv.org. https://arxiv.org/abs/2103.00349

Eriksson, D., Pearce, M., Gardner, J. R., Turner, R., & Poloczek, M. (2019, October 3). Scalable global optimization via local Bayesian optimization. arXiv.org. https://arxiv.org/abs/1910.01739

Sorkun, M. C., Khetan, A., & Er, S. (2019). AqSolDB, a curated reference set of aqueous solubility and 2D descriptors for a diverse set of compounds. Scientific Data, 6(1), 143. https://doi.org/10.1038/s41597-019-0151-1