CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Human-in-the-loop control of molecular reinforcement learning with online adaptive classifiers

Master's thesis in Computer science and engineering

Edwin Holst
Preetha Mutharasu

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2023

Master's thesis 2023

Human-in-the-loop control of molecular reinforcement learning with online adaptive classifiers

Edwin Holst, Preetha Mutharasu

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2023

Edwin Holst, Preetha Mutharasu

Supervisor: Rocío Mercado, CSE
Advisor: Jon Paul Janet, Molecular AI - AstraZeneca BioPharmaceuticals R&D
Examiner: Ola Engkvist, CSE

Master's Thesis 2023
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Gothenburg, Sweden 2023

Abstract

The early stage of drug discovery faces significant challenges of screening through a vast number of compounds to identify potential drug candidates for specific diseases. Amidst a range of AI-based systems employed in efficiently identifying or generating potential drug candidates, this thesis focuses on REINVENT, a prominent production-ready tool for de novo design. Despite being advanced with multiple scoring options, it is challenging for REINVENT to capture human intuitions for generating desired outcomes. This thesis explores the significance of integrating human feedback to REINVENT through interactive visualization and online learning models. A range of methods have been employed during the development, First to enhance users' understanding of generated compounds, diverse compound generation was studied, leading to an interactive visualization platform. We aim to offer a platform enabling effective user guidance. Second, to capture human preference, human feedback was integrated as a separate scoring function using online learning models. Considering the time and resources, surrogate user models were employed to represent real chemists, allowing for efficient development. During this testing, various aspects of the proposed system, including different online learning models, rating frequencies, sampling methods, and the number of rated molecules were tested and estimated. An evaluation experiment involving eight human participants demonstrated that integrating the HITL system to REINVENT can accelerate the drug discovery process by integrating AI capabilities with human expertise. It can effectively enhance the identification of valuable molecules, reduces compound analysis time, and ultimately results in improved patient outcomes and cost-effectiveness.

Keywords: Human-in-the-loop, drug discovery, generative AI, REINVENT, visualization, de novo.

Acknowledgements

We would like to express our gratitude to Ola Engkvist, our Examiner from the Department of Computer Science and Engineering at Chalmers, for his feedback and support throughout the project. Also, we extend our gratitude to Jon Paul Janet, our Supervisor at AstraZeneca, for laying the foundation of our thesis and providing us with outstanding support during our time at AstraZeneca. His guidance and feedback have played a pivotal role in shaping the outcome of our project. Furthermore, We would like to convey our sincere thanks to Rocio Mercado, our Academic Supervisor from the Department of Computer Science and Engineering at Chalmers, for her valuable guidance, advice, and feedback. Thank you all for your assistance in helping us achieve our goals.

Edwin Holst Preetha Mutharasu, Gothenburg, 2023-06-28

Contents

List of Figures

List of Tables

1

Introduction

The early stage of drug discovery is a crucial phase in drug development, as it involves identifying potential drug candidates which act against a specific disease, usually via interaction with a target protein implicated in the disease [1]. This discovery phase can take years of research to understand the biological mechanisms behind the disease and screen through libraries of compounds for their desired activity against the target. However, the vast number of possible compounds that could be screened makes it a challenging, time-consuming, costly, and complex process. The advanced use of technology like in silico or virtual trials [2] significantly accelerates the drug discovery process by allowing for cost-effective identification and optimization of promising drug candidates through computer simulations. Despite the advances, the challenge of efficiently searching and identifying novel active compounds from the staggering 10^{60} available compounds [3] still remains a significant challenge in drug discovery.

AI-based generative models present promising solutions for this challenge by proposing promising small molecules and exploring the chemical space. REINVENT, a production-ready AI generative model tool for drug or de novo design is one proven solution to effectively navigate the chemical space and generate relevant compounds [4][5]. REINVENT utilizes reinforcement learning to generate batches of compounds that are scored and optimized based on a user-defined scoring function, thus its performance is determined by this function. While REINVENT offers multiple scoring options, it can be challenging for users to effectively express their desired outcomes [4][5].

To further enhance the model's ability to navigate toward the desired outcome, this project aims to integrate human feedback into the process. As the distribution of molecules generated by REINVENT changes during the RL process, an online learning approach [6] was used to capture and predict user preferences during a run. This predicted user preference was simultaneously used as one of the optimization goals during the RL process.

In order for users to effectively guide the model during a run, understanding the current state and direction of the generated compounds could be an important aspect. To improve this understanding, a web application was created, displaying a dashboard of statistics and visualizations of the REINVENT run, with the goal of improving understanding of the diversity and quality of generated compounds.

2

Theory

In this section, we provide an overview of the field of computational drug discovery and molecular optimization, highlighting the significance of REINVENT, a state-of-the-art model for molecular optimization. Furthermore, we offer a brief overview of the concepts of Human-in-the-Loop, Numerical Molecular Representation, Dimensionality Reduction techniques, and Machine Learning algorithms. These concepts, which are utilized extensively in this project,

2.1 Computational drug discovery

Advances have been made in recent years in solving complex computational molecular optimization problems encountered in drug discovery, focusing on identifying chemically valid molecules with high diversity and synthesizability along with a desired property profile [7]. Molecular optimization methods consist of two parts: a molecular assembly strategy that defines the chemical space and an optimization algorithm that navigates the chemical space [7].

The molecular assembly strategies explored by researchers include string-based, graph-based, and synthesis-based approaches. String-based strategies represent molecules as strings and modify them directly, either character-by-character or through more complex transformations based on a specific grammar [7], [8]. There are two commonly used string representations: Simplified Molecular-Input Line-Entry System (SMILES) [9] and SELF-referencIng Embedded Strings (SELFIES) [10]. Graph-based strategies define molecular identities through two-dimensional graphs with nodes and edges representing atoms and bonds [11]. There are two main assembling strategies for molecular graphs: atom-based and fragment-based. Synthesis-based strategies target only synthesizable compounds, which can be divided into template-free and template-based approaches [7].

The optimization algorithms commonly used in molecular optimization include screening, genetic algorithm, Monte-Carlo tree search, Bayesian optimization, variational autoencoders, score-based modeling, hill climbing, reinforcement learning (RL), and gradient ascent [7].

2.2 REINVENT

REINVENT is a versatile generative AI tool designed for a range of applications, including Reinforcement Learning, curriculum learning, and transfer learning [4], [5]. A recent benchmark study demonstrated that REINVENT is the most efficient tool, among 25 others, for solving chemical optimization problems in terms of scoring oracle calls [7].

A typical REINVENT run in the RL mode comprises of several key components working together: an RL agent, a multiparameter objective (MPO) score, a prior model, a diversity filter, and an inception module [4], [12]. The RL agent is a generative model, using a pre-trained recurrent neural network to produce molecules in the SMILES format. The user-specified MPO score serves as the primary optimization goal, with a range between zero and one. The prior is a copy of the agent, but is not updated during the run. Instead, it is employed to prevent the agent from diverging too far from the original distribution by computing the likelihood of molecules. The diversity filter enhances the diversity of the generated molecules by setting the score to zero if a user defined number of similar molecules have already been seen [12]. The inception module stores the highest-scoring molecules during a run, where some of these are used when the agent is updated.

An RL run consists of executing a specified number of steps. In each step, the agent generates a batch of molecules, which are then scored using a combination of the MPO, prior function, and diversity filter. The inception module enhances the batch by including some high scoring molecules from previous steps. The agent is then fine-tuned on the enhanced batch and updated according to the scored molecules [12]. Figure 2.1 provides an overview of the REINVENT loop during an RL run [4].

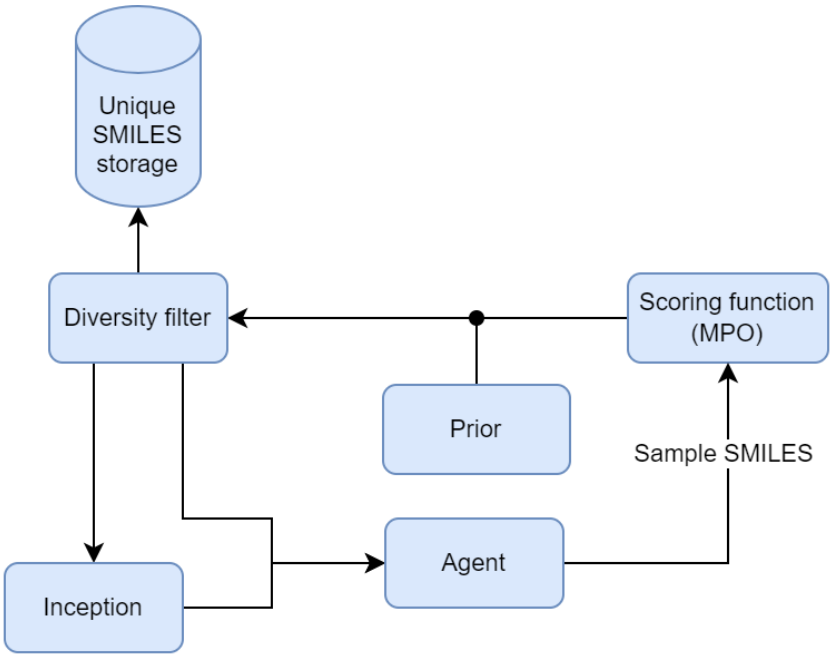

Figure 2.1: Schematic of REINVENT Model's Feedback Loop in RL Run
Overview of the REINVENT model's feedback loop during an RL run. The agent samples molecules in the SMILES format, which are then scored through the scoring function, prior function, and diversity filter. The inception module saves the best-scoring molecules and can augment the scored batch. The scored and potentially augmented batch is used to update the agent's posterior distribution.

A typical workflow of using REINVENT involves a user generating molecules with the model, evaluating them, and then adjusting the MPO iteratively to steer the model to generate molecules with desired traits [13]. REINVENT offers a great variety of different scoring functions, ranging from QSAR models to physics-based simulations such as docking [14], [15] or pharmacophore matching methods [16]. However, the iterative process of evaluating generated molecules and re-defining the MPO can often be time consuming, and requires broad expertise [13]. Additionally, there are some intrinsic qualities of molecules that can be difficult to encapsulate in terms of existing scoring functions or rules. These qualities can reflect the preferences and experiences of the expert users. Effectively utilizing the expertise of users is still an unresolved challenge that requires further investigation [13].

The intricacy of REINVENT's RL process makes it challenging to assess its effectiveness in achieving the objective of chemical diversity and to compare different runs. Currently, the feedback provided by REINVENT includes the loss of various scoring components, the comparison of the prior and agent likelihoods, and the generated molecules themselves. The difference between the agent and prior likelihood may serve as a proxy for diversity, but it does not provide a comprehensive understanding. Evaluating the generated molecules for accuracy is the best way to assess results, however, it is not feasible due to the large number of outputs, which can range from hundreds to thousands, making it challenging to comprehensively review the results in a timely manner.

2.3 Human-in-the-loop

HITL (Human-in-the-loop) is a popular approach in machine learning that leverages human knowledge to enhance the performance of ML models. In recent years, HITL has been applied to various ML tasks [17] with increasing popularity. One particular technique, Reinforcement Learning from Human Feedback (RLHF), has been used for fine-tuning language models [18]. RLHF operates by training a reward model based on human feedback. This reward model is then employed as a reward function optimized through the use of reinforcement learning [18]. It is suggested that RLHF plays a significant role in enhancing the adaptability and performance of Large Language Models, such as ChatGPT and GPT4 [19].

HITL for generative molecular models is largely uncharted territory, with only a few studies having explored its potential. In "Human-in-the-loop assisted de novo molecular design" by Liris Sundin et al., human feedback was utilized to adjust the weighting of different scoring components in the molecular optimization process [13]. However, creating a scoring function based on human feedback on individual molecules and using this feedback as a score in the reinforcement learning loop - an approach simliar to RLHF - has yet to been explored.

2.4 Numerical Molecular Representation

In order to harness human feedback on molecules, a numerical representation of these molecules is essential. This allows the application of various mathematical and computational algorithms in the study of molecules. Among the numerous techniques available to achieve this, one notable method is molecular fingerprinting. This method encodes the presence or absence of specific substructures within the molecules, thus facilitating their analysis and comparison [20].

Among the different types of molecular fingerprints, Extended-connectivity Fingerprints (ECFPs) are widely used due to their ability to capture the structural and topological features of molecules [21]. ECFPs generate a compact binary representation that efficiently encodes molecular substructures, and are applicable to various tasks such as molecular similarity analysis, virtual screening, and drug discovery.

Another approach is the use of Continuous and Data-Driven Descriptors (CDDD) [22]. CDDD leverages a pretrained encoder model derived from an autoencoder architecture to produce continuous and data-driven molecular representations. Compared to traditional molecular fingerprints, CDDD has demonstrated superior performance in certain tasks, showcasing its potential as a robust and versatile alternative for molecular representation [22].

2.5 Machine learning algorithms

In order to exploit HITL feedback, a framework for predicting human preferences is necessary. Classification is a widely used technique in supervised learning that

involves assigning data instances to predefined categories or classes. It enables automated decision-making based on learned patterns [23]. There are various algorithms employed for classification tasks, including logistic regression, K-nearest neighbors (KNN), and random forest.

Logistic regression is a linear classification algorithm that models the relationship between the input features and the probability of belonging to a certain class. It estimates the coefficients of the features to make predictions. Logistic regression is known for its interpretability and ability to handle large dataset efficiently when the dataset can be separated linearly [23].

K-nearest neighbors is a non-parametric lazy algorithm that classifies instances based on their similarity to the K-nearest neighbors in the training data. KNN makes predictions by majority voting or weighted averaging. It is a simple yet effective algorithm and can adapt well to complex decision boundaries. Despite being robust to noisy training dataset the performance depends on the data quality [23].

Random forest is an ensemble learning method that combines multiple decision trees in parallel to make predictions. Each tree is constructed using a subset of the training data and a random subset of features. Random forest improves prediction accuracy by reducing overfitting and capturing the collective knowledge of multiple trees [23].

2.6 Dimensional reduction techniques

To connect the high dimensional numerical representation of molecules with the visual aspects of this project, dimensionality reduction becomes a crucial technique to project high dimensional information to a lower dimensional space, that is visually easier to comprehend. It is a popular technique used in data analysis that involves reducing the number of variables in a dataset while trying to preserve information about the dataset. There are many different techniques used to reduce the number of dimentions, two of these are principle component analysis (PCA) [24], and Uniform Manifold Approximation (UMAP) [25].

PCA is a linear transformation technique. It calculates the eigenvectors and eigenvalues of the covariance matrix of the dataset and projects the data onto a lower-dimensional space. The resulting principal components are uncorrelated and capture the maximum amount of variation in the data. PCA is a powerful technique for identifying the most important features in a dataset and creating a reduced dimensional space for analysis and interpretation. [24]

UMAP, on the other hand, is a non-linear dimensionality reduction technique that uses a graph-based approach to preserve the local structure of the data. It constructs a high-dimensional graph that captures the relationships between neighboring data points and then optimizes a low-dimensional embedding that retains these relationships as much as possible. UMAP can be especially useful for visualizing complex datasets with non-linear relationships between variables. [25]

3

Methods

This chapter presents the methods and techniques utilized to visualize REINVENT runs, incorporate human feedback, and evaluate the proposed human-in-the-loop (HITL) system.

3.1 Visualizing REINVENT runs

Given the importance of understanding the current state and direction of the generated compounds for effectively guiding the model during a run, a variety of visualization techniques were explored. The aim of these visualizations was to shed light on the diversity and property distributions of generated molecules throughout the entire training run such that the user could better understand how these properties change over time.

3.1.1 Proposed Metrics

In this context, several metrics were proposed to illuminate various aspects of molecular diversity within a set of molecules and to quantify the shifts in distributions between different sets of molecules.

3.1.1.1 Scalar representation of Diversity

To quantify the diversity of a set of molecules M, we employ a diversity metric as described in Equation 3.1.

$$\text{Diversity} = \frac{1}{n} \sum_{i=1}^{n} ||m_i - \mu|| \tag{3.1}$$

Here:

- n is the number of molecules in the set M,
- m_i is the CDDD embedding of the i^{th} molecule,
- μ is the mean CDDD embedding of all molecules, given by $\mu = \frac{1}{n} \sum_{i=1}^{n} m_i$,
- and $||.||$ denotes the L2 norm (or Euclidean distance).

3.1.1.2 Unique molecules

A molecule m is defined to be unique with respect to the set of molecules M if its distance to every other molecule in M is greater than a predefined scalar value. This condition is formally defined in Equation 3.2.

$$\forall m' \in M, ||m - m'|| > scalar \tag{3.2}$$

Here:

- m' denotes each molecule in M,
- $||m - m'||$ represents the Euclidean distance between the CDDD embeddings of m and m',
- and *scalar* is a predefined threshold, set to 8 in this project.

Note that the scalar value of 8 is an arbitrary value and could be changed based on the use-case, were a higher value would make for a less strict criteria of uniqueness.

3.1.1.3 Shift in Molecular Distribution

We define a shift in the molecular distribution between two sets of molecules, M_1 and M_2, as the difference in the mean CDDD embeddings of the two sets. This is formally defined in Equation 3.3.

$$\text{Shift} = ||\mu_{M_1} - \mu_{M_2}|| \tag{3.3}$$

Here:

- M_1 and M_2 are the two sets of molecules,
- μ_{M_i} is the mean CDDD embedding of all molecules in the set M_i, given by $\mu_{M_i} = \frac{1}{n_j} \sum_{j=1}^{n_1} m_j$, where m_j is the CDDD embedding of the j^{th} molecule in M_i and n_i is the number of molecules in M_i,
- and $||.||$ denotes the L2 norm (or Euclidean distance).

3.1.2 Visualizing Molecular Generation Metrics during Reinforcement Learning Runs

To comprehend how the proposed metrics evolve during a run, they were computed and graphed as functions of the step in the iterative Reinforcement Learning (RL) process during which the molecules were generated.

To depict molecular diversity, the diversity metric was calculated for each step as per Equation 3.1, where the set of molecules consisted of all molecules generated at that step. Further, the number of unique molecules at each step was determined in comparison with the set of molecules produced in the previous step (in other words,

the intersection of the two sets). This approach offers insights into whether the newly generated molecules are structurally similar to those produced most recently.

To illustrate shifts in molecular distribution, these shifts were computed for each step using Equation 3.3, comparing the current step to its predecessor.

To portray the diversity and distribution of molecules sampled from a specific run, a numerical feature representation of the molecules was generated. Subsequently, dimensionality reduction was applied to these features, resulting in a scatter plot. Experiments were conducted with CDDD and ECFP representations, combined with UMAP and PCA reductions. Both UMAP and PCA were pre-trained on the first 100 000 molecules from a ChEMBL [26] dataset to produce reproducible results and enhance performance. The specific dataset can be found on GitHub [27]. Both UMAP and PCA were implemented with their default values, according to the implementations provided by scikit-learn and the authors of UMAP [25], [28]. In the scatter plot, points were colored according to a user-selected feature to enhance plot interpretability. Additionally, a filter function was incorporated, enabling users to visualize molecules within a specified range of scores or steps.

To facilitate the use and access of these visualizations, a web application was developed using Streamlit [29]. As new molecules are generated during a run, the application automatically updates the data and displays the plots. Additionally, drop-down menus and sliders were incorporated into the application, enabling users to adjust various settings of the scatter chart, such as the feature by which points are color-coded.

3.2 Adding human feedback to REINVENT

To incorporate human feedback into a REINVENT run, a HITL scoring component was introduced as an optional component to the MPO. The HITL scoring component consists of a user preference model trained in an online learning setting during a run for predicting user preference. The user preference model is trained on a subset of the generated molecules, that are presented and rated by the user iteratively during the run. The HITL scoring component then scores molecules based on the predicted user ratings. An overview of the RL system utilizing HITL is shown in figure 3.1.

3.2.1 Learning user preference with Online Learning

The proposed system incorporates an online learning approach that allows the model to adapt to the new data and continually improve the accuracy of user preference prediction. The prediction task is defined as a binary classification task, with the training data labeled 1 for a liked molecule and 0 for a disliked molecule. Specifically, the goal is to predict the user rating of a molecule based on a set of previously scored molecules, and the probability of the molecule being "liked" is the score provided to the MPO.

This project evaluated several machine learning models for the online learning model: Random Forest, Logistic Regression, and K-nearest neighbors. Each of these models

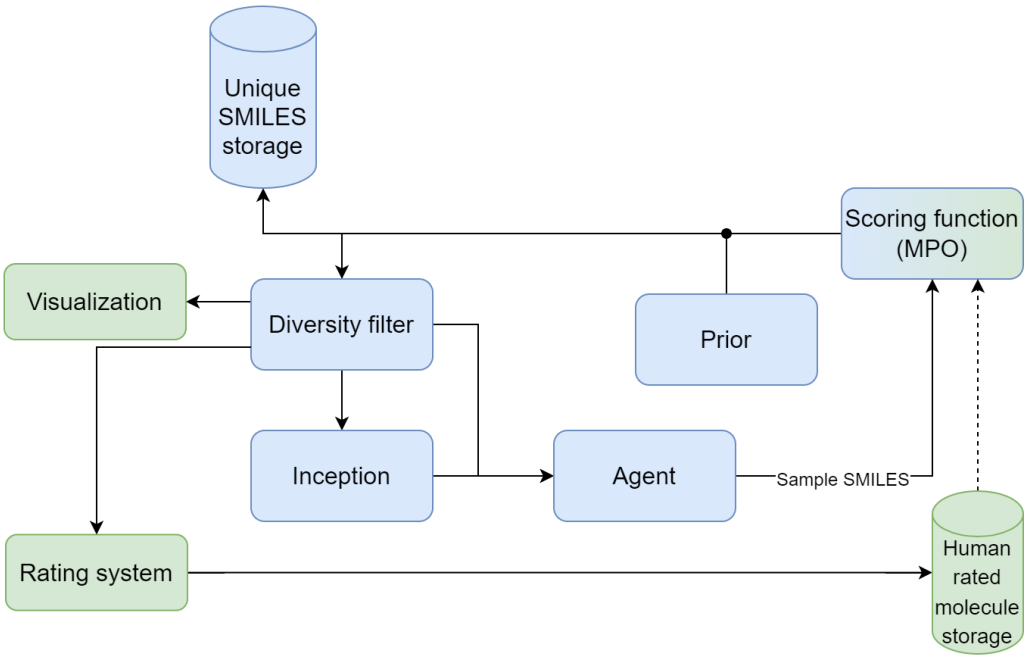

Figure 3.1: Integration of HITL Scoring Component in the REINVENT RL System
An overview of the REINVENT RL system with the suggested HITL scoring component was added. The existing components are colored in blue, while the new ones are colored in green. The existing components are explained in section 2.2. Note that the MPO is colored partially green since it will have new functionality added. In the diagram, a human would interact with the visualization-and rating system components. The rated molecules are stored with their ratings, which are then used to update the HITL scoring component.

takes SMILES as input and uses the user's rating as the target label. These models were implemented using the default hyperparameters in scikit-learn's library [28], and utilized either CDDD or ECFP to convert the SMILES into numerical features. As none of these models possess inherent retraining capabilities, the HITL scoring component was designed to store all training data instead. Whenever new rated molecules are added to the training data, the model is simply retrained from scratch.

3.3 Creating Surrogate User models

In order to design an efficient HITL protocol in light of time constraints and difficulties in collecting enough sample data from real users, this project used surrogate user models to evaluate the HITL-scoring component, and the overall system's ability to learn from feedback with different levels of complexity. The surrogate user models were designed to be orthogonal to existing scoring components in order to mimic a user having difficulty capturing their preference using the current scoring systems. To achieve this, more complex surrogate user models were trained on the BACE dataset [30], specifically on the pIC50, which measures inhibitory activity against the BACE enzyme. These values were thresholded around the median to

create a balanced binary classification task on the first 10000 molecules of ChemBL dataset [26]. Different surrogate user models were created once SMILES inputs were converted to either ECFP and CDDD embeddings. In total, four surrogate user models were created for evaluation: one Rule-based, two random forest models (one using ECFP and the other using CDDD), and one neural network-based model. The predictive performance along with their class balance is presented in table 3.1.

3.3.1 Rule-based user model

The Rule-based user model rates molecules based on simple structure-related criteria that do not directly interfere with the other scoring components used. The purpose of this user model is to verify whether the proposed HITL system can adapt to clear signals, and it was therefore intentionally simplified. Specifically, the model prefers a molecule if it contains two or more Nitrogen atoms; otherwise, it dislikes the molecule [31]. Nitrogen was specifically selected because it is commonly present in pharmaceutical drugs. Furthermore, the threshold of two atoms was set to provide a balanced class distribution in the evaluation set. This balance was based on a subset of ChEMBL data available on GitHub [26], [27].

3.3.2 Random forest-based user models

To train random forest-based user models on the BACE dataset [30], SMILES input was converted into numerical vectors using either ECFPs or CDDD. Two RF models were developed for this project, one using ECFP-converted inputs and the other using CDDD-converted inputs. The random forest classifier model from the scikit-learn [28] library was used to train on the data after partitioning to 80 percent for training and 20 percent for testing.

3.3.3 Neural Network-based user model

The Neural Network based user model has a neural network architecture with three linear layers of 100 neurons each and an output layer with one neuron. The model was trained on the BACE dataset [30] using the stochastic gradient descent (SGD) optimizer and the Binary Cross Entropy with Logits loss (BCEwithLogitsLoss) function [32]. Before training, data was partitioned to 80 percent for training and 20 percent for testing and molecules in SMILE formats were converted to ECFP fingerprints. The trained user model was used to predict the molecules as either human-liked or not, and the decision threshold was determined by evaluating its performance on a subset of the ChemBL dataset [26], [27]. The threshold was chosen such that the model predicts approximately 50 percent of the molecules in the subset of the ChemBL dataset as human-liked. This threshold allows for a balanced rating system and ensures that the model is effective in predicting user preferences.

User model	Tested accuracy	Class balance on ChEMBL
Rule based	-	0.53
Random forest (ECFP) based	0.82	0.58
Random forest (CDDD) based	0.87	0.49
Neural Network based	0.92	0.50

Table 3.1: The table presents a comparison of the four surrogate user models based on their accuracy on their training task. The tested accuracy measures include the performance of the pre-trained model on its own test dataset, and validation on a subset of the ChEMBL dataset includes accuracy used to determine the optimal threshold. Note that the Rule-based surrogate user model does not have a score, as it is not trained on a dataset, instead rating molecules based on predefined rules.

3.4 Testing Methodology

This study utilized both surrogate user models and human participants for testing. This section describes the methodology applied during several experiments with surrogate user models, as well as an experiment conducted with human participants.

3.4.1 Testing with user models

To evaluate the effectiveness of the HITL system, several experiments were conducted, focusing on the performance of a REINVENT RL run using the HITL scoring component. Three key metrics were assessed: mean user (or surrogate user model) favorability, the total score of the non-HITL-scoring components, and molecular diversity.

Three experiments were conducted with surrogate user models to test three separate aspects of the proposed HITL system: the number of rated molecules, the rating frequency, and the technique for selecting which subset of molecules should be presented to the user for rating. In this process, the experiments were conducted sequentially, with the choice of configurations for the second and third experiments being determined after analyzing the results of the first experiment.

For these three, a trial is defined as executing a REINVENT RL run with a specific configuration, then sampling 10,000 molecules from the resulting RL agent. Various metrics are then computed on the sampled molecules.

During the RL run, the run was paused every N steps. In every pause, the surrogate user model was prompted to rate M molecules from the last N steps. To select what molecules should be rated from the steps, a few different selection strategies were tested. The HITL-scoring component was then updated with the newly rated molecules before the run was resumed.

The rating period N and the number of rated molecules M were configurable values, along with the type of selection strategy, user preference model, and surrogate user model, all defined in each trial configuration. Experiments 1-3 subsequently tested and measured the system with different combinations of these parameters.

Three metrics were calculated for the 10 000 sampled molecules. The user favorability was defined as the mean surrogate user model rating (using the same user model as during the RL run). Diversity was calculated according to equation 3.1.

The non-HITL scoring components incorporated in the MPO included QED, molecular weight between 300-600, 1-4 rings, and fewer than 4 hydrogen bond donors. Each of these scoring components was transformed to a score range between 0 and 1. The individual scores were then combined using a geometric mean to derive the total score. For the specific configuration, including scoring transformations, please refer to Appendix A.1. The starting agent used was based on ChEMNBL and can be found at [33].

Each configuration underwent 200 RL steps with feedback from the surrogate user model, followed by an additional 50 steps without updating the HITL component, allowing the agent steps to optimize for the final version of the user preference models.

3.4.1.1 Experiment 1: Testing User Preference Models

Experiment 1 examined the performance of various online learning models and the number of molecules rated by the user. A configuration was created for each combination of the surrogate user model, online learning model, and 3, 5, and 10 rated molecules per rating period. The rating period was every 10 steps, and the selection strategy was a random selection. Three trials were completed for each configuration to account for variability. Baseline trials were also performed, in which no HITL scoring component was included in the MPO.

3.4.1.2 Experiment 2: Testing Rating Frequency

Experiment 2 explored the impact of rating frequency on the performance of the HITL system while maintaining a constant total number of ratings. This investigation specifically focused on altering the frequency of rating molecules while keeping the total number of calls consistent.

The tested configurations were 5, 10, and 20 for the rating period, with the number of rated molecules being equal to the period to maintain constant total ratings. Each rating period was tested with every combination of online learning models using CDDD as features and surrogate user models. The selection strategy employed was random selection.

3.4.1.3 Experiment 3: Testing Selection Techniques

Experiment 3 tested various techniques for selecting molecules to be rated by user models. Three selection techniques were evaluated: random selection, selection based on uncertainty, and selection based on score. For the uncertainty-based and score-based selections, both a greedy and a probabilistic method were tested.

We examined each selection technique using the CDDD-based logistic regression and random forests user preference models, and also with every surrogate user model.

The number of molecules rated during each intermission was 5, and intermissions occurred every 10 steps.

The specific details of the selection strategies are presented below. Note that the set of molecules in these cases consists of all the molecules from the previous 10 steps.

Score-based Selection

Molecules were selected from a set based on their scores from different scoring components in the MPO, excluding the score from the HITL component. Specifically, to select the molecules, the total score of the MPO, excluding the HITL component, was calculated for each molecule i using the formula:

$$S_i = \left(\frac{(TS_i^k + 10^{-6})}{(HITL_i + 10^{-6})} \right)^{\frac{1}{k-1}} \tag{3.4}$$

Here, k is the number of scoring components, $HITL_i$ is the HITL score for the i^{th} molecule, TS_i is the total score including the HITL scoring component, and 10^{-6} is a small constant added to avoid division by zero.

For the greedy selection, the n molecules with the highest S_i were chosen. For the probability-based selection, each molecule was instead assigned a selection probability P_i, which was determined by normalizing the calculated scores. This normalization ensured that the sum of all probabilities equaled 1, allowing them to be used in a random selection process:

$$P_i = \frac{S_i}{\sum_{j=1}^{N} S_j} \tag{3.5}$$

Here, N is the total number of molecules in the set. Finally, n molecules were selected from the set, with the selection probability of each molecule proportional to its assigned probability. This method created a bias towards molecules with higher scores, while still allowing for some degree of randomness in the selection process.

Uncertainty-based Selection

In the uncertainty-based selection strategy, each molecule was evaluated based on the absolute difference between its HITL score and a reference value of 0.5. This reference value was chosen because it represents an equal probability of being liked or disliked, effectively measuring the uncertainty or ambiguity of each molecule's score. The absolute differences were calculated for each molecule i using the formula:

$$S_i = |HITL_i - 0.5| \tag{3.6}$$

where $HITL_i$ is the HITL score for the i^{th} molecule. This process produced a set of scores S_i, each representing the level of uncertainty associated with the respective molecule's HITL score.

Next, the scores were normalized to obtain a set of selection probabilities P_i for each molecule. The normalization was done by dividing each score by the sum of all scores, ensuring that the total of all probabilities equals 1:

$$P_i = \frac{S_i}{\sum_{j=1}^{N} S_j} \tag{3.7}$$

Here, N is the total number of molecules in the set. If the sum of all scores was zero (indicating that all molecules had a HITL score of 0.5), all molecules were assigned an equal probability.

Finally, n molecules were selected from the set. When using greedy selection, the n molecules with the highest selection probabilities were chosen directly. This method favored molecules with the highest levels of uncertainty.

The probability-based instead performed a random selection process, where the selection probability of each molecule was proportional to its assigned probability. This method still favored molecules with higher levels of uncertainty but also allowed for a degree of randomness in the selection process.

3.4.2 Testing with real users

To evaluate the performance of the HITL system when real user feedback is used to guide the system during training, we conducted experiments with 8 volunteers from AstraZeneca, who has expertise in either computational or synthetic chemistry. These runs were constructed similarly to those in Section 3.4.1, but with the participants replacing the role of the surrogate user models.

In these runs, as illustrated in Figure 3.2, every user was presented with a list of molecules and asked to indicate their preference by expressing whether they liked or disliked each molecule. They were asked to rate the molecules based on their perception of what qualifies as a valuable suggestion from an AI system. Additional information on the instructions given during experiments is attached in Appendix 1.

Each of these experiments underwent 140 RL steps with user feedback, followed by an additional 60 RL steps performed without updating the HITL component. This second phase allowed the agent to train on the final version of the user preference model. Note that the scoring components deployed in these experiments were slightly different, also including a Customer Alerts component. The full configuration is found in the Appendix **??**.

In each experiment, the RL run was paused every 20 steps and at each pause, the user was asked to rate 20 molecules selected from the previous 20 steps. The selection method used was probability-based selection with a bias towards high-scoring molecules, as described in Section 3.4.1.3. The HITL scoring component employed a Random Forest with CDDD-based user preference model. Like the previous experiments, the HITL scoring component was updated each time a user

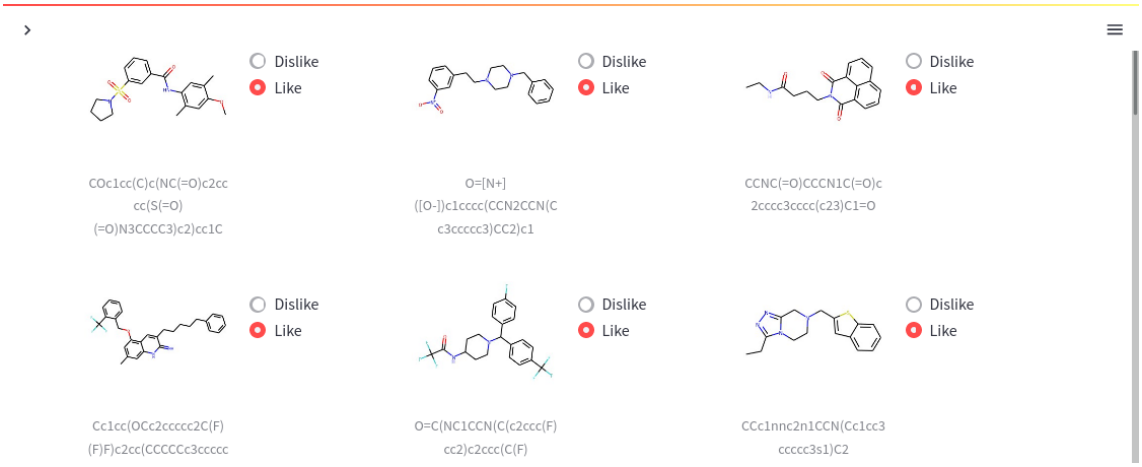

Figure 3.2: Dashboard presenting the molecules to the user for feedback during user experiments. The dashboard displays both the structures of the molecules and also their corresponding SMILES strings.

submitted newly rated molecules. Throughout the course of these experiments, each user was prompted seven times to rate a total of 140 molecules.

To evaluate the impact of the HITL component in the runs, an A/B test was conducted. This test involved providing each participant with an additional 50 molecules to rate after the initial run. The set of 50 molecules was a shuffled mix of 25 molecules each from a baseline run and a HITL run. Each set of 25 molecules was generated by sampling 10,000 molecules from the agent resulting from each run. From the 1,000 highest-scoring sampled molecules, a selection of 25 diverse molecules was made using RDKit's MaxMinPicker, with the Tanimoto distance serving as the distance metric [34]. These molecules were delivered to the participants a few days after the original test, in the form of an Excel sheet that included images of the molecules, their SMILES strings, and a field for rating each molecule as "Like" or "Dislike".

To evaluate the findings, a one-tailed statistical test was performed for both each individual participant's test as well as for the group as a whole. The null hypothesis is defined as: "There is no increase in user favorability when employing the HITL system compared to the baseline." Rejection of the null hypothesis would lead to the conclusion that "There is an increase in user favorability when employing the HITL system compared to the baseline."

4

Results

This chapter presents the results derived from user testing of the Human-in-the-loop (HITL) scoring component of the de novo drug discovery system. The user tests are aimed at evaluating the efficacy of the user preference models in capturing user preferences, and gauging the performance enhancement of the HITL component in generating molecules preferred by the users. We also investigate the extent of correlation among user preferences. In addition to individual results, a comparative analysis with a baseline run using Neural Network surrogate users is carried out. The results of these experiments offer valuable insights into the potential of incorporating user feedback into the molecular design process.

4.1 Visualization

In the interest of computational efficiency, the efficacy of using Continuous Data-Driven Descriptors (CDDD) as a diversity metric was investigated. It was compared with the Tanimoto distances on ECFP, a standard approach within cheminformatics [35]. For this comparison, pairs of molecules were created from two sets, each containing 20,000 molecules. For each molecule in the first set, the top five most similar molecules were selected from the second set based on Tanimoto similarity, resulting in a total of 100,000 pairs. The two sets of 20,000 molecules were selected from the first 20,000 and 20,000-40,000 molecules from a small dataset from ChEMBL [26], available on GitHub [27]. The top five most similar pairs were chosen as the fraction of similar molecules was very low when using random pairs. Figure 4.1 displays a scatter chart comparing the Tanimoto and Euclidean CDDD distances between the 100,000 molecular pairs and a line fitted to minimize the L2 distance. Moreover, the Pearson correlation between the data points was calculated, yielding a correlation score of 0.615 suggesting a moderate positive correlation.

Figure 4.1 displays a scatter chart of distances between pairs of molecules, with the x-axis representing the Tanimoto distance, and the y-axis denoting the Euclidean distance between CDDD embeddings. The data plotted is a subset from ChEMBL, specifically, the first 40 000 molecules were split in half, creating two sets of 20 000. For each molecule in the first set (containing the first 20 000 molecules), the top 5 most similar molecules in the second set, in terms of Tanimoto similarity, were used as pairs. This was done as the diversity in the dataset was generally high, otherwise resulting in a very low fraction of molecular pairs with a high Tanimoto distance

being represented. Based on these pairs,

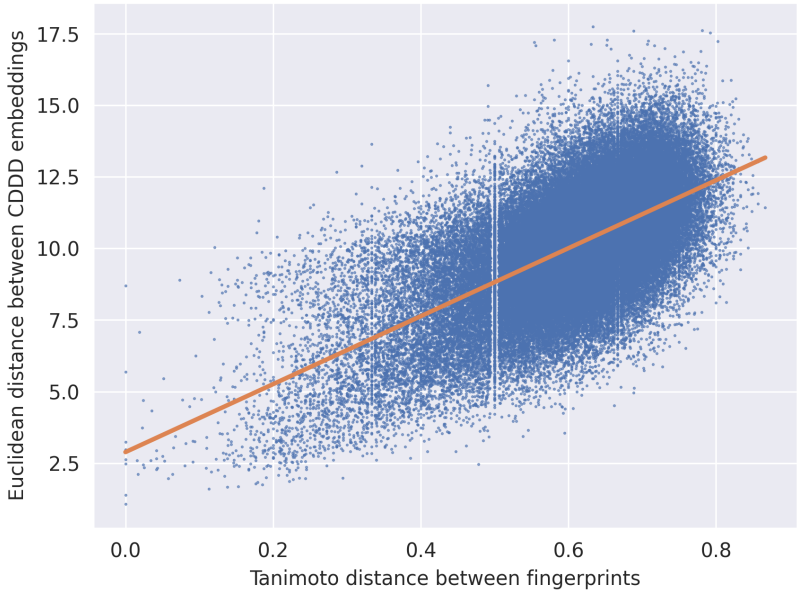

Figure 4.1: Scatterplot comparing the Tanimoto distances and Euclidean CDDD distances between 100,000 molecular pairs. The orange line represents a linear fit that minimizes the L2 distance

Figure 4.2 showcases scatter plots generated as described in Section 3.1.2, where various techniques were employed for dimensionality reduction and numerical feature representation. Distinct differences can be observed among these charts. Notably, plots using CDDD appear to provide more coherent x and y-axis representations of the Total Score and Molecular Weight, as indicated by the smooth color distribution. Moreover, PCA as a dimensionality reduction technique seems to generate larger and more evenly distributed clusters of points, which are more informative for differentiating between molecules. In contrast, UMAP tends to form denser clusters than PCA, with some molecules dispersed outside the primary cluster.

Figure 4.3 presents a screen capture of the visualization dashboard. As illustrated in panel 1, the baseline run features a region of high-scoring molecules centralized on the x-axis, with a higher presence of low-scoring molecules for large negative values. Panel 2 reveals a steady decrease in molecular diversity throughout the run, albeit with a slight deceleration toward the end. Panel 3 indicates that the quantity of unique molecules starts to decline approximately halfway through the run, while the number of valid molecules remains constant. In panel 4, a decrease in distribution shift is observable after the initial steps, following which it maintains a relatively constant state, both for step-by-step comparison and for the comparison of 5-step means.

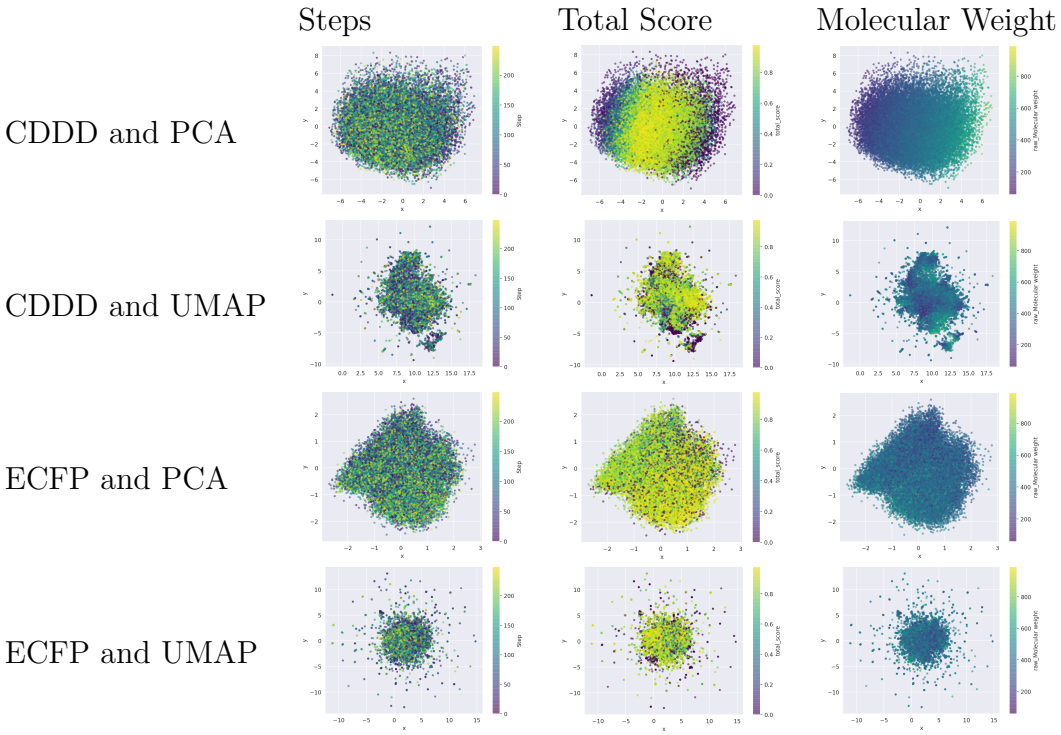

Figure 4.2: Plots of a baseline run from Experiment 1 (Section 3.4.1.1), illustrating the results of various combinations of features and dimensionality reduction techniques. In each column, the points are colored by either step, total score, or molecular weight, respectively. In each row, plots were generated with the same set of features and dimensionality reduction technique. The text in the figures is not of interest, as they are just a part from the

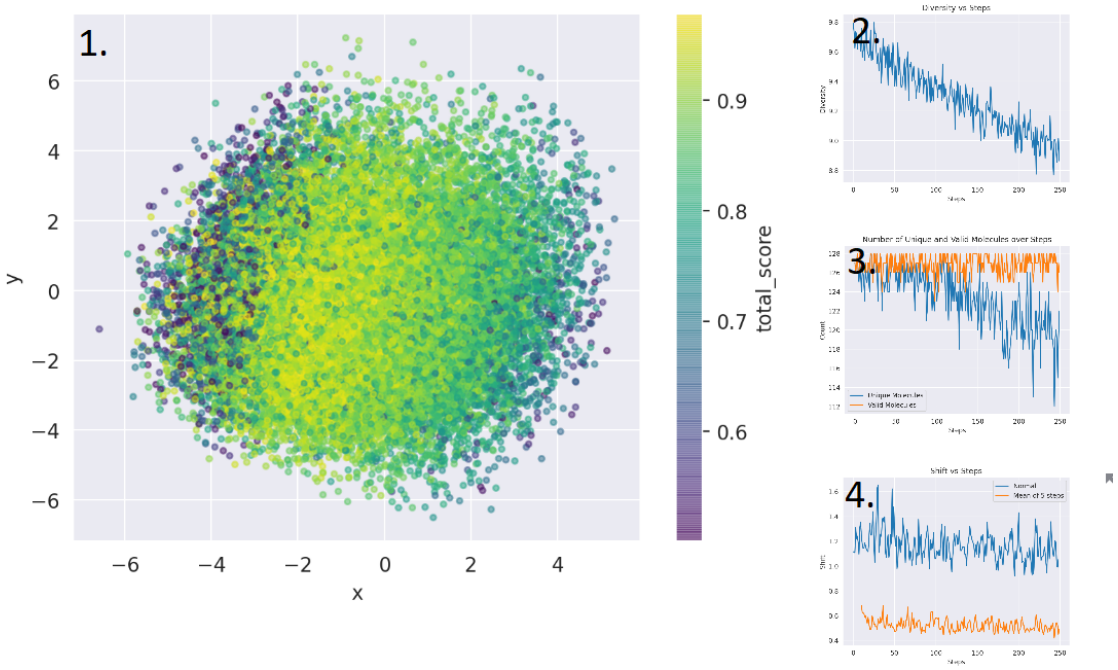

Figure 4.3: Screen capture of the visualization dashboard showcasing a baseline run from Experiment 1 (Section 3.4.1.1). The dashboard comprises four distinct panels, labeled 1-4 in the Figure, all of which are detailed in subsection 3.1.2: Panel 1 contains the scatter plot; panel 2 illustrates the diversity at each step; panel 3 presents the count of valid and unique molecules at each step; and panel 4 captures the shift in distribution.

4.2 Experiments with User Models

This section presents the results and insights obtained from testing REINVENT RL runs with the HITL scoring component, employing different surrogate user models.

4.2.1 Experiment 1 Results

Experiment 1 (3.4.1.1), where we investigated the performance of various user preference models and varied the number of molecules rated by the user, demonstrated an improvement in the average user model favorability for all user preference models compared to the baseline, where no online learning model was used (Figure 4.4). The most significant increase in user model favorability across all configurations came from the logistic regression model with CDDD input features, closely followed by the Random Forest CDDD user model. Both the K-nearest neighbors CDDD and the Random Forest CDDD user preference models exhibited similar performance regardless of the number of user ratings used, with the resulting user favorabilities often being within experimental error of each other (Figure 4.4). In spite of its poor performance compared to other models, the user favorability for Random Forest with ECFP models was better than the baseline. Generally, the improvements in user favorability were positively correlated with the number of ratings, as demonstrated in Figure 4.4. On the other hand, user favorability was highest for user preference models with Rule-based surrogate user models than the other nonlinear surrogate user models.

Furthermore, Experiment 1 revealed that the inclusion of the HITL scoring component did not significantly impede the convergence of other scoring components, as the average total scores of the sampled molecules from HITL-included trials were comparable to those obtained in the baseline trials (Figure 4.4). In addition, despite certain trials exhibiting a decrease in the mean total score, the most significant decline was observed in the utilizing the Random Forest user model with CDDD, particularly in the case of K-nearest neighbors with a rating frequency of 10 ratings per 10 epochs. Nevertheless, it is worth noting that the observed decrease was minimal, amounting to less than 0.05 compared to the baseline configuration (Figure 4.4). As illustrated in Figure 4.4, the standard deviation of the mean total score between trials for some configurations was higher compared to the baseline trials, but it remained within a reasonable range of a few hundredths. Similar to user favorability, the mean score was comparatively higher for user preference models with Rule-based surrogate user models than the other nonlinear surrogate user models.

Based on the results shown in Figure 4.4, it is evident that there were no significant variations in diversity among different user preference models when the number of rated molecules was varied. However, the user preference models employing the Rule-based user model demonstrated significantly lower performance compared to other , particularly those utilizing Logistic Regression, K-nearest neighbors, and Random Forest with CDDD input features. In contrast to user favorability, user preference models with nonlinear surrogate user models demonstrated better diversity compared to user preference models with Rule-based surrogate user models.

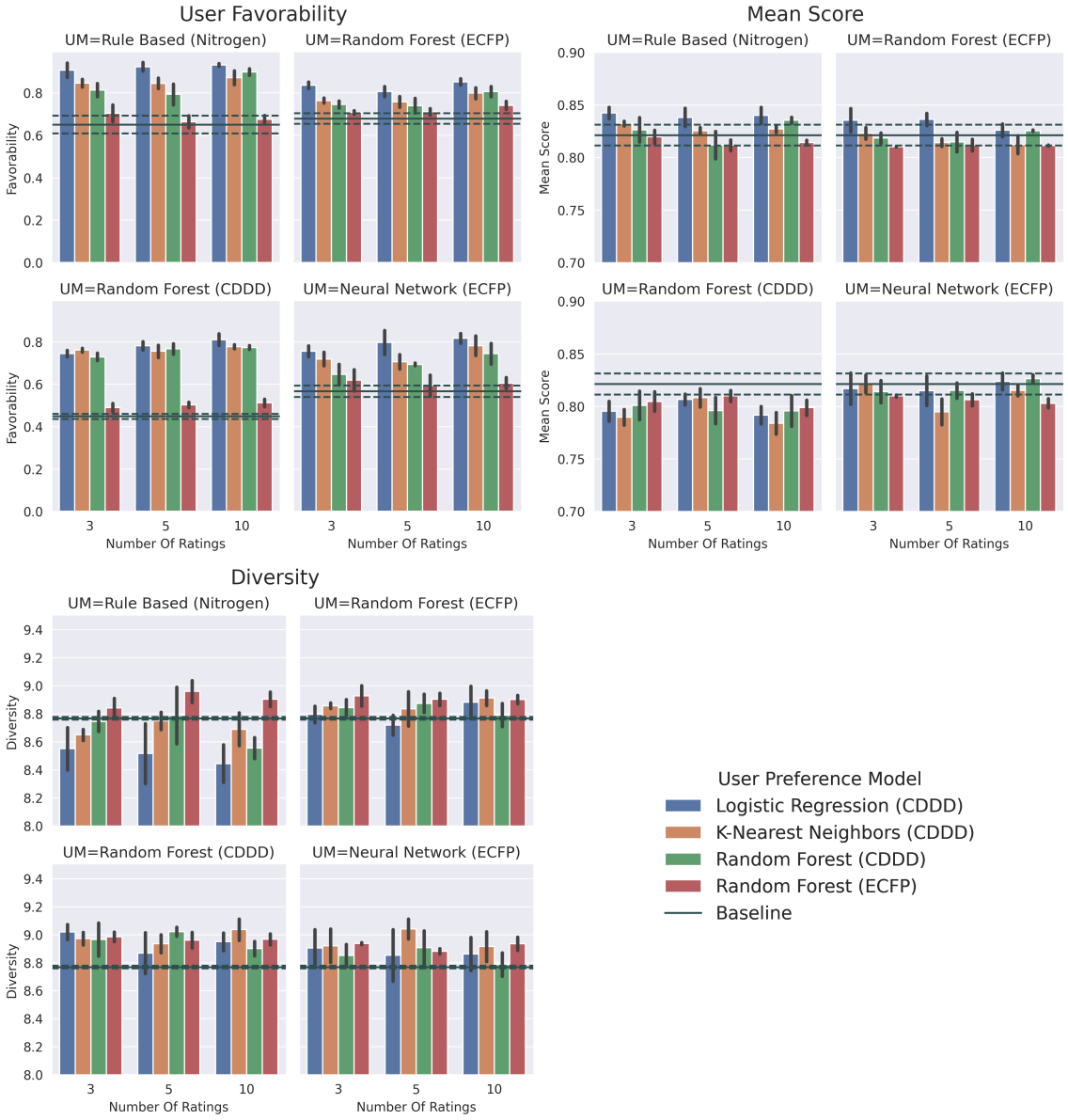

Figure 4.4: User favorability, mean score, and diversity of molecules from Experiment 1 models. In the first plot, the y-axis depicts the mean ratings of user models obtained from the sampled data, the second plot's y-axis depicts the total score of the MPO without the HITL scoring component on the sampled data, the y-axis in the third plot indicates the mean diversity of sampled data. The x-axis in all three plots represents the number of rated molecules. In each of the three plots, the bars denote the average of mean ratings across three trials, with error bars indicating the standard deviation. Each subplot contains the results using a specific surrogate user model. Horizontal lines signify the mean and standard deviation for the baseline runs, where no online learning model was employed.

4.2.2 Experiment 2 Results

Experiment 2 (3.4.1.2) demonstrated that there are no significant changes in terms of user favorability for all user preference models when altering the rating period at intervals of 5, 10, and 20. However, there is a significant improvement in the average user model favorability for most of the user preference models compared to the baseline, where no online learning model was employed (Figure 4.5). As Like in Experiment 1, the most significant increase in user model favorability across all configurations came from the logistic regression model with CDDD, closely followed by the Random Forest CDDD user model and K-nearest neighbors CDDD (Figure 4.5) and then by the Random Forest model with ECFP input features Random Forest model with ECFP mostly performed similarly to the baseline run in most of the configurations. Also, user favorability was highest for user preference models with Rule-based surrogate user models than the other nonlinear surrogate user models.

Relatively, Experiment 2 also revealed that the inclusion of the HITL scoring component did not significantly impede the convergence of other scoring components when the rating frequency is altered 4.4. The most significant decline in the mean total score was all the utilizing the Random Forest surrogate user model with CDDD. As illustrated in Figure 4.4, the standard deviation of the mean total score between trials for some configurations was higher compared to the baseline trials. In addition, there was no significant difference from the results of Experiment 1 when analyzing the mean score of different surrogate user models.

Furthermore, Experiment 2 also revealed that there were no significant variations in diversity among different user preference models when the rating frequency was altered. However, similar to Experiment 1, the employing the Rule-based user model demonstrated significantly lower performance compared to other , especially those utilizing Logistic Regression, K-nearest neighbors, and Random Forest with CDDD input features.

4.2.3 Experiment 3 Results

Experiment 3 (3.4.1.3), where we investigated the performance of two different user preference models using random forests and logistic regression by altering the selection strategy, demonstrated that there are no significant changes in terms of user favorability for both user preference models (Figure 4.6). Despite exhibiting relatively smaller variations when altering the selection strategy, the logistic regression model showed the most significant increase in user model favorability while using random selection. In addition, the Rule-based user model demonstrated the most significant increase in user model favorability across all the other .

In terms of mean score, Experiment 3 also not observed any significant convergence of other scoring components when the selection strategy is altered, as the average total scores of the sampled molecules from HITL trials were comparable to those obtained in the baseline trials, which are random selection in Experiment 3 (Figure 4.6). In addition, despite certain trials exhibiting a decrease in the mean total score, the most significant decline was observed in all the utilizing the Random Forest user

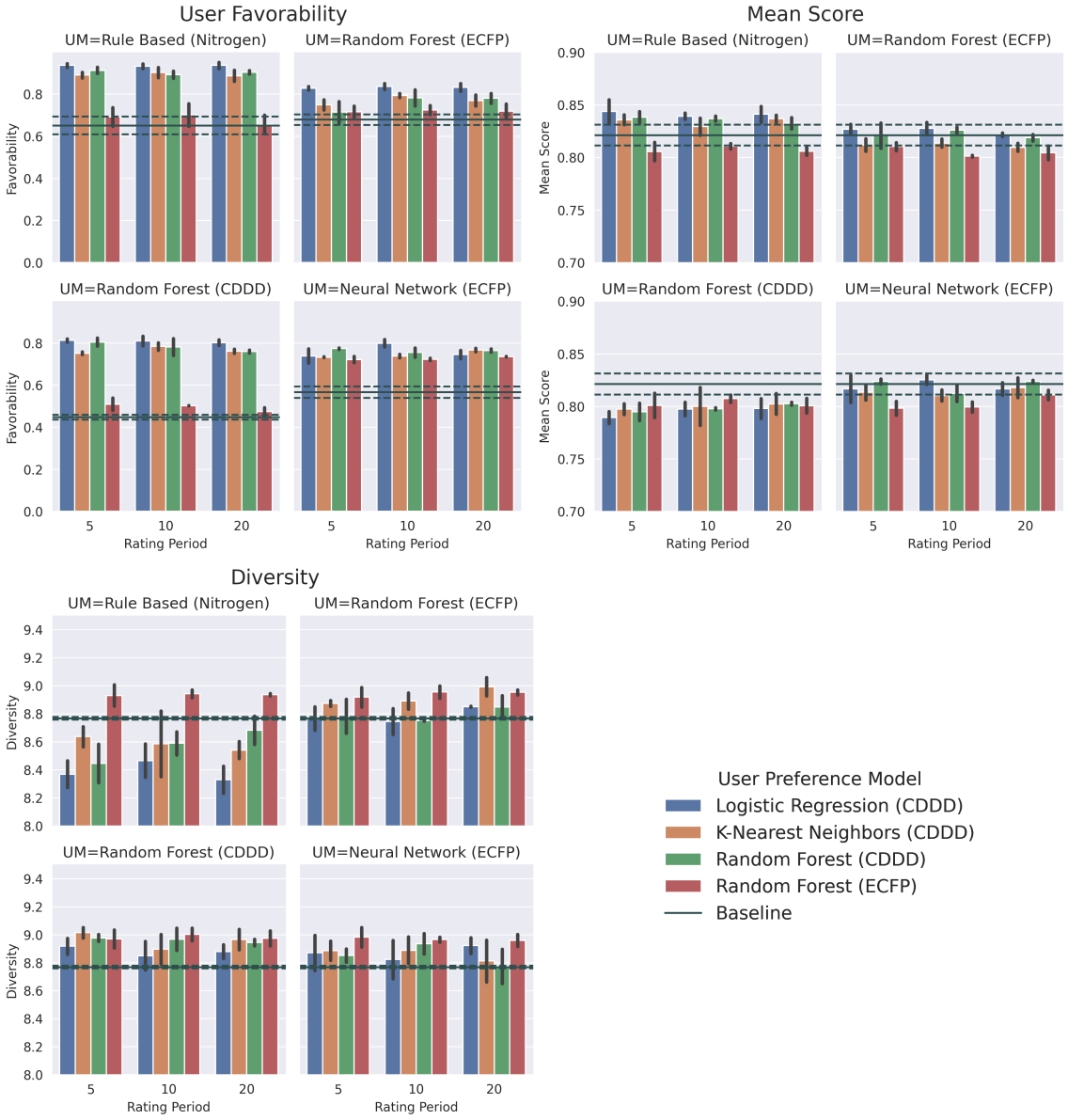

Figure 4.5: User favorability, mean score, and diversity of molecules from Experiment 2 models. In the first plot, the y-axis depicts the mean ratings of user models obtained from the sampled data, the second plot's y-axis depicts the total score of the MPO without the HITL scoring component on the sampled data, the y-axis in the third plot indicates the mean diversity of sampled data. The x-axis in all three plots represents the frequency of rating molecules. In each of the three plots, the bars denote the average of mean ratings across three trials, with error bars indicating the standard deviation. Each subplot contains the results using a specific surrogate user model. The horizontal line signifies the mean and standard deviation for the baseline runs, wherein no online learning model was employed.

model with CDDD when greedy selections were employed.

Furthermore, similar to Experiments 1 and 2, Experiment 3 also revealed that the employing the Rule-based user model demonstrated comparatively lower performance across all the selection strategies. In addition, there is not much variation in diversity among the different user preference models when the selection strategy is altered. Despite less variation in the diversity among different , the most significant increase was observed in both the utilizing the Random Forest user model with CDDD, especially when a high-scoring greedy strategy was employed.

4.3 User testing

This section presents the results and key findings from testing the HITL scoring component with human participants, as detailed in Section 3.4.2.

4.3.1 Evaluating User Preference Models

The initial aspect of our user testing entailed assessing how accurately the user preference models represented the preferences of human users. This was undertaken by examining both the evolution of the user preference models' performance during the run and the predictive ability of the final versions of the user preference models on the validation set.

The Binary Cross Entropy (BCE) was employed as a metric to evaluate the capability of the user preference models to optimize for user preference over time. The BCE was calculated by comparing the predicted probability of a molecule being liked by a user, as determined by the user preference model, with the actual user preference record from the user. Note that the probability is predicted before the online learning model is updated with the rated molecules as training data. This measure essentially captures the discrepancy between the predicted and actual preferences, with lower values indicating better performance.

Figure 4.7 illustrates the evolution of the BCE for each intervention, i.e., every instance where the online learning model is retrained with an additional set of rated molecules. An overall trend of decreasing BCE with an increasing number of interventions is observed, suggesting an improvement in the performance of the user preference models with incremental user feedback. Furthermore, the user preference models from the human tests exhibit a lower BCE compared to those from the Neural Network surrogate user runs. This suggests that on average, the human users were easier to learn from than the Neural Network surrogate models.

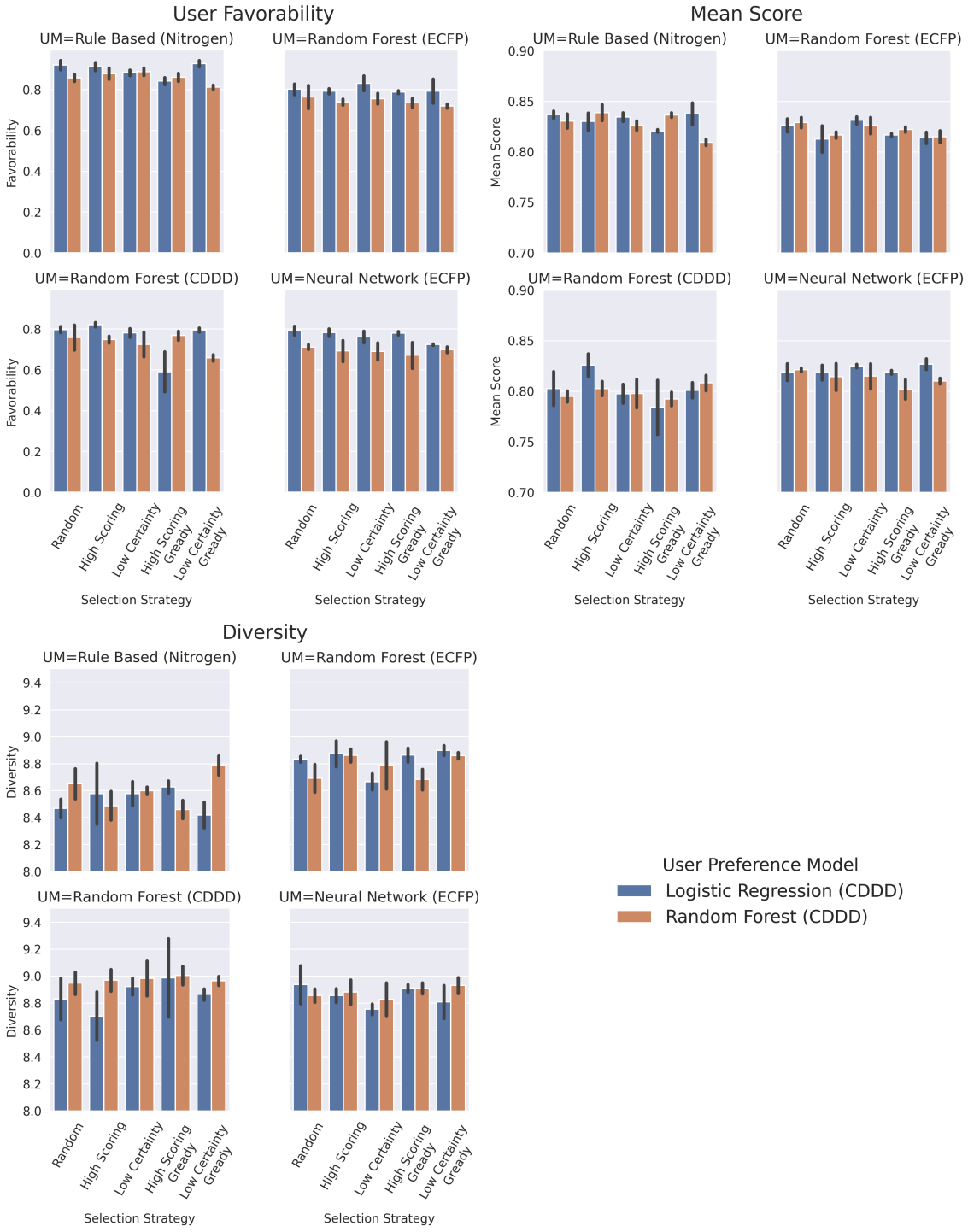

Figure 4.6: User favorability, mean score, and diversity of molecules from the various models explored in Experiment 3. In the first plot, the y-axis depicts the mean ratings of user models obtained from the sampled data, the second plot's y-axis depicts the total score of the MPO without the HITL scoring component on the sampled data, the y-axis in the third plot indicates the mean diversity of sampled data. The x-axis in all three plots represents the selection strategy employed. In each of the three plots, the bars denote the average of mean ratings across three trials, with error bars indicating the standard deviation. Each subplot contains the results using a specific surrogate user model.

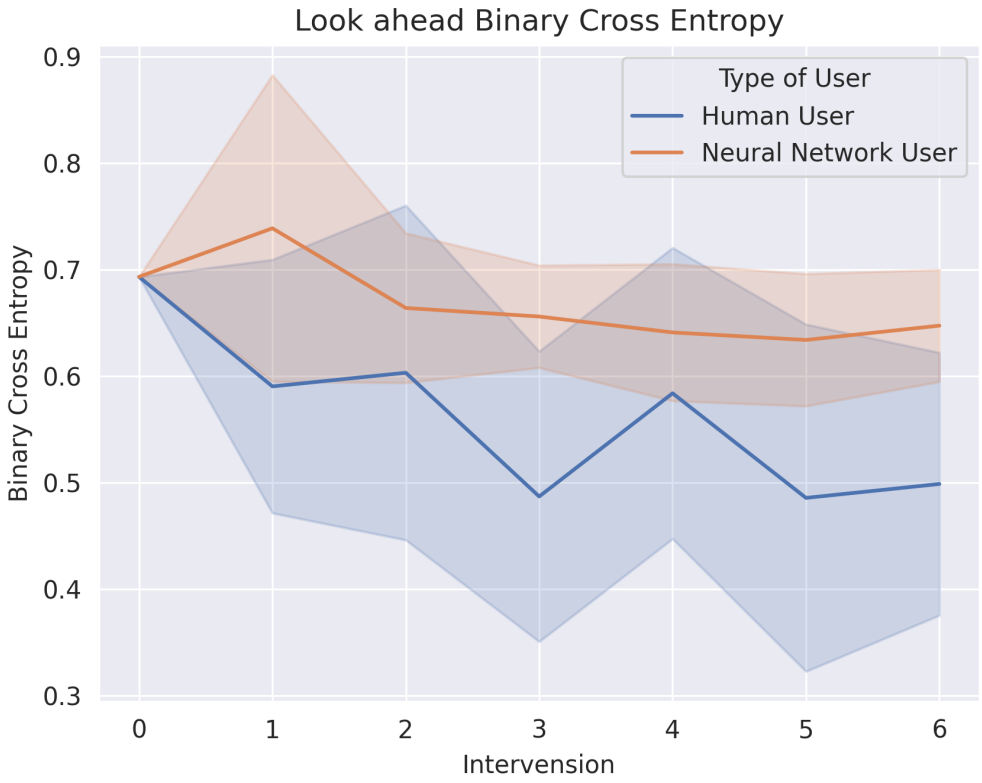

Figure 4.7: Evolution of look-ahead BCE for user preference models across 8 human user runs and 8 neural network user runs. The BCE is calculated between the online model's prediction for rated molecules, prior to retraining the model with these molecules. The y-axis depicts the average BCE for runs with Human users and Neural Network users, with the shaded areas representing the standard deviation. The x-axis is the number of interventions performed, i.e., the number of times the Preference model has been updated with more rated molecules.

The predictive capabilities of the user preference models were further explored by examining the F1 scores on the rated molecules from the validation set. These F1 scores reflect the models' ability to accurately predict human user preferences, with a higher F1 score indicating better predictive performance. In this analysis, F1 scores were computed for two different sets of validation data - one from the HITL run and another from the baseline run.

Figure 4.8 displays the F1 scores for each participant across both sets of validation data. From these plots, it is observed that an F1 score above 0.75 was attained for five out of seven runs from the HITL sets and for four out of the seven users on the baseline set.

Moreover, the F1 scores on the HITL set were generally marginally higher than those from the baseline set, with participant four seeing a substantial increase. This suggests that the user preference models were more successful in predicting user preferences for molecules that were generated with the assistance of the HITL com-

ponent.

Note that for all the cases with an F1 score of zero the User Preference Models rated all molecules as "Disliked".

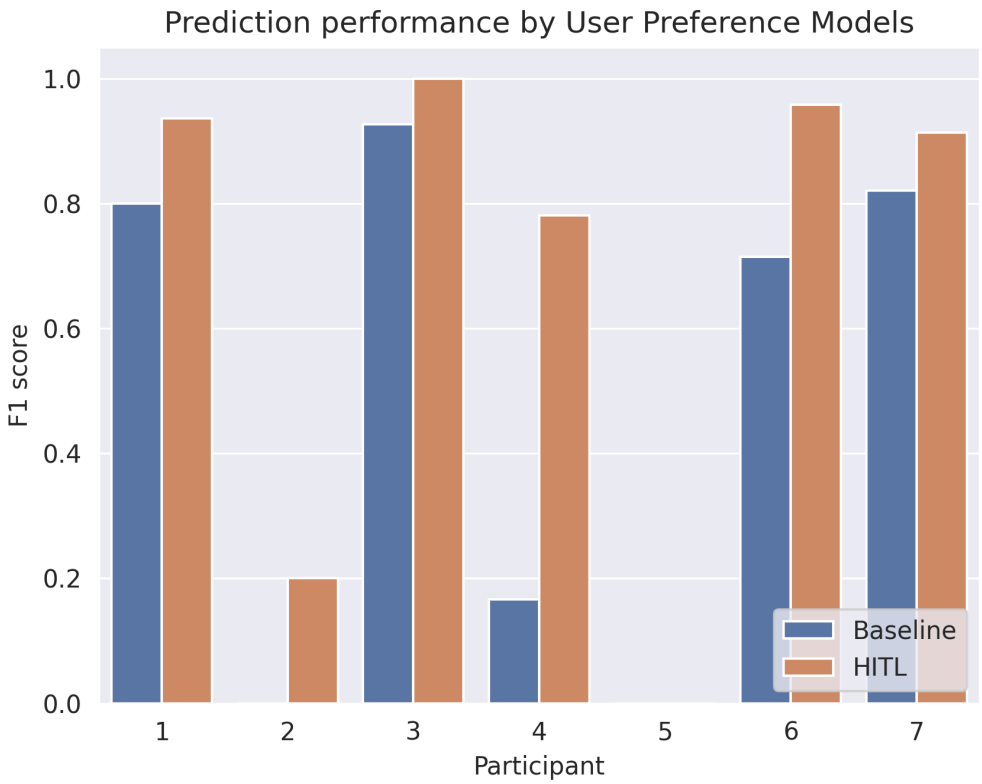

Figure 4.8: Comparison of F1 scores for the user preference models' predictions on the rated molecules from the validation set, for both the HITL and baseline runs. Each participant is represented by two bars, one for each set of validation data. Note that for participants 2 and 5, the F1 scores are zero where there are no bars present.

4.3.2 Comparing User Preferences in HITL vs Control Molecules

The most critical question our study aimed to answer was whether the incorporation of user feedback into the design of molecules through HITL interventions resulted in a significant increase in the number of user-liked molecules. For this purpose, the number of users' "likes" was compared between the HITL-generated molecules and control molecules. The methodology used for generating these two sets of molecules was described in detail in the Methods chapter. Note that only 7 out of the 8 participants responded to the follow-up and one participant is therefore not represented here.

The results demonstrate a clear difference in user preferences for the HITL and control molecules (Figure 4.9). Out of 7 participants, 6 showed a higher number of "likes" for molecules generated during the HITL interventions.

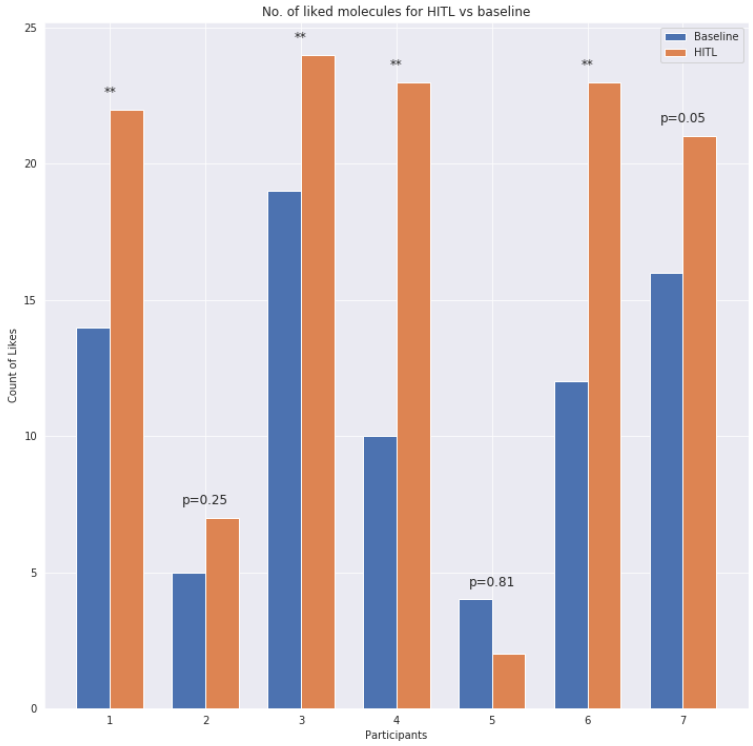

Figure 4.9: Comparison of the number of users' "likes" for the HITL and control molecules for each user. ** indicates a P value of less than 0.05 from a one-sided statistical test.

A one-sided statistical test comparing the number of liked molecules from the combined ratings of all participants revealed a P-value of less than 0.005, providing statistical evidence that the increase in the number of users' "likes" between the HITL and control molecules is significant. Furthermore, 4 out of the 7 participant validation sets also showed a statistically significant increase from the baseline set, as indicated in Figure 4.9

4.3.3 Correlation of User Preference

The distribution of user "likes" for each molecule in the baseline run offers another perspective on understanding user preferences. As each participant was shown the same 25 subset of molecules derived from the baseline set, we can analyze this distribution to observe how often users agreed on liking or disliking the same molecules. This is unlike the HITL sets, each consisting of 25 molecules which are unique for each user, as they were all derived from different runs. A histogram is used to show the number of molecules each user liked in the baseline run (Figure 4.11). However, participant three did not rate all the selected molecules from the baseline set and is therefore not represented in the histogram.

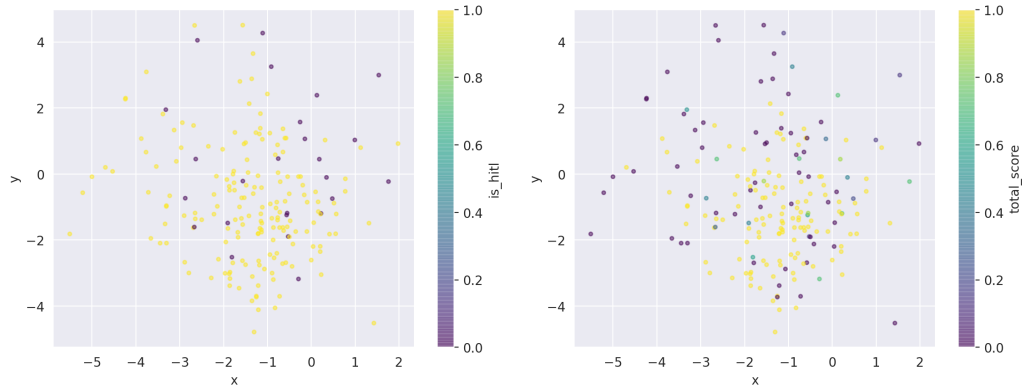

Figure 4.10: This figure contains two plots of the validation data utilizing CDDD and PCA as described in Section 3.1.2. Both plots depict the distribution of the validation data from the experiment involving human users. In the left plot, the yellow points represent the interventions, while the purple points indicate the validation set. The right plot, in contrast, colors the points based on their average rating - a "like" is signified with yellow, while "dislike" is represented in purple. Note that the molecules from the baseline set were evaluated by multiple users, and the color reflects the average rating.

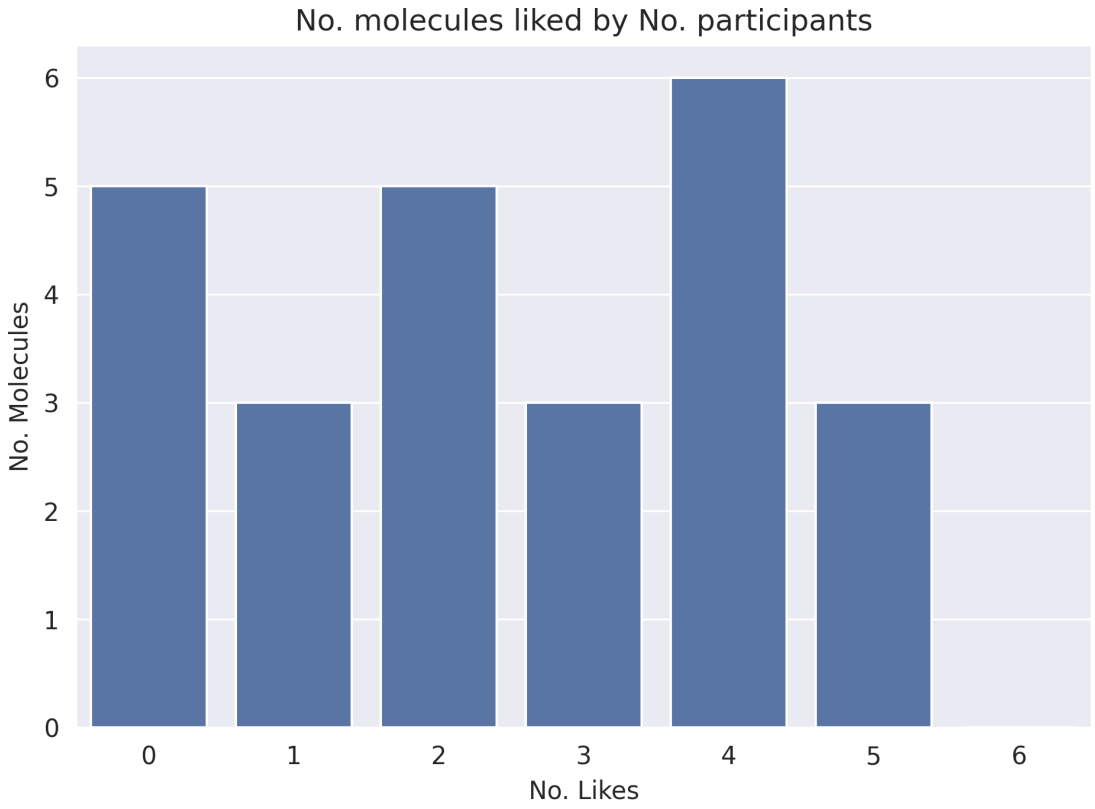

Figure 4.11: Histogram showing the distribution of user "likes" for each molecule in the validation set from the baseline run.

Seven molecules from the baseline set were unanimously disliked by all users, and three molecules were unanimously liked. Additionally, a variable number of molecules were liked by 1 to 3 users, suggesting that some preferences were shared among a subset of users, but not all.

Under the assumption of no correlation between user preferences, we would expect a binomial distribution, whilst a perfect correlation would have no molecules with one to five likes. The observed distribution, however, lies in between these distributions, indicating some degree of correlation in user preferences.

Continuing on, an additional measure of user preference similarity was sought by calculating Pearson's correlation between the predicted probabilities of each pair of user preference models. Here, the final version of the user preference models from each user run was used to rate the top 1000 scoring molecules from the baseline runs. A positive correlation would suggest that the models, while trained on different user preferences, tended to rank the same molecules highly. Conversely, a negative correlation or a correlation near zero would suggest that the models trained on different users tended to rank the molecules differently.

A visual representation of these correlations is given in Figure 4.12, where the color intensity of each cell represents Pearson's correlation between the corresponding pair of user models.

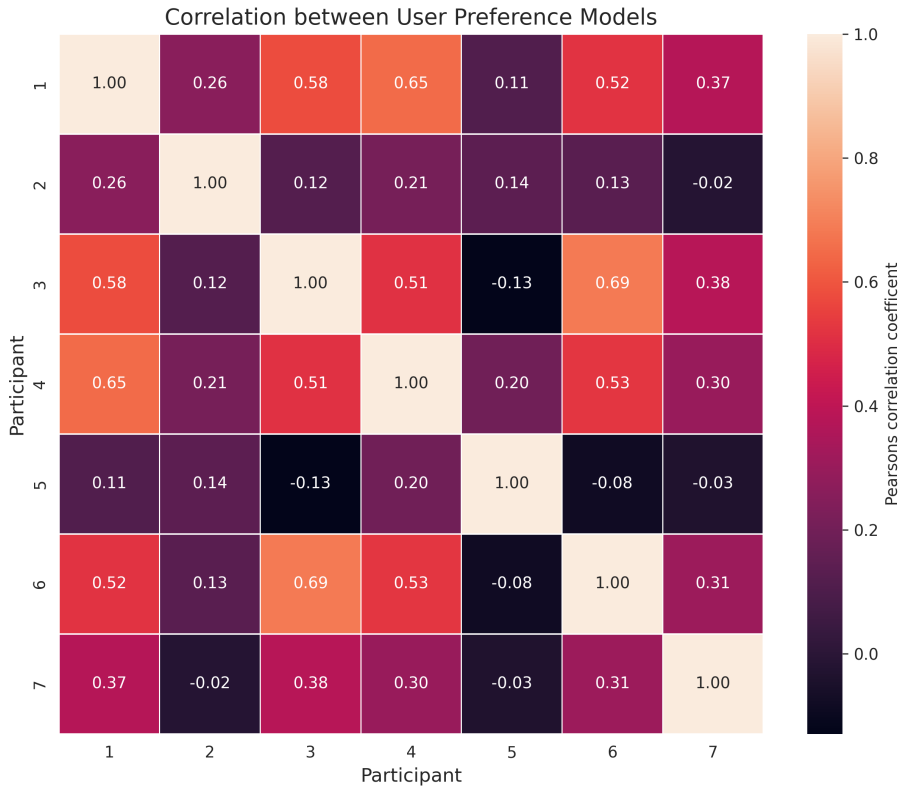

Figure 4.12: Heatmap of Pearson's correlations between each pair of user preference models trained during the user tests. Each cell color represents Pearson's correlation coefficient between the pair of user models corresponding to the row and column.

Figure 4.12 reveals high positive correlations between identical user models, as expected, and varying levels of correlations between different user models, ranging from positive, near-zero, to even slightly negative. A group of correlated models can be identified between participants one, three, four, and six, each demonstrating a correlation above 0.5 with each other. Conversely, participants two and five do not exhibit a correlation above 0.5 with any other model. Notably, these are also the participants with lower F1 scores on both the baseline and HITL sets 4.8.

5

Discussion

This chapter discusses the potential of the Human-in-the-loop (HITL) system, with a particular focus on its influence on molecular favorability. The potential practical applications within drug discovery and future research directions are also explored. Emphasis is placed on the importance of flexible preference functions, given the observed diversity in user preferences. Moreover, the visualization methods implemented in this study are evaluated. Finally, the limitations encountered during the research process are acknowledged.

5.1 Enhancing Molecular Favorability with HITL

The efficacy of the proposed HITL system in enhancing user favorability of generated molecules was substantiated by our A/B test results. Figure 4.9 exhibits an average favorability increase of 69% from the baseline for the 6 out of 7 participants that showed an increase in favorability, with the single largest increase being for participant 4 which increase by 130%. Although the difference between the combined groups is statistically significant, not all users demonstrated a significant difference, possibly due to a fairly small sample size in some cases.

Furthermore, the surrogate user model testing strengthen this claim, as it also demonstrated the system's ability to boost favorability across a variety of surrogate user models with different complexity levels.

These experiments have further demonstrated that the HITL system does not notably impair the performance of other scoring components or the diversity of the generated molecules.

Notably, some nonlinear surrogate user models appear to enhance diversity relative to the baseline. This could be attributed to two factors: either the scoring component favoring a range of molecules when using HITL or a decreased overall learning rate, as the User Preference Model struggles to learn and seldom produces probabilities near 1. The latter would effectively slow down the decrease in diversity, as seen in Figure 4.3, where diversity decreases linearly as the run progresses. This slower convergence towards low-diversity areas might explain the observed increase in diversity.

5.1.1 Different Aspects of the HITL System

As surrogate user models were utilized, various aspects of the proposed system were tested, including different online learning models, rating frequencies, sampling methods, and the number of rated molecules.

The surrogate user model testing revealed that several user preference models were proficient at increasing user favorability. The models tested (K-nearest neighbors (KNN), Random Forest, and Logistic Regression) were all competent at this task when utilizing Continuous and Data-Driven Descriptors (CDDD) embedding, with Logistic Regression appearing to be the best-performing model among the three.

Despite the satisfactory performance of these simple machine learning architectures, a more advanced and optimized model could potentially deliver substantial performance improvements, either in terms of user favorability or by reducing the number of molecules a user needs to rate.

The number of rated molecules required for an increase in user favorability was found to be relatively small. The smallest amount of rated molecules tested was 60, which showed an increase in favorability for all surrogate user models. However, the increase in favorability seems to correlate with the number of rated molecules, with a higher number being more effective but exhibiting diminishing returns. This results in a trade-off between performance and user effort. The complexity of the preferences might also play a role: the Rule-based user models did not experience as significant an increase when rating more molecules compared to the Random Forest models, indicating that more complex preferences require more effort.

Additionally, the frequency of updating the user model was not found to be a crucial factor, within the range of 5 to 20 steps between interventions. Therefore, we advocate for a lower rating frequency of once every 20 steps as it places less demand on the user. However, further reducing this frequency would require additional testing, as we do not want to make assumptions about how this would generalize outside of the tested range.

Similar to the intervention frequency, different sampling strategies did not produce any significant improvements in terms of the measured metrics. However, we speculate that biasing the selection towards higher-scoring molecules could yield other potential benefits, primarily making the online learning model more accurate for high-scoring molecules. This could be useful as these molecules are the ones most relevant to the user, as they are the main contenders for consideration if they also achieve the other goals defined in the Multiparameter Scoring Objective (MPO).

5.2 The Importance of a Flexible Preference Function

As outlined in Section 4.3.3, there was substantial variation in preferences between participants. This suggests that a "one-size-fits-all" solution may not be suitable for optimizing user preference. Another factor noted by several participants during the

experiments was that some molecules could be favored for some projects but not for others. This introduces another level of variance, dependent on the specific project a chemist is working on. This further justifies the online learning approach as it offers a flexible method for optimizing user preferences, on a run-to-run basis.

5.3 Visualization

This project has introduced several metrics using Continuous and Data-Driven Descriptors (CDDD) embedding. However, it is not inherently apparent that our proposed metrics are accurate or useful. From the results, a visual correlation between the Euclidean distance between embeddings and the more commonly used Tanimoto distance is demonstrated. Although there seems to be some correlation, further testing is required before reaching a definitive conclusion based on these metrics. Particularly, the diversity metric shows promise as it is computationally more feasible than, for example, pairwise comparison of Tanimoto distances for larger data sets, due to the complexity growing quadratically for pairwise comparisons. Furthermore, the visualization dashboard presents what we believe are intriguing properties from a run. However, feedback from actual users needs to be collected before determining the usefulness of the proposed visualizations and metrics. In this regard, more comparisons with alternative methods should be considered to properly evaluate the metrics.

5.4 Limitations

This project implemented two primary types of studies: those utilizing surrogate user models and those with human participants. Each approach has its strengths, but also inherent limitations which warrant careful consideration.

Surrogate user models have the advantage of facilitating extensive testing, making it relatively straightforward to draw conclusions from these tests. However, it is uncertain how the findings from these models would generalize to real-world chemists. These models, being machine learning-based, might not capture the complexities of human preferences adequately. Their preference functions, focusing on a single but complex goal, could potentially be easier to exploit since they lack a comprehensive understanding of the multifaceted aspects of chemistry.

On the other hand, tests involving human participants were constrained by the available time individuals were willing to spend on participating in the study, resulting in smaller sample sizes. This limits our ability to demonstrate statistical significance on an individual participant basis. While the combination of all rated molecules from all participants did yield statistical significance over the baseline set, the stochastic nature of REINVENT makes it challenging to definitively attribute this difference to the inclusion of the HITL component.

For instance, two runs with the same MPO function could produce different sets of 100,000 sampled molecules, possibly exhibiting statistically significant differences.

However, this does not necessarily indicate that the change arises from the MPO, as it would be identical in both runs; instead, the discrepancy might be the result of REINVENT's stochastic behavior. This is evident when looking at the variance present in favorability between the baseline runs for some of the surrogate user models.

Another factor that may have influenced the outcome of the testing is the configuration of the run. The settings for the Multiparameter Optimization (MPO) and the number of steps, while not unusual or unrealistic, somewhat lacked a definitive purpose. As the participants did not have a clear target, project, or specific purpose for the generated molecules, defining what makes a good molecule could become ambiguous.

This ambiguity could, in some aspects, be advantageous, as we would not necessarily want a user to replace tasks like molecular docking but rather act as a filter to sort out "undesirable" molecules. However, the lack of a specific goal for REINVENT to optimize towards could have led to the generation of overly simplistic molecules. Some evidence of this was observed when participants noted that although there were no obvious flaws with a molecule, its simplicity negated its usefulness.

5.5 Future Applications

The HITL system could accelerate the drug discovery process by merging the strengths of AI with human expertise, offering a new tool in the REINVENT arsenal.

In practice, the introduction of the proposed HITL system would likely increase the proportion of valuable molecules generated, thereby reducing the time chemists spend sorting through these compounds. As a result, promising drug candidates could be identified more quickly and cost-effectively, ultimately improving patient outcomes.

A production system built on these findings might look slightly different. For instance, a run probably shouldn't be paused during the rating of molecules; instead, a user could choose when to rate molecules rather than being compelled to. Also, there is no strict need to pause a run while a user is rating molecules, as this was done in this project to maintain experimental control.

The suggested HITL system permits different chemists to develop their own preference models. A potential application could involve a user rating a set of molecules during a run and saving the rated molecules. For subsequent runs within a similar domain, the user could start with a pre-trained online learning model incorporating ratings from previous runs, instead of starting from scratch. This online learning strategy would enable the formation of a preference model specific to the chemistry area relevant to the project or target. The efficacy of these online learning models in guiding a new run isn't obvious and would require additional testing, but they could significantly enhance usability by reducing the chemist's workload. Moreover, when several chemists work on a similar project, their preferences could be pooled to create a shared user preference model by pretraining the HITL component with

saved molecules from previous runs.

With frequent interactions with the HITL system, the number of rated molecules within a project could increase, potentially yielding a well-performing pre-trained model that could be shared among chemists working on the same project.

6

Conclusion

In this project, we introduced the Human-in-the-loop (HITL) system as an additional approach to enhance the generation of user-favorable molecules in drug discovery. Through rigorous testing involving user model experiments and human user testing, we confirmed the effectiveness of the HITL system in significantly increasing favorability compared to the baseline. Importantly, the HITL system showcased no evident compromise on the performance of existing scoring components and also demonstrated no adverse effects on the diversity of generated molecules. This highlights the compatibility of the HITL system with existing methods and its potential to improve the efficiency of the drug discovery process.

In conclusion, the HITL system presented in this paper offers a valuable tool for enhancing molecular favorability in drug discovery. By bridging the gap between AI and human expertise, the HITL system offers a transformative approach that addresses the challenge of efficiently navigating large chemical spaces. HITL system has the potential to revolutionize the field, enabling faster and more efficient identification of promising drug candidates, ultimately improving patient outcomes.

Future research can focus on key areas to enhance the HITL system's performance. Exploring advanced machine learning models for online learning can further improve favorability while reducing the number of compounds requiring user ratings will streamline the workflow. Developing a production-ready system that allows continuous user interactions without pausing during human feedback collection will enhance its practical implementation. This will also enable faster identification of promising drug candidates. Creating a personalized pre-trained online learning model based on chemist preferences or project goals can optimize user preferences and tailor the HITL system to specific chemistry areas. This approach reduces the workload for chemists and improves system usability. Additionally, investigate how the specific phrasing used in the instructions influences user ranking of molecules, by differentiating between various definitions and providing valuable insights for the system's accuracy. For e.g: On the definition of "drug-like" and "potential binder for X protein" from the user's perspective. Although this aspect extends beyond the scope of this project, this research direction holds potential for future exploration.

Bibliography

[1] J. Hughes, S. Rees, S. Kalindjian, and K. Philpott, "Principles of early drug discovery," *British journal of pharmacology*, vol. 162, no. 6, pp. 1239–1249, Mar. 2011. DOI: 10.1111/j.1476-5381.2010.01127.x. [Online]. Available: https://doi.org/10.1111/j.1476-5381.2010.01127.x.

[2] *In silico experiments: Is there a breakthrough for pharma*, https://www.avenga.com/magazine/in-silico-experiments-in-pharma, Accessed: 2023-02-10.

[3] J.-L. Reymond, "The chemical space project," *Accounts of Chemical Research*, vol. 48, no. 3, pp. 722–730, 2015, PMID: 25687211. DOI: 10.1021/ar500432k. eprint: https://doi.org/10.1021/ar500432k. [Online]. Available: https://doi.org/10.1021/ar500432k.

[4] T. Blaschke, J. Arús-Pous, H. Chen, *et al.*, "Reinvent 2.0: An ai tool for de novo drug design," *Journal of Chemical Information and Modeling*, vol. 60, no. 12, pp. 5918–5922, 2020, PMID: 33118816. DOI: 10.1021/acs.jcim.0c00915. eprint: https://doi.org/10.1021/acs.jcim.0c00915. [Online]. Available: https://doi.org/10.1021/acs.jcim.0c00915.

[5] M. Olivecrona, T. Blaschke, O. Engkvist, and H. Chen, "Molecular de-novo design through deep reinforcement learning," *Journal of Cheminformatics*, vol. 9, no. 1, p. 48, Sep. 2017, ISSN: 1758-2946. DOI: 10.1186/s13321-017-0235-x. [Online]. Available: https://doi.org/10.1186/s13321-017-0235-x.

[6] S. C. H. Hoi, D. Sahoo, J. Lu, and P. Zhao, *Online learning: A comprehensive survey*, 2018. arXiv: 1802.02871 [cs.LG].

[7] W. Gao, T. Fu, J. Sun, and C. W. Coley, "Sample efficiency matters: A benchmark for practical molecular optimization," *a*, 2022. DOI: 10.48550/ARXIV.2206.12411. [Online]. Available: https://arxiv.org/abs/2206.12411.

[8] X. Zeng, F. Wang, Y. Luo, *et al.*, "Deep generative molecular design reshapes drug discovery," *Cell Reports Medicine*, vol. 3, no. 12, p. 100 794, 2022, ISSN: 2666-3791. DOI: https://doi.org/10.1016/j.xcrm.2022.100794. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666379122003494.

[9] D. Weininger, A. Weininger, and J. L. Weininger, "Smiles. 2. algorithm for generation of unique smiles notation," *Journal of chemical information and computer sciences*, vol. 29, no. 2, pp. 97–101, 1989.

[10] M. Krenn, F. Häse, A. Nigam, P. Friederich, and A. Aspuru-Guzik, "Self-referencing embedded strings (selfies): A 100 percent robust molecular string representation," *Machine Learning: Science and Technology*, vol. 1, no. 4,

p. 045 024, Oct. 2020. DOI: 10.1088/2632-2153/aba947. [Online]. Available: https://dx.doi.org/10.1088/2632-2153/aba947.

[11] L. David, A. Thakkar, R. Mercado, and O. Engkvist, "Molecular representations in ai-driven drug discovery: A review and practical guide," *Journal of Cheminformatics*, vol. 12, no. 1, p. 56, Sep. 2020. DOI: 10.1186/s13321-020-00460-5. [Online]. Available: https://doi.org/10.1186/s13321-020-00460-5.

[12] T. Blaschke, O. Engkvist, J. Bajorath, and H. Chen, "Memory-assisted reinforcement learning for diverse molecular de novo design," *Journal of Cheminformatics*, vol. 12, no. 1, p. 68, Nov. 2020, ISSN: 1758-2946. DOI: 10.1186/s13321-020-00473-0. [Online]. Available: https://doi.org/10.1186/s13321-020-00473-0.

[13] I. Sundin, A. Voronov, H. Xiao, *et al.*, "Human-in-the-loop assisted de novo molecular design," *Journal of Cheminformatics*, vol. 14, no. 1, p. 86, Dec. 2022, ISSN: 1758-2946. DOI: 10.1186/s13321-022-00667-8. [Online]. Available: https://doi.org/10.1186/s13321-022-00667-8.

[14] J. Guo, J. P. Janet, M. R. Bauer, *et al.*, "Dockstream: A docking wrapper to enhance de novo molecular design," *Journal of Cheminformatics*, vol. 13, no. 1, p. 89, Nov. 2021, ISSN: 1758-2946. DOI: 10.1186/s13321-021-00563-7. [Online]. Available: https://doi.org/10.1186/s13321-021-00563-7.

[15] M. García-Ortegón, G. N. C. Simm, A. J. Tripp, J. M. Hernández-Lobato, A. Bender, and S. Bacallado, "Dockstring: Easy molecular docking yields better benchmarks for ligand design," *Journal of Chemical Information and Modeling*, vol. 62, no. 15, pp. 3486–3502, 2022, PMID: 35849793. DOI: 10.1021/acs.jcim.1c01334. eprint: https://doi.org/10.1021/acs.jcim.1c01334. [Online]. Available: https://doi.org/10.1021/acs.jcim.1c01334.

[16] K. Papadopoulos, K. A. Giblin, J. P. Janet, A. Patronov, and O. Engkvist, "De novo design with deep generative models based on 3d similarity scoring," *Bioorganic and Medicinal Chemistry*, vol. 44, p. 116 308, 2021, ISSN: 0968-0896. DOI: https://doi.org/10.1016/j.bmc.2021.116308. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968089621003163.

[17] X. Wu, L. Xiao, Y. Sun, J. Zhang, T. Ma, and L. He, "A survey of human-in-the-loop for machine learning," *Future Generation Computer Systems*, vol. 135, pp. 364–381, Oct. 2022. DOI: 10.1016/j.future.2022.05.014. [Online]. Available: https://doi.org/10.1016%5C%2Fj.future.2022.05.014.

[18] Y. Bai, A. Jones, K. Ndousse, *et al.*, *Training a helpful and harmless assistant with reinforcement learning from human feedback*, 2022. arXiv: 2204.05862 [cs.CL].

[19] Y. Liu, T. Han, S. Ma, *et al.*, *Summary of chatgpt/gpt-4 research and perspective towards the future of large language models*, 2023. arXiv: 2304.01852 [cs.CL].

[20] A. Capecchi, D. Probst, and J.-L. Reymond, "One molecular fingerprint to rule them all: Drugs, biomolecules, and the metabolome," *Journal of Cheminformatics*, vol. 12, no. 1, p. 43, Jun. 2020, ISSN: 1758-2946. DOI: 10.1186/s13321-

020-00445-4. [Online]. Available: https://doi.org/10.1186/s13321-020-00445-4.

[21] D. Rogers and M. Hahn, "Extended-connectivity fingerprints," *Journal of Chemical Information and Modeling*, vol. 50, no. 5, pp. 742–754, May 2010, ISSN: 1549-9596. DOI: 10.1021/ci100050t. [Online]. Available: https://doi.org/10.1021/ci100050t.

[22] R. Winter, F. Montanari, F. Noé, and D.-A. Clevert, "Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations," *Chem. Sci.*, vol. 10, pp. 1692–1701, 6 2019. DOI: 10.1039/C8SC04175J. [Online]. Available: http://dx.doi.org/10.1039/C8SC04175J.

[23] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN Computer Science*, vol. 2, no. 3, Mar. 2021, ISSN: 2661-8907. DOI: 10.1007/s42979-021-00592-x. [Online]. Available: https://doi.org/10.1007/s42979-021-00592-x.

[24] R. Bro and A. K. Smilde, "Principal component analysis," *Analytical methods*, vol. 6, no. 9, pp. 2812–2831, 2014.

[25] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," 2018. DOI: 10.48550/ARXIV.1802.03426. [Online]. Available: https://arxiv.org/abs/1802.03426.

[26] Gaulton, Anna, Bellis, *et al.*, "ChEMBL: A large-scale bioactivity database for drug discovery," en, *Nucleic Acids Res*, vol. 40, no. Database issue, pp. D1100–7, Sep. 2011.

[27] MolecularAI, *Reinventcommunity*, https://github.com/MolecularAI/ReinventCommunity, GitHub repository, 2023.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[29] Streamlit, *Streamlit The fastest way to build custom ML tools*, Accessed: 2023-05-17, 2023. [Online]. Available: https://streamlit.io.

[30] *Dataset collection on moleculenet - a benchmark for molecular machine learning*, https://deepchemdata.s3-us-west-1.amazonaws.com/datasets/bace.csv, Accessed: 2023-04-10.

[31] "Chemical structure-related drug-like criteria of global approved drugs.," *Journal Article*, vol. 21, no. 1, p. 75, 2016. DOI: 10.3390/molecules21010075. [Online]. Available: https://doi:%2010.3390/molecules21010075..

[32] A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[33] MolecularAI, *Reinventcommunity*, https://github.com/MolecularAI/ReinventCommunity, Accessed: 2023-05-17, 2023.

[34] RDKit, *RDKit: Open-source cheminformatics*, Accessed: 2023-05-17, 2023. [Online]. Available: http://www.rdkit.org.

[35] D. Bajusz, A. Rácz, and K. Héberger, "Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations?" *Journal of Cheminfor-*

matics, vol. 7, no. 1, p. 20, 2015. DOI: 10.1186/s13321-015-0069-3. [Online]. Available: https://doi.org/10.1186/s13321-015-0069-3.

A

Appendix 1

A.1 REINVENT run configurations

An example JSON file for the configurations used in section 3.4.1 is shown in Listing A.1. Note that some system specific objects such as system paths have been replaced by generic names such as "path/to", and that the HITL scoring component is not included here.

Listing A.1: Reinvent Configuration for testing with Surrogate User Models

```
{
"version": 3,
"run_type": "curriculum_learning",
"parameters": {
    "curriculum_type": "manual",
    "scoring_function": {
        "name": "custom_product",
        "parallel": False,
        "parameters": [
            {
                "weight": 1,
                "component_type": "qed_score",
                "name": "QED Score"
            },
            {
                "weight": 1,
                "component_type": "molecular_weight",
                "name": "Molecular weight",
                "specific_parameters": {
                    "transformation": {
                        "transformation_type": "
                           double_sigmoid",
                        "high": 600,
                        "low": 300,
                        "coef_div": 500,
                        "coef_si": 20,
                        "coef_se": 20
```

```
                    }
                }
            },
            {
                "weight": 1,
                "component_type": "num_rings",
                "name": "Number of rings",
                "specific_parameters": {
                    "transformation": {
                        "transformation_type": "step",
                        "high": 4,
                        "low": 1
                    }
                }
            },
            {
                "weight": 1,
                "component_type": "num_hbd_lipinski",
                "name": "Number of HB-donors (Lipinski)",
                "specific_parameters": {
                    "transformation": {
                        "transformation_type": "step",
                        "high": 3,
                        "low": 0
                    }
                }
            }
        }
    ]
},

        "diversity_filter": {
            "minscore": 0.0,
            "name": "NoFilter",
            "bucket_size": 125,
            "minsimilarity": 0.4
        },
        "curriculum_learning": {
            "agent": "path/to/prior/random.prior.new"
              ,
            "prior": "path/to/prior/random.prior.new"
              ,
            "pause_lock":  "run/path/pause.lock",
            "pause_limit": 60000,
            "update_lock": "path/to/run/update.lock",
            "general_configuration_path": "path/to/
```

```
                    config",
                "n_steps": n_steps,
                "sigma": 128,
                "learning_rate": 0.0001,
                "batch_size": 128,
                "reset": 0,
                "reset_score_cutoff": 0.5,
                "margin_threshold": 50,
                "scheduled_update_step": -1
            },
            "inception": {
                "smiles": [],
                "memory_size": 100,
                "sample_size": 10
            }
        },
        "logging" : {
                "job_name": "Curriculum Learning HITL",
                "sender": "",
                "recipient": "local",
                "logging_frequency": 1,
                "logging_path": "path/to/logging",
                "result_folder": "path/to/logging/results
                    ",
                "job_id": "XXXXXXXXXXXXX"
        }
    }
}
```

Listing A.2: Reinvent Configuration for the run used in the real user testing

```
{
"version": 3,
"run_type": "curriculum_learning",
"parameters": {
    "curriculum_type": "manual",
    "scoring_function": {
        "name": "custom_product",
        "parallel": False,
        "parameters": [
            {
                "weight": 1,
                "component_type": "qed_score",
                "name": "QED Score"
            },
            {
                "weight": 1,
```

```
                "component_type": "molecular_weight",
                "name": "Molecular weight",
                "specific_parameters": {
                    "transformation": {
                        "transformation_type": "
                           double_sigmoid",
                        "high": 600,
                        "low": 300,
                        "coef_div": 500,
                        "coef_si": 20,
                        "coef_se": 20
                    }
                }
            },
            {
                "weight": 1,
                "component_type": "num_rings",
                "name": "Number of rings",
                "specific_parameters": {
                    "transformation": {
                        "transformation_type": "step",
                        "high": 4,
                        "low": 1
                    }
                }
            },
            {
                "weight": 1,
                "component_type": "num_hbd_lipinski",
                "name": "Number of HB-donors (Lipinski)",
                "specific_parameters": {
                    "transformation": {
                        "transformation_type": "step",
                        "high": 3,
                        "low": 0
                    }
                }
            },
            {
            "component_type": "custom_alerts",
            "name": "Custom alerts",
            "weight": 1,
            "specific_parameters": {
                "smiles": [
                    "[*;r8]",
                    "[*;r9]",
```

```
62                      "[*;r10]",
63                      "[*;r11]",
64                      "[*;r12]",
65                      "[*;r13]",
66                      "[*;r14]",
67                      "[*;r15]",
68                      "[*;r16]",
69                      "[*;r17]",
70                      "[#8][#8]",
71                      "[#6;+]",
72                      "[#16][#16]",
73                      "[#7;!n][S;!$(S(=O)=O)]",
74                      "[#7;!n][#7;!n]",
75                      "C#C",
76                      "C(=[O,S])[O,S]",
77                      "[#7;!n][C;!$(C(=[O,N])[N,O])][#16;!s
                           ]",
78                      "[#7;!n][C;!$(C(=[O,N])[N,O])][#7;!n]
                           ",
79                      "[#7;!n][C;!$(C(=[O,N])[N,O])][#8;!o]
                           ",
80                      "[#8;!o][C;!$(C(=[O,N])[N,O])][#16;!s
                           ]",
81                      "[#8;!o][C;!$(C(=[O,N])[N,O])][#8;!o]
                           ",
82                      "[#16;!s][C;!$(C(=[O,N])[N,O])][#16;!
                           s]"
83                  ]
84              }
85             }
86         ]
87     },
88
89             "diversity_filter": {
90                 "minscore": 0.0,
91                 "name": "NoFilter",
92                 "bucket_size": 125,
93                 "minsimilarity": 0.4
94             },
95             "curriculum_learning": {
96                 "agent": "path/to/prior/random.prior.new"
                       ,
97                 "prior": "path/to/prior/random.prior.new"
                       ,
98                 "pause_lock":  "run/path/pause.lock",
99                 "pause_limit": 60000,
```

```
100                 "update_lock": "path/to/run/update.lock",
101                 "general_configuration_path": "path/to/
                      config",
102                 "n_steps": n_steps,
103                 "sigma": 128,
104                 "learning_rate": 0.0001,
105                 "batch_size": 128,
106                 "reset": 0,
107                 "reset_score_cutoff": 0.5,
108                 "margin_threshold": 50,
109                 "scheduled_update_step": -1
110             },
111             "inception": {
112                 "smiles": [],
113                 "memory_size": 100,
114                 "sample_size": 10
115             }
116         },
117         "logging" : {
118                 "job_name": "Curriculum Learning HITL",
119                 "sender": "",
120                 "recipient": "local",
121                 "logging_frequency": 1,
122                 "logging_path": "path/to/logging",
123                 "result_folder": "path/to/logging/results
                      ",
124                 "job_id": "XXXXXXXXXXXXX"
125         }
126     }
127 }
```

A.2 User experiments

In this section, we include our instructions given to the user during the user experiments.

A.2.1 Experiment Instructions

The aim of this experiment is to enhance our understanding of how to effectively guide AI systems, specifically REINVENT, in generating useful molecules by allowing users to rate a subset of molecules produced during a run. Please read the following instructions carefully. During the experiment, you will be presented with website containing a grid of molecules. Next to each molecule is an option of like or dislike. Select an option for each molecule based on your preference and intuition of what constitutes a valuable suggestion from an AI system. If no option is selected,

the default value will be "like". After selecting a rating for each molecule, press the submit ratings button. You must select at least one molecule as like and at least one as dislike before submitting. After submitting the molecules, the text "Please Press the R key" will replace the grid. Press the "R" key on your keyboard, and the text "Waiting for molecules" will appear. Wait for new molecules to appear before rating each one and pressing the submit button again. Repeat this process until you have submitted ratings for 7 sets of molecules. Following this, the experiment will end, and no new molecules will be displayed. If an error message appears during the experiment, press the "R" key on your keyboard or refresh the webpage.