Reinforced Active Learning for Large-Scale Virtual Screening with Learnable Policy Model

Yicong Chen[1,*], **Jiahua Rao**[1,*], **Jiancong Xie**[1], **Dahao Xu**[1], **Zhen Wang**[1], **Yuedong Yang**[1,†]
[1]Sun Yat-sen University [*]Equal Contribution [†]Corresponding Author
`{chenyc57, xiejc3, xudh6}@mail2.sysu.edu.cn`
`{raojh7, wangzh665, yangyd25}@mail.sysu.edu.cn`

Abstract

Virtual Screening (VS) is vital for drug discovery but struggles with low hit rates and high computational costs. While Active Learning (AL) has shown promise in improving the efficiency of VS, traditional methods rely on inflexible and handcrafted heuristics, limiting adaptability in complex chemical spaces, particularly in balancing molecular diversity and selection accuracy. To overcome these challenges, we propose GLARE[1], a reinforced active learning framework that reformulates VS as a Markov Decision Process (MDP). Using Group Relative Policy Optimization (GRPO), GLARE dynamically balances chemical diversity, biological relevance, and computational constraints, eliminating the need for inflexible heuristics. Experiments show GLARE outperforms state-of-the-art AL methods, with a 64.8% average improvement in Enrichment Factors (EF). Additionally, GLARE enhances the performance of VS foundation models like DrugCLIP, achieving up to an 8-fold improvement in $\text{EF}_{0.5\%}$ with as few as 15 active molecules. These results highlight the transformative potential of GLARE for adaptive and efficient drug discovery.

1 Introduction

Virtual Screening (VS) plays a pivotal role in drug design, facilitating the identification of active molecules that interact with biological targets from vast chemical libraries [30]. By predicting the likelihood of activity for each molecule, VS reduces the need for exhaustive experimental testing [43, 62, 40]. However, despite its importance, VS faces significant challenges, including low hit rates (the proportion of active molecules identified) [46] and the computational cost of screening ultra-large libraries ($10^6 - 10^8$ compounds) [36, 13]. These limitations have driven the search for more efficient and accurate methodologies to improve the scalability and performance of VS [60].

Active Learning (AL) has shown promise in virtual screening by iteratively refining models through the selective acquisition of informative molecular samples [14, 26]. Popular strategies often involve uncertainty-based, diversity-based, and hybrid approaches [48, 64, 7]. Diversity-based approaches ensure broad exploration by selecting diverse samples, while uncertainty-based methods focus on refining predictions by prioritizing uncertain molecules [5, 55]. However, both of them rely on hand-crafted selection strategies. They are often heavily dependent on the diversity and size of the initial molecular pool, thus lacking the flexibility to dynamically balance exploration and exploitation when new data are acquired [5].

Recent advancements in AL have sought to improve efficiency by introducing frameworks that dynamically combine or select strategies, reducing the need for manual intervention [35, 19]. For instance, SelectAL [15] adjusts AL strategies based on specific tasks and computational budgets, while AutoAL [57] employs a differentiable framework to identify optimal strategies. Although

---

[1] Source code is available at https://github.com/biomed-AI/GLARE

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

these approaches mark progress, they remain focused on optimizing the selection of strategies from a predefined set (i.e., "learning to select rules"). This approach does not enable models to directly learn how to identify the most promising candidate molecules (i.e., "learning to select molecules"). As a result, these methods are not well-suited for drug discovery, where the vast and complex chemical space requires highly adaptive and intelligent exploration to effectively balance structural diversity, biological relevance, and computational efficiency [10].

In this work, we introduce **GLARE**, a **G**RPO-based **L**earning framework for **A**ctive **RE**inforced screening, designed to overcome the limitations of traditional active learning methods and enhance large-scale virtual screening. GLARE reformulates the virtual screening process as a Markov Decision Process (MDP), enabling Reinforcement Learning (RL) to dynamically optimize molecular selection strategies. By leveraging Group Relative Policy Optimization (GRPO) [49], GLARE eliminates the reliance on manually-designed heuristics, learning to adaptively screen large-scale chemical spaces. By balancing molecular diversity, biological relevance, and computational efficiency, GLARE provides a robust and scalable solution for AL in drug discovery.

Through extensive experiments, we demonstrate that GLARE significantly outperforms state-of-the-art AL methods, achieving an average 64.8% improvement in Enrichment Factors (EF). Furthermore, it improves the performance of virtual screening foundation models like DrugCLIP, achieving up to an 8-fold increase in $\text{EF}_{0.5\%}$ with as few as 15 active molecules given. These results highlight the transformative potential of GLARE in enabling adaptive, efficient, and intelligent molecular selection for drug discovery.

In summary, our contributions are threefold:

- We reformulate virtual screening as a Markov Decision Process (MDP), enabling reinforcement learning to dynamically optimize molecular selection.
- By utilizing a learnable policy model trained with Group Relative Policy Optimization (GRPO), GLARE eliminates the need for manually-designed heuristics and learns to adaptively screen the complex chemical spaces.
- Extensive experiments demonstrate that GLARE significantly outperforms state-of-the-art active learning methods, enhancing virtual screening efficiency and scalability, while also substantially improving the performance of virtual screening foundation models.

2 Related Works

2.1 Virtual Screening Methods

Virtual screening methods are broadly categorized into docking-based and deep learning-based approaches [12]. Docking-based methods [16, 56] use predefined force fields to estimate binding scores, predicting the binding energy, optimal orientation, and conformation of a small molecule within a protein binding site [23]. However, their reliance on rigid scoring functions and static assumptions limits their accuracy and scalability in complex chemical spaces [21].

Deep learning-based virtual screening approaches typically focus on predicting drug-target interactions and binding affinities [34, 61, 32, 28, 63, 39, 29]. These methods represent drugs and target proteins using various encoding techniques, such as molecular fingerprints, SMILES strings, graph-based representations [38, 37] for drugs, and sequence-based or structural embeddings for proteins. These methods leverage large experimental datasets and known interaction information to predict new potential interactions [20, 9, 50], offering a more comprehensive perspective for drug discovery.

However, as the size of chemical and biological data grows, the computational cost and inefficiency of exhaustively labeling data become significant challenges [13, 46]. This highlights the need for active learning to prioritize data points that are most informative for model improvement [42, 41, 14], enabling more effective exploration of the chemical space.

2.2 Active Learning

Active Learning (AL) has been widely applied in drug discovery and development to iteratively guide the selection of unlabeled data for annotation, maximizing model performance with minimal data [14, 26, 5, 55, 8]. Common strategies include uncertainty-based sampling, diversity-based sampling, and hybrid approaches [48, 64, 7]. For example, Graff et al. [14] explored pool-based AL using uncertainty-based acquisition functions to accelerate virtual screening. Li and Rangarajan [26] proposed diversity-maximizing strategies for graph neural networks, improving chemical space exploration. Van Tilborg and Grisoni [55] systematically evaluated multiple AL strategies under low-data regimes by combining acquisition functions like uncertainty and diversity measures.

Despite these advancements, most AL methods rely on manually designed heuristics or static strategies, limiting their adaptability to dynamic and complex chemical spaces [10]. Recent general AL frameworks such as SelectAL [15] and AutoAL [57] have introduced task-specific optimizations. SelectAL adjusts AL strategies based on computational budgets, while AutoAL employs a differentiable framework to identify optimal strategies.

Other works have attempted to learn acquisition functions using reinforcement learning approaches, primarily based on Q-learning [11, 6]. While these methods have shown promise in domains like semantic and image segmentation, they face challenges in drug discovery, including sparse rewards and the high dimensionality of molecular representations. As a result, they still fall short in navigating vast chemical spaces efficiently under low-data conditions.

In contrast, our work leverages Group Relative Policy Optimization (GRPO) [49], which provides more stable and efficient policy updates compared to Q-learning. GRPO enables dynamic optimization of the acquisition policy, allowing better exploration and exploitation in complex and evolving chemical spaces. By directly addressing the limitations of existing methods, our approach significantly enhances hit discovery efficiency and scalability, particularly in low-data drug discovery scenarios.

3 Methodology

3.1 Problem Definition

Virtual screening aims to identify active molecules that interact with a biological target from a vast molecule library M. The library M contains a large number of unlabeled molecules, with only a small fraction being active. Each molecule $m_i \in M$ can be labeled as active ($y_i = 1$) or inactive ($y_i = 0$) after annotation. The challenge lies in maximizing the identification of active molecules while minimizing annotation costs.

To address this, active learning is employed. The process iteratively proceeds as follows:

1. **Train**: Train the surrogate model ϕ_θ using a small labeled dataset D.
2. **Query**: Apply a selection strategy ς to choose molecules m_{select} from the unlabeled library M.
3. **Annotate**: Label the selected molecules m_{select} using the oracle Ω, yielding m_{label}.
4. **Expand**: Add m_{label} to the labeled dataset D.

The iteration continues until the annotation budget is exhausted, with the final labeled dataset D expected to contain a substantial number of active molecules.

Traditional active learning methods rely on predefined selection strategies ς, often referred to as acquisition functions (e.g., greedy algorithms or mutual information maximization [18]). These fixed strategies unable to adapt to the characteristics of different datasets, which limits their generalizability and performance across diverse tasks. In this case, the selection strategy can be expressed as:

$$m_{\text{select}} = \arg\max\nolimits_{m \in M} \varsigma(m, \phi_\theta), \tag{1}$$

where $\varsigma(m, \phi_\theta)$ scores molecules based on a predefined, hand-crafted acquisition function and ϕ_θ is the surrogate model trained on the labeled dataset D.

To address these limitations, we directly replace the predefined selection strategy ς with a learnable policy network π_θ, i.e., $\varsigma = \pi_\theta$, which is capable of dynamically adapting to different datasets through training. The selection process in our method is expressed as:

$$m_{\text{select}} \sim \pi_\theta(m|D), \tag{2}$$

where $\pi_\theta(m|D)$ represents a probability distribution over the unlabeled molecules M conditioned on the labeled dataset D. Unlike fixed strategies, the policy network π_θ is continuously updated during the training step in active learning, adapting as $\pi_\theta \xrightarrow{D} \pi_{\theta'}$. This adaptability allows the selection strategy to dynamically adjust to different datasets and tasks.

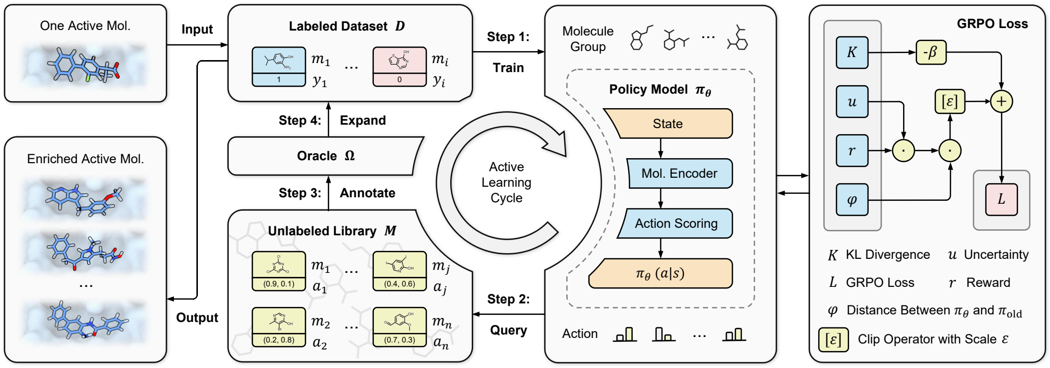

Figure 1: The overall framework of GLARE.

3.2 Overview of GLARE Framework

Figure 1 presents the framework of the proposed GLARE. Similar to traditional active learning methods, it also includes four steps: Train, Query, Annotate, and Expand. However, in the query stage, GLARE introduces a learnable policy model π_θ to adaptively select molecules for annotation after formulating a Markov Decision Process (MDP), addressing the diverse and complex nature of virtual screening tasks. The policy model π_θ consists of a Molecular Encoder, which extracts key features from the molecule library M, and an Action Scoring Layer, which selects molecules for annotation based on these features. To optimize the policy model and enhance overall performance, GLARE employs Group Relative Policy Optimization (GRPO), balancing structural diversity, biological relevance, and computational efficiency, ensuring adaptive and efficient molecular selection.

3.3 Reinforcement Learning for the Policy Model

The optimization task of the policy model π_θ is reformulated as a Markov Decision Process (MDP) within the framework of reinforced active learning-based virtual screening. In this setup, candidate molecules are divided into a series of groups, and during each active learning iteration, the policy model sequentially selects molecules for annotation. The MDP is defined by the following key components:

- **State:** The state s_i represents the embedding x_i of the molecule m_i in the group that currently being considered for annotation.
- **Action:** The action $a_i = [p^e, p^s]$ corresponds to the probabilities of "exclude" (p^e) and "select" (p^s) for a molecule m_i. The policy model π_θ determines the action for each molecule based on its state s_i, denoted as $\pi_\theta(a_i|s_i)$.
- **Reward:** The reward evaluates the quality of actions, balancing two critical aspects:

  (1) **Exploitation**: Reflects the immediate utility of the selected molecules, specifically whether they are active.

  (2) **Exploration**: Captures the long-term benefit of selecting molecules, providing insights into the broader library, even for inactive molecules.

  The reward r_i of policy network for a molecule m_i is defined as:

  $$r_i = 1 - (y_i - a_{i,y_i})^2 \cdot u_i, \tag{3}$$

  where y_i represents the label of m_i provided by the oracle Ω, and u_i is a discount factor that adjusts the exploration contribution which will be discussed in Section 3.4.3.

By framing the optimization task in this manner, the algorithms of molecular selection and model training for GLARE are summarized in Algorithm 1 and Algorithm 2, respectively. After optimization in MDP, the policy model π_θ learns to balance structural diversity and biological relevance, enabling adaptive and efficient molecular selection.

Algorithm 1 Molecular Selection under MDP

```
Input: Unlabeled library $M$, Budget $B$
Output: Labeled dataset $D$
while $|B| \neq 0$ do
    Divide $M$ into groups $G$
    for $j \in \{1, 2, ..., |G|\}$ do
        $\mathbf{x}_j \xleftarrow{\text{embed}} \{m_1, ..., m_i\} \in G_j$
        Construct state $\mathbf{s}_j$ using $\mathbf{x}_j$
        $\mathbf{a} = \{a_1, ..., a_j - 1\} + \pi_\theta(\mathbf{s}_j)$
    end for
    Sample $m_{\text{select}}$ with $\mathbf{a}$
    $m_{\text{label}} = \Omega(m_{\text{select}}; B)$
    $D \leftarrow D + m_{\text{select}}$
end while
return the latest labeled dataset $D$
```

Algorithm 2 Policy Model Training with GRPO

```
Input: Labeled dataset $D$
Output: Trained policy model $\pi_\theta$
for episode $\in \{1, 2, ..., N\}$ do
    Divide $D$ into groups $G$
    for $j \in \{1, 2, ..., |G|\}$ do
        $\mathbf{x}_j \xleftarrow{\text{embed}} \{m_1, ..., m_i\} \in G_j$
        Construct state $\mathbf{s}_j$ using $\mathbf{x}_j$
        $\mathbf{a}_j = \pi_\theta(\mathbf{s}_j)$
        Receive reward $\mathbf{r}_j$ after taken $\mathbf{a}_j$
        Do gradient descent step on $\mathscr{L}_{\text{GRPO}}$
        Update $\pi_\theta$ with $\theta$
    end for
end for
return the latest policy model $\pi_\theta$
```

3.4 Policy Model for Molecular Selection

The policy model π_θ is responsible for determining the probabilities of actions based on the molecular features extracted by the encoder, allowing GLARE to adaptively select molecules for annotation. Given the molecular embedding x_i as state s_i, the policy network predicts the probability of action a_i, denoted as $\pi_\theta(a_i|s_i)$. The policy model consists of Molecular Encoder (Section 3.4.1) and Action Scoring Layer (Section 3.4.2), optimized with GRPO strategy (Section 3.4.3).

3.4.1 Molecular Encoder

To enable the policy model π_θ to take actions based on molecules, a molecular encoder f_{enc} is utilized to extract features h_i from the input molecules. The encoder f_{enc} is a flexible component that can adapt to various molecular representations. In this study, we consider three types of f_{enc}:

- **Molecular Fingerprint MLP**. Each molecule is represented using a 1,024-bit Extended Connectivity Fingerprint (ECFP) with a radius of 2 Å [44]. These fingerprints are processed by a MultiLayer Perceptron (MLP) to extract molecular features h_i:
$$h_i = \sigma(W_m \cdot x_i + b_m), \tag{4}$$
where σ is the activation function, and W_m and b_m are learnable parameters of the MLP.
- **Graph Neural Network (GNN)**. For molecules represented as graph structures, a Graph Isomorphism Network (GIN) [59] is implemented. The GIN updates the features of each atom x_{ij} in the molecule m_i and an add pooling operation aggregates the features into a molecular hidden feature h_i:
$$x_{ij}^{(l)} = W_g \left((1 + \varepsilon) \cdot x_{ij}^{(l-1)} + \sum_{k \in \mathcal{N}(j)} x_{ik}^{(l-1)} \right), \quad h_i = \sum_{j \in \mathcal{V}(i)} x_{ij}^{(l)}, \tag{5}$$
where W_g is a learnable parameter, ε is a fixed scaling factor, $\mathcal{N}(j)$ denotes the neighbors of atom j and $\mathcal{V}(i)$ denotes the set of all atoms in molecule m_i.
- **Pretrained GNN**. Pretrained molecular representation models, such as GraphMVP [27] and 3DInfomax [51], can also be adopted as molecular encoders. These pretrained models leverage prior knowledge from large molecular datasets to provide robust initializations, thereby improving training efficiency and enhancing the quality of molecular representations.

3.4.2 Action Scoring Layer

Extracted by the encoder, the hidden feature h_i is passed through a fully connected layer to compute an action score z_i for each possible action (e.g., "exclude" or "select"):
$$z_i = W_a \cdot h_i + b_a, \tag{6}$$
where W_a and b_a are learnable parameters of the action scoring layer.

The scores z_i are then transformed into probabilities using a softmax function to gain action a_i:

$$a_i = \text{softmax}(z_i). \tag{7}$$

This transformation ensures that the output is a valid probability distribution, where the sum of probabilities for all possible actions equals to 1. The probability distribution represents the confidence of model in selecting or excluding a specific molecule.

3.4.3 Training Strategy Based on GRPO

The goal of the policy model π_θ in active learning is to identify the most informative molecules from a group of candidate molecules, which naturally aligns with the objective of the widely used Group Relative Policy Optimization (GRPO) strategy [49]. Thus, we apply an enhanced GRPO to train the policy model π_θ with a carefully designed reward function mentioned in Equation 3 based on the requirements of active learning. We further enhance the reward by incorporating uncertainty, ensuring better exploration during molecular selection.

Objective of GRPO in Reinforced Active Learning During the training phase of active learning, the GRPO strategy is adopted to optimize the policy model π_θ, maximizing the following objective over a group of candidate molecules G:

$$\mathscr{L}_{\text{GRPO}} = \frac{1}{|G|}\sum_{i=1}^{|G|}\left\{\min\left[\frac{\pi_\theta(a_i|s_i)}{\pi_{\text{old}}(a_i|s_i)}A_i, \text{clip}\left(\frac{\pi_\theta(a_i|s_i)}{\pi_{\text{old}}(a_i|s_i)}, 1-\epsilon, 1+\epsilon\right)A_i\right] - \beta\mathbb{D}_{\text{KL}}[\pi_\theta||\pi_{\text{ref}}]\right\}, \tag{8}$$

where π_{ref}, π_{old}, and π_θ are the policy model in previous active learning iteration, previous training epoch, and current epoch, respectively. clip$(\cdot)$ is a clip operation for stabilizing training and ϵ is the clip ratio [47]. The definition of advantage A_i and KL divergence $\mathbb{D}_{\text{KL}}[\cdot]$ as follows:

$$A_i = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}, \quad \mathbb{D}_{\text{KL}}[\pi_\theta||\pi_{\text{ref}}] = \frac{\pi_{\text{ref}}(a_i|s_i)}{\pi_\theta(a_i|s_i)} - \log\frac{\pi_{\text{ref}}(a_i|s_i)}{\pi_\theta(a_i|s_i)} - 1, \tag{9}$$

where $\mathbf{r}$ is the reward vector of the group of candidate molecules G.

The GRPO objective ensures that the policy model π_θ focuses on selecting the most informative molecules while maintaining training stability. Additionally, the KL divergence term prevents overfitting, especially when meeting the limited available training data in active learning due to the annotation budget.

Exploration Modification Although the sampling process of the policy model π_θ already incorporates randomness (as discussed in Section 3.4.2), explicitly considering the value of exploration in the reward function further enhances its ability to avoid aimless or uninformative exploration [5]. This is achieved by introducing gradient-based uncertainty to estimate the exploration value and designing a discount factor u_i to modify the reward.

The discount factor u_i is defined as:

$$u_i = \begin{cases} 1-\nu_i, & \text{if } \; y_i = 0 \; \text{ and } \; a_{i,y_i} = 1 \\ 1, & \text{if } \; y_i \neq 0 \end{cases}, \tag{10}$$

where ν is the normalized 2-norm of the gradient g^{y_i}, which is defined as:

$$||g^{y_i}||_2^2 = ||[g_0^{y_i}, g_1^{y_i}]||_2^2 = \left[[p_i^e - I(y_i = 0)]^2 + [p_i^s - I(y_i = 1)]^2\right] \cdot \left|\left|w(s_i)\right|\right|_2^2. \tag{11}$$

As defined in [2], molecules with higher uncertainty yield greater $[p_i^e - I(y_i = 0)]^2 + [p_i^s - I(y_i = 1)]^2$, resulting in greater 2-norm values of the gradient g^{y_i}. The detailed derivation is provided in the Appendix A.

By introducing the discount factor u_i, the model reduces the negative reward for inactive molecules with high uncertainty. This adjustment ensures such molecules can still contribute valuable information for model training and exploration despite being inactive.

Table 1: Enrichment Factor (EF) on ALDH1, PKM2 and VDR.

Method	Model	Strategy	ALDH1		PKM2		VDR	
			Iter 10	Iter 16	Iter 10	Iter 16	Iter 10	Iter 16
TcsAL [55]	MLP	Random	1.020	0.988	0.868	0.994	1.071	1.036
		Similarity	2.438	2.362	1.943	2.113	2.438	2.664
		Uncertainty	0.761	0.791	1.364	1.087	0.833	0.888
		Greedy	5.196	5.675	2.604	3.729	3.530	4.055
		MI	4.794	5.116	3.059	4.070	4.363	4.529
	GNN	Random	1.020	0.988	0.868	1.036	1.071	0.994
		Similarity	2.438	2.362	1.943	2.113	2.419	2.664
		Uncertainty	0.907	0.923	0.868	0.777	0.754	0.681
		Greedy	3.126	3.750	1.901	2.548	3.014	2.871
		MI	3.482	4.077	1.777	2.952	4.085	4.085
GLARE	MLP	Policy	6.574	6.535	2.067	5.904	3.173	6.512
	GNN	Policy	6.179	7.067	2.480	7.146	7.535	7.104
	Pre. GNN	Policy	**7.274**	**7.205**	**4.547**	**7.768**	**7.932**	**7.992**

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate GLARE on two widely used virtual screening benchmarks: LIT-PCBA [52] and Enamine [1]. For LIT-PCBA, we conduct experiments on the three subsets (ALDH1, PKM2, and VDR) with most experimentally validated molecules adhering to the protocols in [55]. Each subset is allocated a total annotation budget of 1% (1,000 molecules) of the library (100,000), with 64 molecules annotated per iteration. Enamine is a large-scale screening database with subsets named Enamine50k and EnamineHTS. the total annotation budget of Enamine50k is set as 6% (3,000) of the library (50,240), with 1% (500) per active learning iteration. Due to the large scale of EnamineHTS, the budget is reduced to 0.6% (12,600) and 1.2% (25,200) of the library (2,141,514), with 0.1% (2,100) and 0.2% (4,200) per iteration, noted as EnamineHTS-0.1 and EnamineHTS-0.2, respectively.

Baselines. We evaluate GLARE against two state-of-the-art active learning methods for virtual screening: TcsAL [55], with two model architectures and diverse acquisition functions, and PtAL [5], which leverages pretrained models like MoLFormer [45] and MolCLR [58]. Additional baselines include D-MPNN [17], RF [3], and LightGBM [24], with acquisition functions such as greedy, Mutual Information (MI) [18], Upper Confidence Bound (UCB), uncertainty, similarity, and random.

Metrics. Enrichment Factor (EF) is used to evaluate virtual screening performance, calculated as EF $= \frac{|D_{\text{active}}|}{|D|} / \frac{|M_{\text{active}}|}{|M|}$, where D_{active} and M_{active} represent active molecules in the selection and entire library, respectively. Retrieving Rate (RR$_n$) measures the ability to retrieve the highly active molecules, defined as RR$_n = \frac{|D_{\text{top-}n}|}{n}$, where $|D_{\text{top-}n}|$ is the number of top n active molecules selected.

4.2 Evaluation of Active Learning for Virtual Screening

4.2.1 Evaluation on General Benchmarks

We evaluate GLARE on general active learning virtual screening benchmarks using three subsets, ALDH1, PKM2, and VDR from LIT-PCBA, following the protocols in [55]. Table 1 reports the Enrichment Factor (EF) scores at the mid-phase (10th iteration) and final-phase (16th iteration). Results for all iterations are shown in Appendix B.1. GLARE consistently outperforms all baseline methods, achieving significant improvements. Notably, GLARE with a pretrained GNN achieves the highest EF scores of 7.205, 7.768, and 7.992 on ALDH1, PKM2, and VDR, reflecting improvements of 27.0%, 90.9%, and 76.5%, respectively, over the best baseline. Even without a pretrained model, GLARE with MLP and GNN significantly exceeds the best baseline in almost all scenarios, showcasing its strong performance across various molecular encoders. These findings highlight that learning a policy model for adaptive molecular selection strategies, rather than relying on predefined functions, substantially enhances performance.

Table 2: Retrieving Rate (RR_{500} and $\mathrm{RR}_{1,000}$) on Enamine50k and EnamineHTS. The upper part is the comparison among non-pretrained models, while the lower part corresponds to the pretrained.

Method	Model	Strategy	Enamine50k		EnamineHTS-0.1		EnamineHTS-0.2	
			Iter 4	Iter 6	Iter 4	Iter 6	Iter 4	Iter 6
-	RF [3]	Greedy	0.4316	0.5452	0.4360	0.5474	0.5802	0.6818
		UCB	0.2724	0.3708	0.2796	0.3582	0.4260	0.5096
	LightGBM [24]	Greedy	**0.5392**	0.6944	0.5218	0.6778	0.7018	0.8250
		UCB	0.4459	0.4088	0.4346	0.5614	0.5448	0.6278
	D-MPNN [17]	Greedy	0.4736	0.6532	0.5762	0.7276	0.7784	0.8988
		UCB	0.4863	0.6688	**0.5932**	0.7462	**0.8046**	0.8974
GLARE	GNN	Policy	0.4926	**0.7424**	0.4385	**0.7526**	0.7024	**0.9032**
PtAL [5]	MolCLR [58]	Greedy	0.5000	0.6708	0.5512	0.7278	0.7574	0.8698
		UCB	0.4972	0.6796	0.5384	0.7276	0.7624	0.8844
	MoLFormer [45]	Greedy	0.5812	0.7836	0.6742	0.8158	0.8534	0.9224
		UCB	0.6054	0.7924	0.6976	0.8412	0.8594	0.9338
GLARE	Pre. GNN	Policy	**0.7765**	**0.8869**	**0.8637**	**0.9181**	**0.9618**	**0.9732**

4.2.2 Evaluation on Large-scale Benchmarks

To further validate the scalability of GLARE, we conducted experiments on larger-scale virtual screening benchmarks. Table 2 presents the RR_n scores of various methods at the mid-phase (4th iteration) and final-phase (6th iteration) of active learning. More details are shown in Appendix B.2.

Among all the methods, GLARE with a pretrained GNN achieves the best performance, benefiting from faster initialization, which enables the model to effectively identify active molecules early in the active learning process. On Enamine50k, GLARE with a pretrained GNN achieves RR_{500} scores of 0.7765 and 0.8869 at the mid and final phases, representing improvements of 28.3% and 11.9% over the best baseline (PtAL with MoLFormer and UCB). Furthermore, GLARE demonstrates excellent efficiency on the larger EnamineHTS, achieving $\mathrm{RR}_{1,000}$ scores of 0.9181 (EnamineHTS-0.1) and 0.9732 (EnamineHTS-0.2), with substantial improvements over the best baseline. Although underperform at the beginning, GLARE with GNN surpasses all non-pretrained methods and PtAL with MolCLR at final-phase, despite MolCLR being a pretrained GNN, which further validate the superiority of our adaptive selection strategy. These experiments on large-scale benchmarks clearly demonstrate the scalability and effectiveness of GLARE for large-scale virtual screening tasks.

4.3 Improvement over Foundation Virtual Screening Methods

To further showcase the practicality of GLARE, we apply it to enhance the performance of virtual screening foundation models with a limited number of active molecules. The experiment is conducted on the entire LIT-PCBA dataset using the pretrained DrugCLIP [12], with a budget of n active molecules, denoted as GLARE(n).

Table 3 summarizes the results of various virtual screening methods on the LIT-PCBA. Impressively, the screening performance improves by 46.7% on $\mathrm{EF}_{0.5\%}$ even with the addition of just a single active molecule during active learning. With 15 additional active molecules, GLARE(15) achieves an 8-fold improvement in $\mathrm{EF}_{0.5\%}$ compared to the baseline model DrugCLIP. These results highlight that our method not only enables efficient virtual screening but also significantly enhances the capability of existing foundational virtual screening models with a minimal budget of active molecules.

4.4 Ablation Studies and Visualization

Ablation studies (Figure 2.a) demonstrate the effectiveness of the GRPO-based reinforced active learning in GLARE. GLARE w/o AL discards the active learning process, while GLARE w/o Policy adopts an active learning process but without reinforcement learning. Both of them perform significantly worse, even with a larger annotation budget. Experiments on batch size and total annotation budget reveal that medium batch sizes (64 and 128) strike the best balance between efficiency and performance, while excessively small or large batch sizes lead to trade-offs between screening time and effectiveness (Figure 2.b and c). More analysis is shown in Appendix B.4.

Table 3: AUROC, BEDROC and EF_{α} on LIT-PCBA

	AUROC(%)	BEDROC(%)	$\mathrm{EF}_{0.5\%}$	$\mathrm{EF}_{1\%}$	$\mathrm{EF}_{5\%}$
Surflex [22]	51.47	-	-	2.50	-
Glide-SP [16]	53.15	4.00	3.17	3.41	2.01
Planet [61]	57.31	-	4.64	3.87	2.43
Gnina [32]	60.93	5.40	-	4.63	-
DeepDTA [34]	56.27	2.53	-	1.47	-
BigBind [4]	60.80	-	-	3.82	-
DrugCLIP [12]	57.17	6.23	8.56	5.51	2.27
GLARE(1)	57.21	9.65	12.56	7.65	2.69
GLARE(3)	61.76	18.76	23.32	13.61	4.25
GLARE(10)	68.33	25.17	57.93	32.31	8.21
GLARE(15)	**70.17**	**28.49**	**77.03**	**40.64**	**9.94**

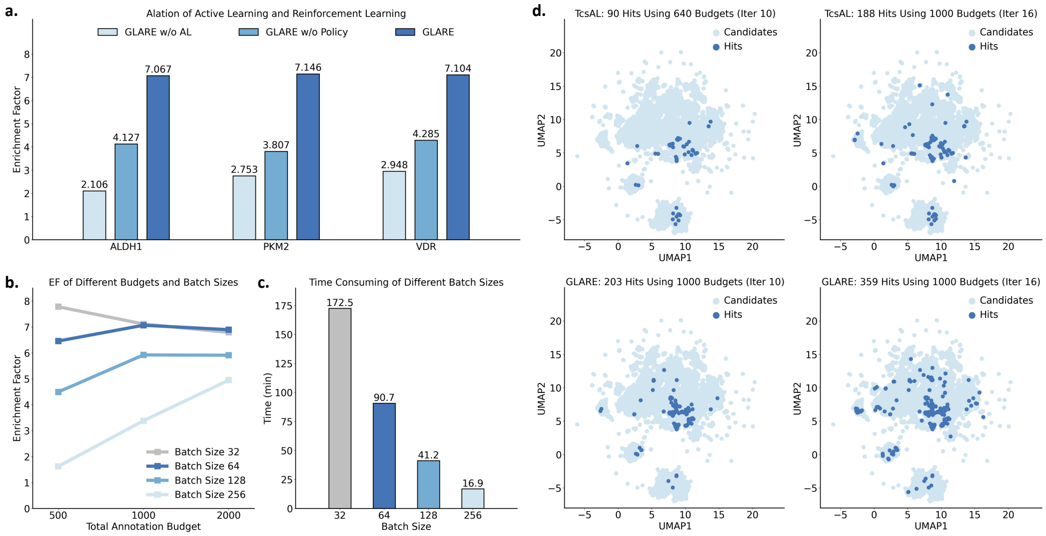

Figure 2: Results of ablation studies and visualization. **a**, Ablation study of GLARE for active learning and reinforcement learning. **b**, Enrichment Factor (EF) under different total annotation budgets and batch sizes for GLARE. **c**, Time consumption for different batch sizes of GLARE. **d**, Visualization results of the active learning selection process for TcsAL (upper) and GLARE (lower).

The UMAP [31] visualization (Figure 2.d) shows GLARE explores a broader and more diverse chemical space compared to TcsAL, achieving higher hit rates at both the mid-phase (10th iteration) and final-phase (16th iteration) of active learning. This highlights the superior adaptability and effectiveness of GLARE in discovering novel active molecules while maintaining high structural diversity.

5 Discussion

This work presents GLARE, an active learning-based virtual screening framework that uses a GRPO-based strategy to efficiently explore chemical space with a limited annotation budget. Experiments show the superior performance of GLARE, significantly improving $\mathrm{EF}_{0.5\%}$ and hit rates on datasets like LIT-PCBA and ALDH1, while visualizations highlight its ability to identify diverse and novel active molecules. GLARE enhances molecule discovery efficiency and reduces costs, making it a valuable tool for drug discovery. However, GLARE faces challenges when scaling to ultra-large chemical spaces (e.g., 10^{10} molecules) due to increased time costs, and its performance tends to decline as the annotation budget increases. Future work could focus on improving scalability and mitigating the diminishing returns associated with larger annotation budgets.

Acknowledgments and Disclosure of Funding

This study has been supported by the National Natural Science Foundation of China [T2394502], the Guangdong S&T Program [2024B1111140001], the National Natural Science Foundation of China [62302537], the China Postdoctoral Science Foundation [2025M771540, GZB20250391], the Guangzhou Basic and Applied Basic Research Foundation [2024A04J4449], and the Lingang Laboratory [LGL-8888].

References

[1] REAL Database - Enamine — enamine.net. `https://enamine.net/compound-collections/real-compounds/real-database`. [Accessed 07-05-2025].

[2] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. arXiv preprint arXiv:1906.03671, 2019.

[3] Leo Breiman. Random forests. Machine learning, 45:5–32, 2001.

[4] Michael Brocidiacono, Paul Francoeur, Rishal Aggarwal, Konstantin I Popov, David Ryan Koes, and Alexander Tropsha. Bigbind: learning from nonstructural data for structure-based virtual screening. Journal of Chemical Information and Modeling, 64(7):2488–2495, 2023.

[5] Zhonglin Cao, Simone Sciabola, and Ye Wang. Large-scale pretraining improves sample efficiency of active learning-based virtual screening. Journal of Chemical Information and Modeling, 64(6):1882–1891, 2024.

[6] Arantxa Casanova, Pedro O Pinheiro, Negar Rostamzadeh, and Christopher J Pal. Reinforced active learning for image segmentation. arXiv preprint arXiv:2002.06583, 2020.

[7] Gui Citovsky, Giulia DeSalvo, Claudio Gentile, Lazaros Karydas, Anand Rajagopalan, Afshin Rostamizadeh, and Sanjiv Kumar. Batch active learning at scale. Advances in Neural Information Processing Systems, 34:11933–11944, 2021.

[8] Xun Deng, Junlong Liu, Zhike Liu, Jiansheng Wu, Fuli Feng, Jieping Ye, and Zheng Wang. Improving the hit rates of virtual screening by active learning from bioactivity feedback. Journal of Chemical Theory and Computation, 21(9):4640–4651, 2025.

[9] Jeremy Desaphy, Eric Raimbaud, Pierre Ducrot, and Didier Rognan. Encoding protein–ligand interaction patterns in fingerprints and graphs. Journal of chemical information and modeling, 53(3):623–637, 2013.

[10] Michael Dodds, Jeff Guo, Thomas Löhr, Alessandro Tibo, Ola Engkvist, and Jon Paul Janet. Sample efficient reinforcement learning with active learning for molecular design. Chemical Science, 15(11):4146–4160, 2024.

[11] Meng Fang, Yuan Li, and Trevor Cohn. Learning how to active learn: A deep reinforcement learning approach. arXiv preprint arXiv:1708.02383, 2017.

[12] Bowen Gao, Bo Qiang, Haichuan Tan, Yinjun Jia, Minsi Ren, Minsi Lu, Jingjing Liu, Wei-Ying Ma, and Yanyan Lan. Drugclip: Contrastive protein-molecule representation learning for virtual screening. Advances in Neural Information Processing Systems, 36:44595–44614, 2023.

[13] Christoph Gorgulla, Andras Boeszoermenyi, Zi-Fu Wang, Patrick D Fischer, Paul W Coote, Krishna M Padmanabha Das, Yehor S Malets, Dmytro S Radchenko, Yurii S Moroz, David A Scott, et al. An open-source drug discovery platform enables ultra-large virtual screens. Nature, 580(7805):663–668, 2020.

[14] David E Graff, Eugene I Shakhnovich, and Connor W Coley. Accelerating high-throughput virtual screening through molecular pool-based active learning. Chemical science, 12(22): 7866–7881, 2021.

[15] Guy Hacohen and Daphna Weinshall. How to select which active learning strategy is best suited for your specific problem and budget. Advances in Neural Information Processing Systems, 36: 13395–13407, 2023.

[16] Thomas A Halgren, Robert B Murphy, Richard A Friesner, Hege S Beard, Leah L Frye, W Thomas Pollard, and Jay L Banks. Glide: a new approach for rapid, accurate docking and

scoring. 2. enrichment factors in database screening. Journal of medicinal chemistry, 47(7): 1750–1759, 2004.

[17] Esther Heid, Kevin P Greenman, Yunsie Chung, Shih-Cheng Li, David E Graff, Florence H Vermeire, Haoyang Wu, William H Green, and Charles J McGill. Chemprop: a machine learning package for chemical property prediction. Journal of Chemical Information and Modeling, 64 (1):9–17, 2023.

[18] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. arXiv preprint arXiv:1112.5745, 2011.

[19] Wei-Ning Hsu and Hsuan-Tien Lin. Active learning by learning. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 29, 2015.

[20] Ilia Igashov, Arian R Jamasb, Ahmed Sadek, Freyr Sverrisson, Arne Schneuing, Pietro Lio, Tom L Blundell, Michael Bronstein, and Bruno Correia. Decoding surface fingerprints for protein-ligand interactions. bioRxiv, pages 2022–04, 2022.

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

[22] Ajay N Jain. Surflex: fully automatic flexible molecular docking using a molecular similarity-based search engine. Journal of medicinal chemistry, 46(4):499–511, 2003.

[23] Anat Levit Kaplan, Danielle N Confair, Kuglae Kim, Ximena Barros-Álvarez, Ramona M Rodriguiz, Ying Yang, Oh Sang Kweon, Tao Che, John D McCorvy, David N Kamber, et al. Bespoke library docking for 5-ht2a receptor agonists with antidepressant activity. Nature, 610 (7932):582–591, 2022.

[24] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30, 2017.

[25] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. Pubchem 2019 update: improved access to chemical data. Nucleic acids research, 47(D1):D1102–D1109, 2019.

[26] Bowen Li and Srinivas Rangarajan. A diversity maximizing active learning strategy for graph neural network models of chemical properties. Molecular Systems Design & Engineering, 7 (12):1697–1706, 2022.

[27] Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. Pre-training molecular graph representation with 3d geometry. arXiv preprint arXiv:2110.07728, 2021.

[28] Wei Lu, Qifeng Wu, Jixian Zhang, Jiahua Rao, Chengtao Li, and Shuangjia Zheng. Tankbind: Trigonometry-aware neural networks for drug-protein binding structure prediction. Advances in neural information processing systems, 35:7236–7249, 2022.

[29] Wei Lu, Jixian Zhang, Weifeng Huang, Ziqiao Zhang, Xiangyu Jia, Zhenyu Wang, Leilei Shi, Chengtao Li, Peter G Wolynes, and Shuangjia Zheng. Dynamicbind: predicting ligand-specific protein-ligand complex structure with a deep equivariant generative model. Nature Communications, 15(1):1071, 2024.

[30] Eduardo Habib Bechelane Maia, Letícia Cristina Assis, Tiago Alves De Oliveira, Alisson Marques Da Silva, and Alex Gutterres Taranto. Structure-based virtual screening: from classical to artificial intelligence. Frontiers in chemistry, 8:343, 2020.

[31] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426, 2018.

[32] Andrew T McNutt, Paul Francoeur, Rishal Aggarwal, Tomohide Masuda, Rocco Meli, Matthew Ragoza, Jocelyn Sunseri, and David Ryan Koes. Gnina 1.0: molecular docking with deep learning. Journal of cheminformatics, 13(1):43, 2021.

[33] Maruti Naik, Anandkumar Raichurkar, Balachandra S Bandodkar, Begur V Varun, Shantika Bhat, Rajesh Kalkhambkar, Kannan Murugan, Rani Menon, Jyothi Bhat, Beena Paul, et al. Structure guided lead generation for m. tuberculosis thymidylate kinase (mtb tmk): discovery of 3-cyanopyridone and 1, 6-naphthyridin-2-one as potent inhibitors. Journal of medicinal chemistry, 58(2):753–766, 2015.

[34] Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. Deepdta: deep drug–target binding affinity prediction. Bioinformatics, 34(17):i821–i829, 2018.

[35] Kunkun Pang, Mingzhi Dong, Yang Wu, and Timothy M Hospedales. Dynamic ensemble active learning: A non-stationary bandit with expert advice. In 2018 24th International Conference on Pattern Recognition (ICPR), pages 2269–2276. IEEE, 2018.

[36] Hitesh Patel, Wolf-Dietrich Ihlenfeldt, Philip N Judson, Yurii S Moroz, Yuri Pevzner, Megan L Peach, Victorien Delannée, Nadya I Tarasova, and Marc C Nicklaus. Savi, in silico generation of billions of easily synthesizable compounds through expert-system type rules. Scientific data, 7(1):384, 2020.

[37] Jiahua Rao, Dahao Xu, Wentao Wei, Yicong Chen, Mingjun Yang, and Yuedong Yang. Quadruple attention in many-body systems for accurate molecular property predictions. In Forty-second International Conference on Machine Learning.

[38] Jiahua Rao, Shuangjia Zheng, Yutong Lu, and Yuedong Yang. Quantitative evaluation of explainable graph neural networks for molecular property prediction. Patterns, 3(12), 2022.

[39] Jiahua Rao, Shuangjia Zheng, Sijie Mai, and Yuedong Yang. Communicative subgraph representation learning for multi-relational inductive drug-gene interaction prediction. In Lud De Raedt, editor, Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22, pages 3919–3925. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/544. URL `https://doi.org/10.24963/ijcai.2022/544`. Main Track.

[40] Jiahua Rao, Jiancong Xie, Qianmu Yuan, Deqin Liu, Zhen Wang, Yutong Lu, Shuangjia Zheng, and Yuedong Yang. A variational expectation-maximization framework for balanced multi-scale learning of protein and drug interactions. Nature Communications, 15(1):4476, 2024.

[41] Daniel Reker. Practical considerations for active machine learning in drug discovery. Drug Discovery Today: Technologies, 32:73–79, 2019.

[42] Daniel Reker and Gisbert Schneider. Active-learning strategies in computer-assisted drug discovery. Drug discovery today, 20(4):458–465, 2015.

[43] Lars Richter, Chris De Graaf, Werner Sieghart, Zdravko Varagic, Martina Mörzinger, Iwan JP De Esch, Gerhard F Ecker, and Margot Ernst. Diazepam-bound gabaa receptor models identify new benzodiazepine binding-site ligands. Nature chemical biology, 8(5):455–464, 2012.

[44] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. Journal of chemical information and modeling, 50(5):742–754, 2010.

[45] Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Payel Das. Large-scale chemical language representations capture molecular structure and properties. Nature Machine Intelligence, 4(12):1256–1264, 2022.

[46] Arman A Sadybekov, Anastasiia V Sadybekov, Yongfeng Liu, Christos Iliopoulos-Tsoutsouvas, Xi-Ping Huang, Julie Pickett, Blake Houser, Nilkanth Patel, Ngan K Tran, Fei Tong, et al. Synthon-based ligand discovery in virtual libraries of over 11 billion compounds. Nature, 601 (7893):452–459, 2022.

[47] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

[48] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. arXiv preprint arXiv:1708.00489, 2017.

[49] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300, 2024.

[50] Ao Shen, Mingzhi Yuan, Jie Du, Qiao Huang, Manning Wang, et al. Drug-tta: Test-time adaptation for drug virtual screening via multi-task meta-auxiliary learning. In Forty-second International Conference on Machine Learning.

[51] Hannes Stärk, Dominique Beaini, Gabriele Corso, Prudencio Tossou, Christian Dallago, Stephan Günnemann, and Pietro Liò. 3d infomax improves gnns for molecular property prediction. In International Conference on Machine Learning, pages 20479–20502. PMLR, 2022.

[52] Viet-Khoa Tran-Nguyen, Célien Jacquemard, and Didier Rognan. Lit-pcba: an unbiased data set for machine learning and virtual screening. Journal of chemical information and modeling, 60(9):4263–4273, 2020.

[53] Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. Journal of computational chemistry, 31(2):455–461, 2010.

[54] Jean-François Truchon and Christopher I Bayly. Evaluating virtual screening methods: good and bad metrics for the "early recognition" problem. Journal of chemical information and modeling, 47(2):488–508, 2007.

[55] Derek Van Tilborg and Francesca Grisoni. Traversing chemical space with active deep learning for low-data drug discovery. Nature Computational Science, 4(10):786–796, 2024.

[56] Marcel L Verdonk, Jason C Cole, Michael J Hartshorn, Christopher W Murray, and Richard D Taylor. Improved protein–ligand docking using gold. Proteins: Structure, Function, and Bioinformatics, 52(4):609–623, 2003.

[57] Yifeng Wang, Xueying Zhan, and Siyu Huang. Autoal: Automated active learning with differentiable query strategy search. arXiv preprint arXiv:2410.13853, 2024.

[58] Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. Nature Machine Intelligence, 4(3): 279–287, 2022.

[59] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? arXiv preprint arXiv:1810.00826, 2018.

[60] Jie Yu, Xutong Li, and Mingyue Zheng. Current status of active learning for drug discovery. Artificial Intelligence in the Life Sciences, 1:100023, 2021.

[61] Xiangying Zhang, Haotian Gao, Haojie Wang, Zhihang Chen, Zhe Zhang, Xinchong Chen, Yan Li, Yifei Qi, and Renxiao Wang. Planet: a multi-objective graph neural network model for protein–ligand binding affinity prediction. Journal of Chemical Information and Modeling, 64 (7):2205–2220, 2023.

[62] Shuangjia Zheng, Yongjian Li, Sheng Chen, Jun Xu, and Yuedong Yang. Predicting drug–protein interaction using quasi-visual question answering system. Nature Machine Intelligence, 2(2):134–140, 2020.

[63] Shuangjia Zheng, Jiahua Rao, Ying Song, Jixian Zhang, Xianglu Xiao, Evandro Fei Fang, Yuedong Yang, and Zhangming Niu. Pharmkg: a dedicated knowledge graph benchmark for bomedical data mining. Briefings in bioinformatics, 22(4):bbaa344, 2021.

[64] Jia-Jie Zhu and José Bento. Generative adversarial active learning. arXiv preprint arXiv:1702.07956, 2017.

A Proof for Uncertainty Estimation Based on Gradient

The policy model can be simply modeled as $\pi_\theta(a_i|s_i) = W \cdot w(s_i) + b$, then cross entropy loss between $a_i = [p_i^e, p_i^s]$ and y_i is:

$$\begin{aligned} l_{CE}(a_i, y_i) &= I(y_i = 0) \cdot \ln(\frac{1}{p_i^e}) + I(y_i = 1) \cdot \ln(\frac{1}{p_i^s}) \\ &= \ln(e^{W_0 \cdot w(s_i)} + e^{W_1 \cdot w(s_i)}) - W_{y_i} \cdot w(s_i). \end{aligned} \tag{12}$$

We define $g_k^{y_i} = \frac{\partial}{\partial W_k} l_{CE}(a_i, y_i)$ as the gradients corresponding to y_k:

$$\begin{aligned} g_k^{y_i} &= \frac{1}{e^{W_0 \cdot w(s_i)} + e^{W_1 \cdot w(s_i)}} \cdot e^{W_k \cdot w(s_i)} \cdot w(s_i) - I(y_i = k) \cdot w(s_i) \\ &= [a_{i,y_i} - I(y_i = k)] \cdot w(s_i). \end{aligned} \tag{13}$$

Replacing y_i with $\hat{y}_i = \arg\max(a_i)$, then the 2-norm of whole gradient $g^{\hat{y}_i} = \frac{\partial}{\partial W} l_{CE}(a_i, \hat{y}_i)$ is:

$$||g^{\hat{y}_i}||_2^2 = ||[g_0^{\hat{y}_i}, g_1^{\hat{y}_i}]||_2^2 = \left[[p_i^e - I(\hat{y}_i = 0)]^2 + [p_i^s - I(\hat{y}_i = 1)]^2\right] \cdot \left|\left|w(s_i)\right|\right|_2^2. \tag{14}$$

Evidently, high-uncertainty samples yield greater $\left[[p_i^e - I(\hat{y}_i = 0)]^2 + [p_i^s - I(\hat{y}_i = 1)]^2\right]$ since $\hat{y}_i = \arg\max(a_i)$, i.e., the length of $g^{\hat{y}_i}$ is long. Therefore, samples with high uncertainty tend to have a great 2-norm of the gradient.

B Additional Results

B.1 Detailed Results of General Benchmarks

Table 4 presents the complete results of GLARE on three LIT-PCBA subsets: ALDH1, PKM2, and VDR, covering a total of 16 iterations, with an annotation budget of 64 allocated for each iteration.

Table 4: Enrichment Factor (EF) on ALDH1, PKM2 and VDR of each Iteration.

Iter	ALDH1			PKM2			VDR		
	MLP	GNN	Pre. GNN	MLP	GNN	Pre. GNN	MLP	GNN	Pre. GNN
1	0.483	0.483	0.483	1.754	1.754	1.754	0.869	0.869	0.869
2	1.910	1.500	1.364	1.559	1.559	1.559	0.768	0.768	2.303
3	3.802	3.516	2.566	1.404	1.404	1.404	0.687	1.374	2.062
4	4.303	3.573	3.792	1.277	1.915	1.277	1.866	3.110	3.110
5	4.496	4.260	4.674	1.170	1.756	1.170	1.705	3.978	3.978
6	4.877	4.330	5.873	1.081	1.621	1.081	2.092	5.753	5.230
7	5.540	4.510	6.614	1.505	1.505	2.007	2.422	6.781	6.297
8	5.855	5.024	6.875	1.405	1.405	1.873	2.255	7.218	6.315
9	6.135	5.764	7.045	2.196	2.196	2.635	2.533	7.598	6.754
10	6.574	6.179	7.274	2.067	2.480	4.547	3.173	7.535	7.932
11	6.797	6.631	7.352	1.952	3.904	5.076	4.114	7.854	8.602
12	6.754	6.780	7.289	2.219	4.809	6.288	5.662	8.138	8.846
13	6.765	6.812	7.212	2.811	5.974	7.029	5.372	8.058	8.394
14	6.818	6.840	7.168	3.013	7.029	7.364	6.069	7.666	7.985
15	6.638	7.028	7.335	4.154	7.030	7.349	6.701	7.310	7.919
16	6.535	7.067	7.205	5.904	7.146	7.768	6.512	7.104	7.992

B.2 Detailed Results of Large-scale Benchmarks

Table 5 presents the complete results of GLARE on Enamine50k and EnamineHTS, covering a total of 6 iterations. For Enamine50k, the annotation budget is 1% (500) of the library per active learning iteration. For EnamineHTS, the annotation budget is 0.6% (12,600) and 1.2% (25,200) of the library per active learning iteration, noted as EnamineHTS-0.1 and EnamineHTS-0.2, respectively.

Table 5: Retrieving Rate (RR_{500} and $RR_{1,000}$) on Enamine50k and EnamineHTS of each Iteration.

Iter	Enamine50k		EnamineHTS-0.1		EnamineHTS-0.2	
	GNN	Pre. GNN	GNN	Pre. GNN	GNN	Pre. GNN
1	0.0162	0.0167	0.0010	0.0010	0.0010	0.0010
2	0.1325	0.2340	0.0194	0.7150	0.0473	0.7011
3	0.3463	0.5829	0.2595	0.8197	0.3955	0.9435
4	0.4926	0.7765	0.4385	0.8637	0.7024	0.9618
5	0.6388	0.8387	0.6359	0.8987	0.8314	0.9664
6	0.7424	0.8869	0.7526	0.9181	0.9032	0.9732

B.3 Detailed Results with Standard Deviations

To further validate the stability and robustness of the proposed method, Table 6 presents the detailed experimental results of GLARE on ALDH1, PKM2, and VDR, including the mean performance metrics and their standard deviations (Mean ± STD). All results were obtained from multiple independent runs to ensure the reliability and statistical significance of the evaluation.

Table 6: Enrichment Factor (EF) and of GLARE on ALDH1 at final-phase (16th iteration).

	ALDH1	PKM2	VDR
MLP	6.535 ± 0.182	5.904 ± 1.312	6.512 ± 0.674
GNN	7.067 ± 0.384	7.146 ± 1.105	7.104 ± 0.358
Pre. GNN	7.205 ± 0.124	7.768 ± 0.926	7.992 ± 0.425

B.4 Supplement of Ablation Studies

To investigate the effectiveness of active learning, we train two variant models, denoted as GLARE w/o AL and GLARE w/o Policy. Considering maintaining the same size of training data, we design GLARE w/o AL with 1000 labeled samples to start and an additional 1000 budget for screening. GLARE w/o Policy adopts an active learning process but without reinforcement learning. As Table 7 shows, although using twice the annotation budget, GLARE w/o AL performs the worst, which can be attributed to the absence of active learning, leading to a substantial waste of annotation resources. In addition, GLARE w/o Policy also underperforms compared to GLARE due to the lack of an adaptive molecular selection strategy. These results demonstrate the advantage of active learning-based virtual screening over conventional one-shot screening.

Table 7: EF of ablation study for active learning and reinforcement learning.

	ALDH1	PKM2	VDR
REAL w/o AL	2.106	2.753	2.948
REAL w/o Policy	4.127	3.807	4.285
REAL	7.067	7.146	7.104

Table 8: EF of ablation study under different annotation budgets and batch sizes on ALDH1.

	32	64	128	256
500	7.781	6.459	4.495	1.624
1000	7.114	7.067	5.923	3.385
2000	6.792	6.896	5.911	4.949

We investigate the effects of annotation budget per iteration (i.e., batch size) and total budget in active learning, using batch sizes of 32 / 64 / 128 / 256 and budgets of 500 / 1,000 / 2,000. As shown in Table 8, EFs for batch sizes 64 and 128 initially increase but slightly decline as the budget grows. The initial increasement can be attributed to the reason that larger budget facilitates better model training and allocates more resources for screening. But the subsequent slight decline occurs because the easily identifiable active molecules are quickly selected in the earlier iterations, leaving behind more challenging ones. Due to the large batch size of 256, only a few active learning iterations can be operated, resulting in inferior performance compared to others. EF of batch size 32 is the greatest when annotation budget comes to 500, because smaller batch size allows the model to update without delay and avoid wasting the screening budget. However, this comes at the cost of doubled screening time (Figure 2.c), with minimal EF improvement when the annotation budget exceeds 1000.

B.5 Time Consumption

To provide a comprehensive comparison, we evaluated both training and inference runtimes on the ALDH1 dataset, using the same hyperparameters and hardware as TcsAL [55].

Active learning typically involves training on a small number of labeled samples while making predictions on a much larger pool of unlabeled data. As a result, inference time becomes the dominant computational cost, especially in large-scale virtual screening scenarios. As shown in the Table 9, while GLARE requires extra training time due to the computational cost of gradient-based uncertainty in reward calculation, this overhead is confined to the training phase. During inference, reward computation is not required, enabling GLARE to achieve inference speeds comparable to TcsAL.

Table 9: The runtime of training and inference on ALDH1.

Method	Model	Training Time (Sec. per Epoch)	Inference Time (Sec.)
TcsAL	MLP	1.5	18.2
	GNN	4.2	148.6
GLARE	MLP	4.7	14.5
	GNN	8.9	152.6

The characteristic is particularly important for large-scale dataset. To further demonstrate the scalability of our approach, we tested it on the AmpC dataset (10^8 compounds) using the same hyperparameters as PtAL[5]. As shown in the Table 10, GLARE achieves inference speeds comparable to the baseline PtAL across both medium (EnamineHTS-0.1, 2 million compounds) and ultra-large (AmpC, 99.5 million compounds) datasets. As dataset size increases, inference time becomes even more significant. Our method maintains inference speed similar to baseline methods while delivering much higher accuracy, making it especially well-suited for ultra-large virtual screening tasks.

Table 10: The runtime of training and inference on EnamineHTS and AmpC.

Method	Model	EnamineHTS-0.1 (2m)		AmpC (99.5m)	
		Train Time (Sec.)	Infer Time (Sec.)	Train Time (Sec.)	Infer Time (Sec.)
PtAL	GNN	41.7	825.5	2274.6	40423.3
	Pre. GNN	68.3	922.3	3861.5	45374.8
GLARE	GNN	84.2	913.3	4868.2	49662.4
	Pre. GNN	155.9	1179.1	8237.6	55127.2

B.6 Detailed Results of Training Loss

Regarding training convergence, we ensured sufficient training within each active learning round. Following the TcsAL baseline, we trained each model for 50 epochs per iteration and monitored the training loss. As shown in the Table 11, the loss curves for MLP, GNN, and Pre. GNN all reach convergence within each iteration, confirming that our models are well-trained.

Table 11: The loss of model training on ALDH1.

Epoch	MLP		GNN		Pre. GNN	
	Iter 4	Iter 6	Iter 4	Iter 6	Iter 4	Iter 6
1	0.4328	0.4204	0.4548	0.555	0.4858	0.5183
2	0.1292	0.1828	0.1717	0.2749	0.2854	0.1897
5	0.0225	0.0181	0.0478	0.0871	0.0351	0.0181
10	0.0016	0.0097	0.0062	0.0158	0.0039	0.0036
20	0.0027	0.0023	0.0112	0.0092	0.0185	0.0256
50	0.0003	0.0009	0.0071	0.0029	0.0006	0.0221

B.7 Comparison for Different Pretrained GNN

To further isolate the effect of the learning strategy, we compared GLARE (with MolCLR) and PtAL (also with MolCLR) under same pretrained GINs(i.e. MolCLR), shown in the Table 12. GLARE consistently outperforms PtAL, showing that the performance gain is primarily due to GLARE's superior learning strategy rather than the encoder alone.

Table 12: The comparison for different pretrained GIN encoders.

Method	Model	Strategy	Enamine50k		EnamineHTS-0.1		EnamineHTS-0.2	
			Iter 4	Iter 6	Iter 4	Iter 6	Iter 4	Iter 6
PtAL	MolCLR	Greedy	0.5000	0.6708	0.5512	0.7278	0.7574	0.8698
		UCB	0.4972	0.6796	0.5384	0.7276	0.7624	0.8844
GLARE	GNN	Policy	0.4926	0.7424	0.4385	0.7526	0.7024	0.9032
	MolCLR		0.7695	0.8652	0.8425	0.8814	0.9356	0.9527
	Pre. GNN		**0.7765**	**0.8869**	**0.8637**	**0.9181**	**0.9618**	**0.9732**

B.8 Different Optimization Strategies and Virtual Screening Methods

We experimented with Direct Policy Optimization (DPO), another effective RL algorithm, for optimizing the policy network. As shown in the Table 13, GRPO consistently outperforms DPO in our setting. We attribute this to the utilization of group-wise advantage estimation in GRPO, which better captures the relative quality of selected molecules within a batch (a property that aligns well with the objectives of virtual screening). This provides a direct motivation for our choice of GRPO.

Table 13: The result for different reinforcement learning optimization strategies.

Method	Model	Strategy	ALDH1		PKM2		VDR	
			Iter 10	Iter 16	Iter 10	Iter 16	Iter 10	Iter 16
GLARE	MLP	Policy (DPO)	5.759	6.012	1.878	5.526	2.397	5.961
		Policy (GRPO)	6.574	6.535	2.067	5.904	3.173	6.512
	GNN	Policy (DPO)	4.892	6.743	1.927	6.491	3.618	6.635
		Policy (GRPO)	6.179	7.067	2.480	7.146	7.535	7.104
	Pre. GNN	Policy (DPO)	6.387	6.834	4.166	7.394	7.293	7.590
		Policy (GRPO)	7.274	7.205	4.547	7.768	7.932	7.992

Recently TTA has been applied in virtual screening, which typically leverages self-supervised auxiliary tasks during inference to dynamically update the model parameters, enabling the learning of feature representations tailored to individual test instances. This approach requires no prior knowledge of the test data distribution and instead adjusts model parameters dynamically on a per-instance basis during inference. Experiments about the comparison with DrugTTA[50] baseline are conducted. The results are summarized is Table 14.

Table 14: The comparison for TTA baseline.

Method	Strategy	AUCROC(%)	BEDROC(%)	EF0.5%	EF1%	EF5%
DrugCLIP	-	57.17	6.23	8.56	5.51	2.27
GLARE(20)	MI	65.08	29.75	37.85	21.34	6.41
GLARE(20)	DPO	70.36	38.03	65.53	37.63	9.61
DrugTTA	-	71.24	45.08	74.39	42.74	10.61
GLARE(18)	GRPO	75.63	41.55	80.8	44.05	10.32
GLARE(20)	GRPO	79.78	46.39	83.51	47.36	12.08

C Additional Details

C.1 Overview of the Benchmarks

LIT-PCBA PCBA includes a total of 15 targets, comprising 7,844 active and 407,381 inactive compounds. Following [55], we selected the three LIT-PCBA datasets with the highest numbers of experimentally validated molecules, which correspond to targets of clinical and therapeutic interest: pyruvate kinase M2 (PKM2, agonism), aldehyde dehydrogenase 1 (ALDH1, inhibition), and vitamin D receptor (VDR, antagonism). For each dataset, 100,000 molecules were randomly sampled while maintaining the ratio of active to inactive compounds, which were used to construct a screening library. For details, refer to Table 15.

Table 15: Summary of the three subsets from LIT-PCBA (PKM2, ALDH1 and VDR).

Dataset	Screening Library Size	Hits Size	Hits Ratio	Test Set Size	Hits Size	Hits Ratio
ALDH1	100,000	4,986	5%	20,000	997	5%
PKM2	100,000	223	0.2%	20,000	44	0.2%
VDR	100,000	239	0.2%	20,000	48	0.2%

Enamine Enamine is a commercially available database for large-scale screening. The Enamine Discovery Diversity Set (Enamine50k) and Enamine HTS collection (EnamineHTS) consist of 50,240 compounds and 2,141,514 molecules, respectively. The Enamine datasets used for these studies were generated from docking the compounds against thymidylate kinase (PDB ID: 4UNN). Both Enamine50k and EnamineHTS are publicly accessible via the MolPAL code repository[14], undergo molecular docking against thymidylate kinase (PDB ID: 4UNN)[33] using AutoDockVina[53].

C.2 Baseline Methods

Baselines We consider two challenging baselines for a thorough evaluation, which contain a range of surrogate models and acquisition functions as traditional active learning methods. Van Tilborg and Grisoni [55] proposes an active learning virtual screening baseline containing two model architectures and some acquisition functions (for convenience, we name it as TcsAL). Cao et al. [5] conducted experiments on large-scale datasets and proposed a large-scale active learning virtual screening baseline incorporating pretrained models (for convenience, we name it as PtAL).

Pretrained Models The pretrained models used in PtAL include MoLFormer [45] and MolCLR [25]. MoLFormer is a transformer-based model that takes tokenized SMILES strings as input and learns molecular representations by capturing intrinsic spatial relationships between atoms. To improve computational efficiency, it employs linear attention and rotary position embedding instead of standard quadratic attention. MoLFormer was pretrained on an ultralarge dataset of 1.1 billion small molecules from ZINC and PubChem using the mask-language-modeling technique. MolCLR is a graph isomorphism network pretrained on 10 million molecules from PubChem [25] using a contrastive learning strategy. Other comparative models include D-MPNN [17], a GNN variant, as well as RF [3] and LightGBM [24], which are decision tree-based ensemble methods.

Acquisition Functions Acquisition functions used for comparison comprise greedy, mutual information (MI) [18], upper confidence bound (UCB), uncertainty, similarity, and random.

To estimate the expected value $\mathbb{E}$ of a molecule m_i, the mean prediction across all surrogate models in the ensemble is considered when performing traditional active learning-based virtual screening:

$$\mathbb{E}(y_i|m_i) = \frac{1}{K}\sum_{k=1}^{K} p_k(y_i|m_i). \tag{15}$$

Similarly, the prediction uncertainty for a molecule m_i is defined as the mean entropy $\mathbb{H}$ over the ensemble:

$$\mathbb{H}\left(y_i|m_i\right) = -\frac{1}{K}\sum_{k=1}^{K} p_k\left(y_i|m_i\right)\log\left(p_k\left(y_i|m_i\right)\right). \tag{16}$$

Following these definition, the acquisition functions are defined as follows.

Similarity: samples are selected on the basis of their highest Tanimoto coefficient (computed with ECFPs; with 1,024 bits and a radius of 2) to any previously acquired hit compound.

Greedy: the best predicted samples are selected with

$$\psi = \arg\max_n \left(\mathbb{E}\left(y|m\right)\right). \tag{17}$$

Uncertainty: most uncertain samples are selected with

$$\psi = \arg\max_n (\mathbb{H}(y|m)). \tag{18}$$

Mutual Information (MI) [18]: selects samples with low mutual information with

$$\psi = \arg\max_n \left(\mathbb{H}(y|m) - \mathbb{E}_M \left[\mathbb{H}\left(y|m, \theta\right)\right]\right). \tag{19}$$

Upper Confidence Bound (UCB): selects samples with the highest upper confidence bound

$$\psi = \arg\max_n \left(\mathbb{E}\left(y|m\right) + \beta \cdot \mathbb{V}(y|m)\right), \tag{20}$$

where $\mathbb{V}$ denotes the standard deviation between the models.

C.3 Evaluation Metrics

BEDROC Boltzmann-enhanced discrimination of receiver operating characteristic (BEDROC)[54] is designed to assess early recognition performance, giving higher weights to active compounds that are ranked closer to the top. The formal definition is:

$$\text{BEDROC}_\alpha = \frac{\sum_{i=1}^{|\text{M}|} e^{-\alpha r_i/N}}{R_\alpha(\frac{1-e^{-\alpha}}{e^{\alpha/N}-1})} \times \frac{R_\alpha \sinh(\alpha/2)}{\cosh(\alpha/2) - \cosh(\alpha/2 - \alpha R_\alpha)} + \frac{1}{1 - e^{\alpha(1-R_\alpha)}}. \tag{21}$$

The commonly used variant is BEDROC_{85}, where the top 2% of ranked candidates contribute to 80% of the BEDROC score.

Enrichment Factor (EF_α) For evaluating early retrieval, we use EF_α defined as:

$$\text{EF}_\alpha = \frac{|M_\alpha|}{|M| \times \alpha}, \tag{22}$$

where M_α is the true active molecules in top $\alpha\%$, M is the whole library.

C.4 Implementation Details

Hyper-parameters We used the Adam optimizer with a learning rate of 3e-4. The training batch size was set to 64 and the inference batch size was set to 512 for quicker inference. The embedding dim of molecules is 130, and the hidden dim is 1024. The MLP and GIN employed in the molecular encoder have 3 layers. An MLP with 3 layers is also used in the action scoring layer. During the training step of active learning, the number of training epochs is set to 50. The ϵ and β in GRPO are set to $7\text{e}{-2}$ and $1\text{e}{-2}$, respectively.

Hardware The experiments were performed on a computational cluster equipped with dual Intel Xeon Gold 6248R CPUs (3.00 GHz, 48 cores) and an NVIDIA RTX 4090 GPU with 24 GB of memory, running Ubuntu 22.04.2 LTS as the operating system.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have made clear and precise claims in the abstract and introduction that accurately reflect the contributions and scope of the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the challenges in scaling GLARE to chemical spaces with 10^8 molecules due to time costs and note that its performance diminishes as the annotation budget increases, suggesting areas for future improvement.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provided details of experimental setup, including parameters and hardware, as shown in Appendix C.4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have attached our code to the supplementary materials and will open-source our data and code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We have included all the training and test details in the appendix, such as data splits, hyperparameters and metrics.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We have provided the detailed results with standard deviations in Appendix B.3.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have described in Appendix C.4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have provided the broader impacts of our work in the introduction section. Extensive experiments show that GLARE outperforms state-of-the-art active learning methods, boosting virtual screening efficiency, scalability, and foundation model performance.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: We strictly follow the licenses and terms of use for all assets utilized in our paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We introduce a reinforced active learning framework GLARE for virtual screening, and we have provided the details of our method in this paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: This paper does not incorporate large language models (LLMs) as significant, novel, or unconventional components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.