S-MolSearch: 3D Semi-supervised Contrastive Learning for Bioactive Molecule Search

Gengmo Zhou[1,2]*, **Zhen Wang**[2]*, **Feng Yu**[2], **Guolin Ke**[2] , **Zhewei Wei**[1]†, **Zhifeng Gao**[2]†
[1]Renmin University of China [2]DP Technology
{zgm2015, zhewei}@ruc.edu.cn, {wangz, yufeng, kegl, gaozf}@dp.tech

Abstract

Virtual Screening is an essential technique in the early phases of drug discovery, aimed at identifying promising drug candidates from vast molecular libraries. Recently, ligand-based virtual screening has garnered significant attention due to its efficacy in conducting extensive database screenings without relying on specific protein-binding site information. Obtaining binding affinity data for complexes is highly expensive, resulting in a limited amount of available data that covers a relatively small chemical space. Moreover, these datasets contain a significant amount of inconsistent noise. It is challenging to identify an inductive bias that consistently maintains the integrity of molecular activity during data augmentation. To tackle these challenges, we propose S-MolSearch, the first framework to our knowledge, that leverages molecular 3D information and affinity information in semi-supervised contrastive learning for ligand-based virtual screening. Drawing on the principles of inverse optimal transport, S-MolSearch efficiently processes both labeled and unlabeled data, training molecular structural encoders while generating soft labels for the unlabeled data. This design allows S-MolSearch to adaptively utilize unlabeled data within the learning process. Empirically, S-MolSearch demonstrates superior performance on widely-used benchmarks LIT-PCBA and DUD-E. It surpasses both structure-based and ligand-based virtual screening methods for AUROC, BEDROC and EF.

1 Introduction

Virtual Screening [1, 2, 3, 4] plays a crucial role in the early stages of drug discovery by identifying potential drug candidates from large molecular libraries. Structure-Based Virtual Screening (SBVS) [5, 6, 7, 8], a widely used virtual screening method, attempts to predict the best interaction between ligands against a protein target to form a protein-ligand complex. Recently, deep learning methods have also been explored. Methods trained on affinity labels [9, 10, 11] conduct virtual screening by modeling binding affinities and ranking based on prediction. Additionally, a method [12] uses similarities between embedding of pockets and molecules to search for active molecules. However, these SBVS methods cannot escape the dependency on the structure of protein targets, which is unavailable for challenging or novel targets, such as disordered proteins like c-Myc, limiting the applicability of SBVS. Besides, plenty of assays used in Virtual Screening [13] are cell-based rather than target-specific, introducing noise into the active molecules since their activity is not entirely dependent on interaction with the protein target.

To remedy this, Ligand-Based Virtual Screening (LBVS) [14, 15, 16, 17, 18] searches similar molecules via known bioactive molecules and does not depend on the structure of protein targets, which has attracted increasing attention. Computational LBVS methods [19, 20, 18] rigidly employ

[*] Equal contribution.
[†] Corresponding authors.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

structural similarity to search for molecules, often using atom-centered, smooth Gaussian overlays to assess molecular similarity. Searching for bioactive molecules needs to consider both structure and electronic similarity. Some methods [14] also require charge comparison, which is expensive and time-consuming. These methods struggle with inefficiency when handling large databases. Moreover, since only structural or charge information is considered while affinity is ignored, even if molecules with similar structures or charges are identified, they may still exhibit poor affinity in practical applications due to activity cliffs [21].

A natural question arises: how can we enhance LBVS by collecting molecule similarity data from large-scale unlabeled molecules? To search both similar and bioactive molecules, we can choose those bioactive molecules that bind to the same protein. Meanwhile, labeled molecule-protein binding data is limited due to the expensive affinity acquisition. Also various standards [22, 13] across different datasets, introducing substantial noise. It is difficult to cover the searching chemical space with affinity data alone. A feasible solution is to leverage similarity from finite affinity data to broader chemical space.

Inspired by the success of contrastive learning [23, 24, 25, 26], which can extract informative representations from large-scale data, we explore its feasibility for molecule data. However, those data augmentation techniques in vision or language cannot be directly applied to molecules due to their inherent 3D structure. This limitation stems from a lack of inductive bias to guarantee augmentation maintains the integrity of molecular activity.

To address the challenges, we propose S-MolSearch, a novel semi-supervised contrastive learning framework based on inverse optimal transport (IOT) [27]. S-MolSearch directly uses 3D molecular structures to capture structure similarity. It consists of two main components: one encoder f_θ for labeled dataset and another encoder g_ψ for the full dataset, which includes both labeled and unlabeled data. Both encoders are trained simultaneously to effectively leverage the two types of data. We organize a dataset of labeled molecule-protein binding data from ChEMBL [28]. Molecules are assigned to different targets based on binding affinity, with the active molecules corresponding to each target forming clusters. We sample from these clusters, treating molecules from the same cluster as positive samples and those from different clusters as negative samples to train encoder f_θ. This approach incorporates affinity information into training and avoids the limitation caused by relying solely on structural similarity. For the update of encoder g_ψ, we assume that the similarity measurements obtained by encoder f_θ can be generalized to unlabeled data. We input the same data from the full dataset into these two encoders separately and use optimal transport to obtain soft labels from f_θ. The encoder g_ψ is then trained using these soft labels. This integration enables S-MolSearch to effectively utilize unlabeled data, ensuring that the model learns from both affinity-labeled samples and broader structural similarities across the molecular dataset.

Empirically, S-MolSearch demonstrates superior performance on widely-used benchmarks LIT-PCBA [13] and DUD-E [22]. It consistently achieves state-of-the-art results, surpassing both SBVS methods and LBVS methods on AUROC, BEDROC and EF. Notably, S-MolSearch trained with a 0.9 similarity threshold significantly outperforms existing methods, achieving more than a 49% improvement on BEDROC and over a 30% improvement on EF compared to the best baseline on DUD-E. These results provide strong empirical support for the effectiveness of S-MolSearch, confirming its advanced capability for virtual screening.

Our main contributions are summarized as follows:

- We introduce S-MolSearch, which is the first time integrates both molecular 3D structures and affinity information into molecule search.

- Built upon inverse optimal transport, we develop a semi-supervised contrastive learning framework, which induces S-MolSearch. By combining limited labeled data with extensive unlabeled data, S-MolSearch can learn more informative representations and explore the chemical space more effectively.

- S-MolSearch is evaluated on widely-used benchmarks LIT-PCBA and DUD-E, surpassing both SBVS and LBVS methods to achieve state-of-the-art results.

2 Related work

2.1 Virtual Screening

Virtual screening can be broadly divided into two main categories: Structure-Based Virtual Screening (SBVS) and Ligand-Based Virtual Screening (LBVS). SBVS [5, 6, 7, 8] heavily relies on the structure of protein targets and typically employs molecular docking. Recently, many deep learning methods [29, 10, 11, 12] have also emerged. In contrast, LBVS [14, 15, 16, 17, 18] uses known active ligands as seeds to identify potential ligands. Molecule search is a major LBVS approach, typically divided into two categories: 2D similarity search and 3D similarity search. 2D molecule search methods [30, 31] use molecular fingerprints to search for similar molecules, while 3D molecular search methods [15, 16, 18] depend on shape overlap.

2.2 Optimal Transport and inverse optimal transport

Optimal Transport (OT) is a mathematical problem that aims to determine the most efficient way to redistribute one initial distribution (known as the source distribution) into another distribution (known as the target distribution) while minimizing a defined transportation cost. To handle computational complexities, OT often incorporates regularization [32], leading to a softened optimization problem. The regularized OT objective is a convex function, thereby ensuring a unique solution that can be efficiently solved using iterative methods [33, 34].

Inverse Optimal Transport (IOT) seeks to determine the cost matrix that explains an observed optimal transport. [27] introduces a method to infer unknown costs. [35] explores the mathematical theory behind IOT. In many IOT studies [36, 37], optimization is directly performed over the cost matrix, typically focusing on learnable distances between samples rather than on the sample features.

2.3 Semi-supervised learning

Semi-supervised learning [38, 39, 40, 41] is typically used in scenarios where labeled data is limited but unlabeled data is abundant. Pseudo-labeling [42] is a classic technique of semi-supervised learning. [43] introduce a self-ensembling method that generates pseudo-labels by forming a consensus prediction using the outputs of the network under different regularization and augmentation conditions. UPS [44] proposes an uncertainty-aware pseudo-label selection framework that improves pseudo-labeling accuracy by reducing the amount of noise in the training process. UST [45] employs a teacher-student training paradigm. The teacher model is responsible for selecting and generating pseudo-labels, while the student model learns from the labeled set augmented with these pseudo-labels. Recent work [46] utilizes additional unpaired images to construct caption-level and keyword-level pseudo-labels, enhancing training.

3 Method

3.1 Overview

Molecular similarity search is a type of ligand-based virtual screening whose purpose is to perform a rapid search and filtering of similar molecules in a molecular database based on a query molecule provided by the user. We model the task as a dense retrieval problem, using a well-trained encoder to extract embedding representations of molecules and rank them by their cosine similarity to a query molecule, thereby identifying the top k most similar candidates.

Building upon the principles of inverse optimal transport (IOT), we have developed the S-Molsearch method. As shown in Figure 1, S-Molsearch uses a molecular structure encoder f_θ for labeled dataset D_{sup} and another encoder g_ψ for the full dataset D_{full}, encompassing both labeled and unlabeled dataset. f_θ utilizes contrastive learning to learn from labeled dataset. g_ψ optimizes its parameters using the soft labels produced by f_θ, which have been processed through smooth optimal transport. The two encoders are initialized with a molecular pretraining backbone Uni-Mol [47]. Both encoders are trained simultaneously under the guidance of a unified loss function $\mathcal{L}_{total}$. The encoder g_ψ, trained on full dataset, is used for inference.

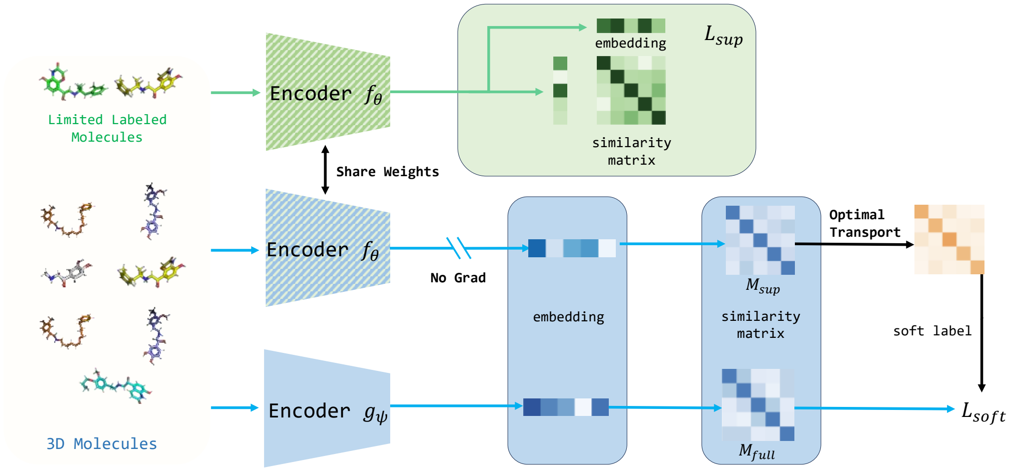


Figure 1: Overview of S-MolSearch Framework

In the following sections, we will provide more details about the core components and training strategies that underpin S-Molsearch. In section 3.2, we explain the pretraining backbone of the molecular structural encoder, Uni-Mol. In sections 3.3 and 3.4, we will sequentially examine the training strategies for both f_θ and g_ψ. In section 3.5, we will discuss the regularization techniques employed to enhance model generalizability and stability. In section 3.6, we will provide an analysis of S-Molsearch from the perspective of IOT, offering insights into its methodological strengths.

3.2 Pretraining Backbone of Molecular Encoder

To effectively encode the structural information of molecules, we choose Uni-Mol as the backbone of molecular encoder for S-Molsearch. Uni-Mol is a molecular pretraining model specifically designed to adeptly process molecular 3D conformation data. It has achieved state-of-the-art performance across a range of downstream tasks. The UniMol model utilizes a self-attention mechanism that incorporates distance bias to integrate information about atoms and their spatial relationships, thereby generating a structural representation of molecules. The molecular embedding is created by using the embedding of the CLS token, and the embedding vector is normalized using the Euclidean norm.

3.3 Training Strategy of Encoder on Labeled Dataset

In this part, we employ molecular-protein binding data sourced from the ChEMBL. Molecules are assigned to different targets based on binding affinity. The active molecules corresponding to each target form clusters. We sample from these clusters to obtain data for contrastive learning. Molecules from the same cluster are considered positive pairs, while molecules from different clusters are considered negative pairs. We employ InfoNCE loss for encoder f_θ on labeled dataset, as shown in Equation 1:

$$\mathcal{L}_{sup} = -\sum_{i=1}^{N} \log \frac{\exp(\text{sim}(x_i, y_i)/\tau)}{\sum_{j=1}^{N} \exp(\text{sim}(x_i, y_j)/\tau)} \tag{1}$$

where x_i and y_i are the embeddings of positive molecule pairs, while x_i and y_j form negative molecule embedding pairs within the batch when $j \neq i$. $\text{sim}(x, y)$ denotes the similarity score between embeddings, typically computed as the inner product of the normalized vectors xy^T. τ is the temperature parameter that scales the similarity scores. By utilizing the InfoNCE loss, f_θ pulls the embeddings of positive samples close while enforcing them away from the negative samples in the embedding space.

3.4 Training Strategy of Encoder on Full Dataset

To harness large-scale unsupervised data effectively, S-MolSearch utilizes the f_θ to guide the learning of another encoder on full data g_ψ. Specifically, we employ f_θ to preprocess the data for the g_ψ, ensuring that f_θ remains detached from the backpropagation process at this stage. The embeddings obtained from f_θ are subsequently utilized for computing a similarity matrix $M_{sup} \in R^{N \times N}$, where $M_{sup}(i,j) = x_{sup,i}x^T_{sup,j}$ The task of generating soft labels for g_ψ based on this similarity matrix is presented as a smooth and sparse optimal transport (OT) problem:

$$\begin{aligned} &\min_{\Gamma \in U(p,q)} \langle \Gamma, C \rangle + \frac{\lambda}{2}||\Gamma||^2 \\ &\text{subject to} \quad U(p,q) = \{\Gamma \in R_+^{N\times N} | \Gamma \mathbf{1}_N = p, \Gamma^\top \mathbf{1}_N = q\} \end{aligned} \tag{2}$$

Where Γ denotes the transportation plan matrix between the embeddings, $C \in R_+^{N \times N}$ is the cost matrix derived from the cosine similarities between embeddings: $C_{i,j} = c - x_{sup,i}x^T_{sup,j}$, $\langle \Gamma, C \rangle$ denotes Frobenius inner product of Γ and C, and p and q are the source and sink distributions, respectively. In this context, $c = 1$, $p = 1_N$ and $q = 1_N$. 1_N is an N-dimensional vector of all ones. By introducing the OT formulation, we guarantee that signals from the supervised model f_θ are more effectively transferred to the unsupervised model g_ψ, while handling high label uncertainty in the supervised model f_θ with appropriate regularization. The OT could be effectively addressed by the POT[48]. We define $\Gamma_{i,j}$ as pseudo-labels for the similarity matrix M_{full} of g_ψ, where M_{full} is created by embedding of g_ψ $M_{full}(i,j) = x_{full,i}x^T_{full,j}$.

The cross-entropy loss H is employed to optimize g_ψ, using the pseudo-labels provided:

$$\mathcal{L}_{soft} = H(\Gamma, M_{full}) \tag{3}$$

3.5 Regularization techniques

In order to promote uniformity of embedding space, we apply KoLeo regularizer[49, 50] to the embeddings of the semi-supervised encoder. KoLeo regularizer is defined as:

$$\mathcal{L}_{reg} = -\frac{1}{n}\sum_{i=1}^{n} \log(\rho_{n,i}) \tag{4}$$

Here, $\rho_{n,i}$ represents the minimum distance between the i-th sample and all other samples, which serves as a proxy for local density. This loss function has a geometric interpretation that effectively pushes closer points apart, ensuring diminishing returns as distances increase, thereby encouraging a uniformly dispersed embedding space.

3.6 Framework for S-Molsearch Induced by Inverse Optimal Transport

By integrating the losses and regularization terms from sections 3.3, 3.4, and 3.5, we have derived the overall loss function $\mathcal{L}_{total}$ for the S-MolSearch model:

$$\mathcal{L}_{total} = \mathcal{L}_{sup} + \mathcal{L}_{soft} + \mu\mathcal{L}_{reg} \tag{5}$$

where μ is 0.1 in our setting. Building on the relationship between contrastive learning and IOT established in [51], we extend this relationship to a semi-supervised contrastive learning in proposition 1.

Proposition 1 *Given encoder f_θ for labeled dataset X_{sup} and g_ψ for full dataset X_{full}, x_{sup} represents the embeddings of labeled data from f_θ , while x_{full} represents the embeddings of the full dataset from g_ψ. Semi-supervised contrastive learning is then formulated using IOT as follows:*

$$\begin{aligned} &\min_{\theta,\psi} \Big(KL(\Gamma^g || \Gamma^\theta) + KL(\hat{\Gamma}^\theta || \Gamma^\psi) + \mu Reg_1(\Gamma^\psi) + \nu Reg_2(\Gamma^\theta)) \Big) \\ &\textit{subject to} \quad \Gamma^\theta = \arg \min_{\Gamma \in U(a),\, a = \frac{1}{N}} \big(\langle C^\theta, \Gamma \rangle - \tau H(\Gamma) \big)\,, \\ &\qquad\qquad\quad \Gamma^\psi = \arg \min_{\Gamma \in U(a),\, a = \frac{1}{N}} \big(\langle C^\psi, \Gamma \rangle - \tau H(\Gamma) \big)\,, \\ &\qquad\qquad\quad \hat{\Gamma}^\theta = \mathcal{T}(f_\theta^{fixed}, g_\psi^{fixed}, X_{label}, X_{full}) \end{aligned} \tag{6}$$

where $KL(X||Y) = \sum_{ij} x_{ij} log \frac{x_{ij}}{y_{ij}} - x_{ij} + y_{ij}$ represents the Kullback-Leibler divergence, and $H(\Gamma) = -\sum_{i,j} \Gamma_{ij}(\log(\Gamma_{ij}) - 1)$ represents entropic regularization. $\Gamma^\theta, \Gamma^\psi, \Gamma^g \in R_+^{N\times N}$, $\Gamma_{ij}^g = \frac{\delta_{ij}}{N}$ represents the ground truth based on labeled data, δ_{ij} denotes the Kronecker delta function. $C^\theta, C^\psi \in R_+^{N\times N}$ are cost matrix of f_θ, g_ψ and $C^\theta(i,j) = c - x_{sup,i}x_{sup,j}^T$, $C^\psi(i,j) = c - x_{full,i}x_{full,j}^T$. $\mathcal{T}$ generally refers to a technique for transferring supervised information to unsupervised data. Reg_1, Reg_2 denotes regularization term.

The proof is provided in the Appendix C.1. In proposition 1, we model the contrastive learning problem on the labeled dataset and full dataset as two optimal transport problems. Additionally, we use $\mathcal{T}$ to transfer knowledge from the labeled dataset to the unlabeled data, with the method of transfer depending on prior assumptions about the dataset and certain bias structures. For instance, If we initially optimize f_θ on a large-scale labeled dataset to obtain f_θ^*, then generate $\hat{\Gamma}^\theta$ as $\hat{\Gamma}^\theta(i,j) = e_{full,i}e_{full,j}^T$, where $e_{full,i} = f_\theta^*(x_i), x_i \in X_{full}$, we can develop a model that leverages knowledge distillation for contrastive learning. In the context of molecular search tasks, we employ smooth optimal transport for the modeling of $\hat{\Gamma}^\theta$, leading to the development of S-MolSearch as follows:

Proposition 2 *Assuming the conditions outlined in Proposition 1 are satisfied, the optimal parameters θ^* and ψ^* of S-MolSearch can be regarded as the solution to the following IOT problem:*

$$\begin{aligned}
&\min_{\theta,\psi} \left(KL(\Gamma^g \| \Gamma^\theta) + KL(\hat{\Gamma}^\theta \| \Gamma^\psi) + \mu Reg_1(\Gamma^\psi) \right) \\
&\textit{subject to} \quad \Gamma^\theta = \arg \min_{\Gamma \in U(a),\, a=\frac{\mathbf{1}}{N}} \left(\langle C^\theta, \Gamma \rangle - \tau H(\Gamma) \right), \\
&\qquad\qquad\quad \Gamma^\psi = \arg \min_{\Gamma \in U(a),\, a=\frac{\mathbf{1}}{N}} \left(\langle C^\psi, \Gamma \rangle - \tau H(\Gamma) \right), \\
&\qquad\qquad\quad \hat{\Gamma}^\theta = \arg \min_{\Gamma \in U(a,b),\, a=\mathbf{1}_N,\, b=\mathbf{1}_N} \left(\langle C^{\theta^{\text{fixed}}}, \Gamma \rangle + \frac{\lambda}{2} \|\Gamma\|^2 \right)
\end{aligned} \tag{7}$$

where $C^{\theta^{fixed}} \in R_+^{N\times N}$ and $C^{\theta^{fixed}}(i,j) = c - x_{full,i}^{fixed}(x_{full,j}^{fixed})^T$. The x_{full}^{fixed} represents the embeddings of the same data in X_{full} obtained from the supervised encoder f_θ, where f_θ is detached.

The proof is located in the Appendix C.2. We compute the KL divergence to guide the optimization of f_θ and g_ψ, where the regularization term simplification only affects the full data. Moreover, we set the marginal values of $U(a,b)$ to an all-ones vector. In this way, we find that the knowledge in f_θ transfers effectively to g_ψ, thereby achieving excellent performance on molecule search task.

4 Experiments

4.1 Training Data

The labeled data comes from ChEMBL [28], an open-access database containing extensive information on bioactive compounds with drug-like properties. We prepare nearly 600,000 protein-molecule pairs, encompassing about 4,200 protein targets and 300,000 molecules. The details of data curation can be found in appendix A. To prevent information leakage, the data is filtered based on protein sequence similarity. Specifically, the amino acid sequences of all protein targets in the benchmarks DUD-E and LIT-PCBA are extracted. Then, we use MMseqs [52] tool with similarity thresholds of 0.4 and 0.9 to filter out proteins in ChEMBL. Using a 0.9 threshold helps filter out identical and highly similar targets, while the stricter 0.4 threshold filters out nearly all similar targets. After filtering, 3,369 proteins and 327,917 protein-ligand pairs remain for the 0.4 threshold, while 4,102 proteins and 529,856 pairs remain for the 0.9 threshold. We sample 1 million pairs from the filtered data as labeled data respectively.

The unlabeled data, consistent with what is used by Uni-Mol, comes from a series of public databases, totaling about 19 million entries. Additionally, we incorporate the small molecule data from ChEMBL into this collection, thereby obtaining the full dataset.

4.2 Benchmarks

We choose the widely used virtual screening benchmarks DUD-E [22] and LIT-PCBA [13] to evaluate the performance of S-MolSearch. DUD-E is designed to help benchmark virtual screening programs by providing challenging decoys. It includes 102 protein targets along with 22,886 active ligands, each accompanied by 50 decoys with similar physico-chemical properties. LIT-PCBA is designed for virtual screening and machine learning, aiming to address the chemical biases present in other benchmarks such as DUD-E. It consists of 15 targets, with 7,844 confirmed active compounds and 407,381 inactive compounds.

4.3 Baselines

We choose a range of LBVS and SBVS methods as comparative baselines for a thorough evaluation. ROCS [14], Phase Shape [15], LIGSIFT [18], and SHAFTS [19, 20] are LBVS methods that evaluate similarity by calculating the overlap of molecular 3D shapes. Other methods are SBVS methods. Among them, DeepDTA [9], OnionNet [10], Pafnucy [11], BigBind [53], and Planet [54] are trained on binding affinity labels. Glide [7], Vina [8], and Surflex [55] are molecular docking software. Gnina [56] is a deep learning based molecular docking method. DrugClip [12] utilizes the similarity between targets and molecules to find active compounds.

4.4 Results

4.4.1 Main Results

Table 1: Performance on DUD-E in zero-shot setting. The best results are **bolded** and the second-best results are underlined.

Method	AUROC (%)	BEDROC (%)	EF 0.5%	EF 1%	EF 5%
ROCS	75.20	-	-	23.79	6.89
Phase Shape	76.70	-	-	30.33	9.01
LIGSIFT	78.40	-	-	25.89	8.01
SHAFTS	78.20	-	-	32.49	9.67
Glide-SP	76.70	40.70	19.39	16.18	7.23
Vina	71.60	-	9.13	7.32	4.44
Pafnucy	63.11	16.50	4.24	3.86	3.76
OnionNet	59.71	8.62	2.84	8.83	5.40
Planet	71.60	-	10.23	8.83	5.40
DrugCLIP	80.93	50.52	38.07	31.89	10.66
S-MolSearch$_{0.4}$	84.61	54.22	40.85	34.60	11.44
S-MolSearch$_{0.9}$	**92.56**	**75.37**	**51.50**	**47.94**	**15.82**

Table 2: Performance on LIT-PCBA in zero-shot setting.

Method	AUROC (%)	BEDROC (%)	EF 0.5%	EF 1%	EF 5%
ROCS	52.41	-	-	2.48	-
Phase Shape	52.24	-	-	2.98	-
LIGSIFT	54.94	-	-	2.39	-
SHAFTS	54.53	-	-	2.79	-
Surflex	51.47	-	-	2.50	-
Glide-SP	53.15	4.00	3.17	3.41	2.01
Planet	57.31	-	4.64	3.87	2.43
Gnina	60.93	5.40	-	4.63	-
DeepDTA	56.27	2.53	-	1.47	-
BigBind	60.80	-	-	3.82	-
DrugCLIP	57.17	6.23	8.56	5.51	2.27
S-MolSearch$_{0.4}$	57.34	7.58	10.93	6.28	2.47
S-MolSearch$_{0.9}$	**61.78**	**8.48**	**11.97**	**7.36**	**3.21**

Tables 1 and 2 respectively present the performance of S-MolSearch on DUD-E and LIT-PCBA compared with other competitive baselines, where the best results are highlighted in bold and the second-best results are underlined. The methods in the upper part of the two tables are ligand-based virtual screening methods, and their results come from [57]. The methods in the middle part of the two tables are structure-based virtual screening methods, and their results come from DrugClip. We also present the results of S-MolSearch trained on data filtered with 0.4 and 0.9 similarity thresholds. Following previous work, we choose AUROC, BEDROC, Enrichment factor (EF) as performance metrics to evaluate both general accuracy and screening capacity, with higher values indicating better performance. Their definitions are in appendix B. The zero-shot setting means inferring directly without using any data from the benchmarks for training, which more close to real virtual screening scenarios.

Table 1 shows that S-MolSearch achieves the best on all metrics. S-MolSearch trained with a strict 0.4 similarity threshold avoid overfitting similar targets and surpasses all ligand-based and structure-based virtual screening baselines. S-MolSearch trained with a 0.9 similarity threshold shows substantial improvements over existing methods, with over a 49% increase in BEDROC and more than a 30% boost in EF compared to the best baseline. We find that S-MolSearch, as a ligand-based virtual screening method, demonstrates strong performance without requiring specific protein structures.

S-MolSearch also achieves SOTA on LIT-PCBA as shown in Table 2. While its AUROC performance is not the best at the 0.4 similarity threshold, S-MolSearch perform better in BEDROC and EF, indicating its strength in screening scenarios. We notice that the metrics for all methods decline on LIT-PCBA compared to DUD-E. Unlike DUD-E, which uses putative decoys, LIT-PCBA is based on experimental results. Since many of its assays are cell-based rather than target-specific, there is noise in the active molecules, which we consider may lead to the decline. Meanwhile, S-MolSearch demonstrates its advantage over structure-based virtual screening by not requiring specific target information, but instead performing searches based on active molecules.

4.4.2 Ablation Study

Table 3: Ablation studies performance on DUD-E and LIT-PCBA.

Soft label	Regularizer	Pretrain	DUD-E			LIT-PCBA		
			EF 0.5%	EF 1%	EF 5%	EF 0.5%	EF 1%	EF 5%
✗	✓	✓	37.35	30.73	10.43	10.59	6.19	**2.72**
✓	✗	✓	39.64	33.32	**11.47**	9.01	5.24	2.38
✓	✓	✗	38.09	31.86	10.88	8.26	5.24	2.30
✓	✓	✓	**40.85**	**34.60**	11.44	**10.93**	**6.28**	2.47

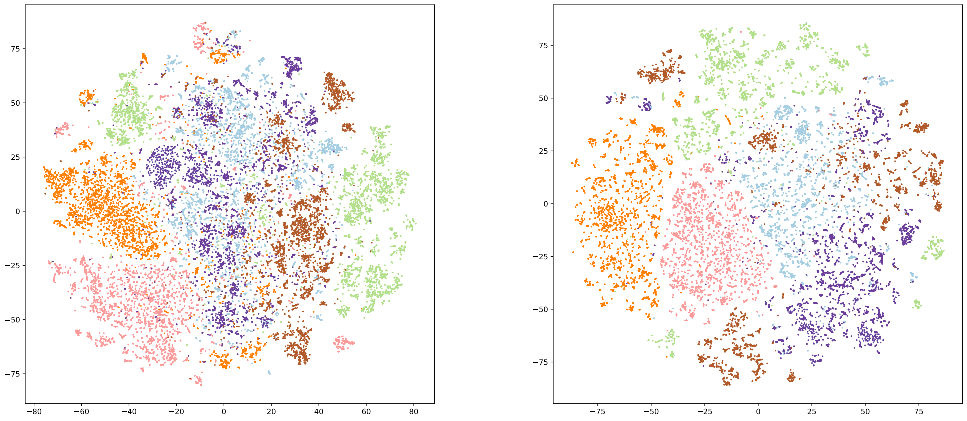
We conduct extensive ablation studies to explore how S-MolSearch works. These results are derived from S-MolSearch trained with a 0.4 similarity threshold. First, we performed ablation studies on several key techniques of S-MolSearch. The results are summarized in Table 3, where the best results are bolded and the second-best results are underlined. 'Soft label' refers to training the encoder g_ψ using soft labels obtained from inverse optimal transport. Without this, we directly use the similarity matrix as the hard label for training. 'Regularizer' indicates the use of KoLeo regularizer. 'Pretrain' refers to starting the training of S-MolSearch from a pretrained checkpoint of Uni-Mol. Otherwise, it starts from random initialization. The results show that each component contributes to the final results. Although S-MolSearch may not be the best in some individual metrics, the absence of these components leads to poor performance on at least one benchmark. For example, not using Soft label significantly degrades performance on DUD-E. S-MolSearch performs consistently on both DUD-E and LIT-PCBA, with nearly all metrics being the best.

To visually illustrate the difference between the embeddings learned by S-MolSearch and those from Uni-Mol, we visualize their embeddings, as shown in Figure 2. The molecules are from ChEMBL, with different colors indicating different protein targets. Comparing the two, the classification boundaries in Figure 2b from S-MolSearch are clearer, and the intra-class molecular distances are more appropriate. Some clusters split into several subclusters, possibly reflecting the hierarchical structure within the molecules.

Table 4: Performance under different learning paradigms on DUD-E and LIT-PCBA.

	DUD-E			LIT-PCBA		
	EF 0.5%	EF 1%	EF 5%	EF 0.5%	EF 1%	EF 5%
Self-supervised	27.33	20.13	6.81	4.78	3.26	1.97
Supervised	34.61	28.51	9.83	7.55	4.73	1.98
Finetuning	35.11	29.20	9.96	7.01	5.71	2.44
S-MolSearch	**40.85**	**34.60**	**11.44**	**10.93**	**6.28**	**2.47**

In addition, we conduct ablation studies on the semi-supervised learning paradigm of S-MolSearch. The results are summarized in Table 4. For 'Self-supervised', we train the model using a self-supervised learning paradigm. Specifically, we cluster the unlabeled molecule data based on their scaffolds. Molecules from the same cluster are considered as positive pairs, while those from different clusters are considered as negative pairs, and contrastive learning is performed using the InfoNCE loss. For 'Supervised', we use only the ChEMBL data for supervised contrastive learning, where molecules binding to the same target are treated as positive pairs and those binding to different targets as negative pairs. For 'Finetuning', we first train the model under the self-supervised paradigm described above, then, starting from the self-supervised checkpoint, perform the supervised learning described above on the ChEMBL data. The results show that S-MolSearch consistently performs well on both benchmarks, achieving the best results in almost all metrics. Notably, compared to finetuning, S-MolSearch demonstrates a superior ability to integrate information from both unlabeled and labeled data in ligand-based virtual screening scenarios.



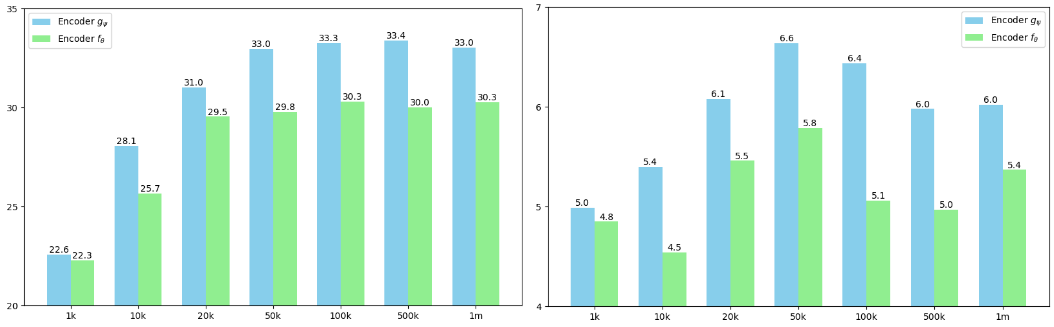
(a) Representations from pretrained checkpoint

(b) Representations learned by S-MolSearch

Figure 2: t-SNE visualization of molecular representations learned by S-MolSearch versus pretrained checkpoint. Different colors represent different protein targets' active molecules.

4.4.3 Impact of Labeled Data Scale

In the molecular field, obtaining or creating labeled data can be expensive. We also analyze how the scale of labeled data affects the results. These results are derived from S-MolSearch trained with a 0.4 similarity threshold. As shown in Figure 3, experiments are conducted with varying amounts of labeled data, while keeping the unlabeled data fixed at 1 million. The performance of encoder g_ψ improves as the amount of labeled data increases, especially when the absolute number of labeled data is limited. Additionally, the results of encoder g_ψ are consistently higher than encoder f_θ trained using only labeled data. The best results are achieved with 50k and 100k labeled data, corresponding to labeled-to-unlabeled data ratios of 1:20 and 1:10, respectively. Beyond these amounts, increasing labeled data results in stable or slightly declining performance, suggesting that further improvements



(a) EF 1% on DUD-E

(b) EF 1% on LIT-PCBA

Figure 3: Performance on DUD-E and LIT-PCBA with varying numbers of labeled data, while keeping unlabeled data fixed at 1m. The blue bars represent the results of encoder g_ψ, while the green bars represent the results of encoder f_θ.

may necessitate an increase in unlabeled data or a reevaluation of hyperparameters. This trend helps us to find an optimal balance between labeled and unlabeled data to maximize efficiency and performance in S-MolSearch.

5 Conclusion

This study introduces S-MolSearch, a novel semi-supervised contrastive learning framework that significantly enhances the generalizability of machine learning models in virtual screening. Built on inverse optimal transport, S-MolSearch skillfully integrates limited labeled data with a vast reservoir of unlabeled data and excels at identifying potential drug candidates from extensive molecular libraries, substantially improving the accuracy and efficiency of molecule searches. This advancement addresses current challenges in virtual screening by facilitating efficient filtering of large datasets, highlighting the framework's capability in scenarios where data annotation is costly.

Currently, S-MolSearch predominantly focuses on the molecular affinity data, omitting broader biochemical interactions, which suggests a potential area for improvement. Future work could integrate more extensive unsupervised datasets to further refine the framework's effectiveness and explore additional applications in various bioinformatics fields.

Acknowledgments and Disclosure of Funding

This research was supported in part by National Natural Science Foundation of China (No. U2241212, No. 61932001), by National Science and Technology Major Project (2022ZD0114802), by Beijing Natural Science Foundation (No. 4222028), by Beijing Outstanding Young Scientist Program No.BJJWZYJH012019100020098. This research was also an outcome of “AI-Aided Drug Design Based on Universal Representation of Multi-Modal Graph Structures" (RUC24QSDL014), funded by the "Qiushi Academic - Dongliang" Talent Cultivation Project at Renmin University of China in 2024. We also wish to acknowledge the support provided by the fund for building world-class universities (disciplines) of Renmin University of China, by Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education, Intelligent Social Governance Interdisciplinary Platform, Major Innovation & Planning Interdisciplinary Platform for the “Double-First Class” Initiative, Public Policy and Decision-making Research Lab, and Public Computing Cloud, Renmin University of China. The work was partially done at Gaoling School of Artificial Intelligence, Beijing Key Laboratory of Big Data Management and Analysis Methods, MOE Key Lab of Data Engineering and Knowledge Engineering, and Pazhou Laboratory (Huangpu), Guangzhou, Guangdong 510555, China.

References

[1] Gerhard Klebe. “Virtual ligand screening: strategies, perspectives and limitations”. In: *Drug discovery today* 11.13-14 (2006), pp. 580–594.

[2] Hongbin Huang et al. “Reverse screening methods to search for the protein targets of chemopreventive compounds”. In: *Frontiers in chemistry* 6 (2018), p. 138.

[3] Dominique Sydow et al. “Advances and challenges in computational target prediction”. In: *Journal of chemical information and modeling* 59.5 (2019), pp. 1728–1742.

[4] Berin Karaman and Wolfgang Sippl. “Computational drug repurposing: current trends”. In: *Current medicinal chemistry* 26.28 (2019), pp. 5389–5409.

[5] Xuan-Yu Meng et al. “Molecular docking: a powerful approach for structure-based drug discovery”. In: *Current computer-aided drug design* 7.2 (2011), pp. 146–157.

[6] Wenchao Lu et al. “Computer-aided drug design in epigenetics”. In: *Frontiers in chemistry* 6 (2018), p. 57.

[7] Thomas A Halgren et al. “Glide: a new approach for rapid, accurate docking and scoring. 2. Enrichment factors in database screening”. In: *Journal of medicinal chemistry* 47.7 (2004), pp. 1750–1759.

[8] Oleg Trott and Arthur J Olson. “AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading”. In: *Journal of computational chemistry* 31.2 (2010), pp. 455–461.

[9] Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. “DeepDTA: deep drug–target binding affinity prediction”. In: *Bioinformatics* 34.17 (2018), pp. i821–i829.

[10] Liangzhen Zheng, Jingrong Fan, and Yuguang Mu. “Onionnet: a multiple-layer intermolecular-contact-based convolutional neural network for protein–ligand binding affinity prediction”. In: *ACS omega* 4.14 (2019), pp. 15956–15965.

[11] Marta M Stepniewska-Dziubinska, Piotr Zielenkiewicz, and Pawel Siedlecki. “Development and evaluation of a deep learning model for protein–ligand binding affinity prediction”. In: *Bioinformatics* 34.21 (2018), pp. 3666–3674.

[12] Bowen Gao et al. “DrugCLIP: Contrasive Protein-Molecule Representation Learning for Virtual Screening”. In: *Advances in Neural Information Processing Systems* 36 (2024).

[13] Viet-Khoa Tran-Nguyen, Célien Jacquemard, and Didier Rognan. “LIT-PCBA: an unbiased data set for machine learning and virtual screening”. In: *Journal of chemical information and modeling* 60.9 (2020), pp. 4263–4273.

[14] Paul CD Hawkins, A Geoffrey Skillman, and Anthony Nicholls. “Comparison of shape-matching and docking as virtual screening tools”. In: *Journal of medicinal chemistry* 50.1 (2007), pp. 74–82.

[15] G Madhavi Sastry, Steven L Dixon, and Woody Sherman. “Rapid shape-based ligand alignment and virtual screening method based on atom/feature-pair similarities and volume overlap scoring”. In: *Journal of chemical information and modeling* 51.10 (2011), pp. 2455–2466.

[16] Jonatan Taminau, Gert Thijs, and Hans De Winter. “Pharao: pharmacophore alignment and optimization”. In: *Journal of Molecular Graphics and Modelling* 27.2 (2008), pp. 161–169.

[17] Xin Yan et al. “Enhancing molecular shape comparison by weighted Gaussian functions”. In: *Journal of chemical information and modeling* 53.8 (2013), pp. 1967–1978.

[18] Ambrish Roy and Jeffrey Skolnick. “LIGSIFT: an open-source tool for ligand structural alignment and virtual screening”. In: *Bioinformatics* 31.4 (2014), pp. 539–544.

[19] Xiaofeng Liu, Hualiang Jiang, and Honglin Li. “SHAFTS: a hybrid approach for 3D molecular similarity calculation. 1. Method and assessment of virtual screening”. In: *Journal of chemical information and modeling* 51.9 (2011), pp. 2372–2385.

[20] Weiqiang Lu et al. “SHAFTS: a hybrid approach for 3D molecular similarity calculation. 2. Prospective case study in the discovery of diverse p90 ribosomal S6 protein kinase 2 inhibitors to suppress cell migration”. In: *Journal of medicinal chemistry* 54.10 (2011), pp. 3564–3574.

[21] Dilyana Dimova and Jürgen Bajorath. “Advances in activity cliff research”. In: *Molecular informatics* 35.5 (2016), pp. 181–191.

[22] Michael M Mysinger et al. “Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking”. In: *Journal of medicinal chemistry* 55.14 (2012), pp. 6582–6594.

[23] Ting Chen et al. “A simple framework for contrastive learning of visual representations”. In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.

[24] Kaiming He et al. “Momentum contrast for unsupervised visual representation learning”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 9729–9738.

[25] Alec Radford et al. “Learning transferable visual models from natural language supervision”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8748–8763.

[26] Jean-Bastien Grill et al. “Bootstrap your own latent-a new approach to self-supervised learning”. In: *Advances in neural information processing systems* 33 (2020), pp. 21271–21284.

[27] Andrew M Stuart and Marie-Therese Wolfram. “Inverse optimal transport”. In: *SIAM Journal on Applied Mathematics* 80.1 (2020), pp. 599–619.

[28] Anna Gaulton et al. “ChEMBL: a large-scale bioactivity database for drug discovery”. In: *Nucleic acids research* 40.D1 (2012), pp. D1100–D1107.

[29] Matthew Ragoza et al. “Protein–ligand scoring with convolutional neural networks”. In: *Journal of chemical information and modeling* 57.4 (2017), pp. 942–957.

[30] Peter Willett. “Similarity-based virtual screening using 2D fingerprints”. In: *Drug discovery today* 11.23-24 (2006), pp. 1046–1053.

[31] Xin Yan et al. “GSA: a GPU-accelerated structure similarity algorithm and its application in progressive virtual screening”. In: *Molecular diversity* 16 (2012), pp. 759–769.

[32] Alan Geoffrey Wilson. “The use of entropy maximising models, in the theory of trip distribution, mode split and route split”. In: *Journal of transport economics and policy* (1969), pp. 108–126.

[33] Richard Sinkhorn. “Diagonal equivalence to matrices with prescribed row and column sums”. In: *The American Mathematical Monthly* 74.4 (1967), pp. 402–405.

[34] Mathieu Blondel, Vivien Seguy, and Antoine Rolet. *Smooth and Sparse Optimal Transport*. 2018. arXiv: `1710.06276 [stat.ML]`.

[35] Wei-Ting Chiu, Pei Wang, and Patrick Shafto. “Discrete probabilistic inverse optimal transport”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 3925–3946.

[36] Arnaud Dupuy, Alfred Galichon, and Yifei Sun. “Estimating matching affinity matrix under low-rank constraints”. In: *arXiv preprint arXiv:1612.09585* (2016).

[37] Ruilin Li et al. “Learning to match via inverse optimal transport”. In: *Journal of machine learning research* 20.80 (2019), pp. 1–37.

[38] Semi-Supervised Learning. “Semi-supervised learning”. In: *CSZ2006. html* 5 (2006).

[39] David Berthelot et al. “Mixmatch: A holistic approach to semi-supervised learning”. In: *Advances in neural information processing systems* 32 (2019).

[40] Qizhe Xie et al. “Self-training with noisy student improves imagenet classification”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 10687–10698.

[41] Mahmoud Assran et al. “Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 8443–8452.

[42] Dong-Hyun Lee et al. “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks”. In: *Workshop on challenges in representation learning, ICML*. Vol. 3. 2. Atlanta. 2013, p. 896.

[43] Samuli Laine and Timo Aila. “Temporal ensembling for semi-supervised learning”. In: *arXiv preprint arXiv:1610.02242* (2016).

[44] Mamshad Nayeem Rizve et al. “In Defense of Pseudo-Labeling: An Uncertainty-Aware Pseudo-label Selection Framework for Semi-Supervised Learning”. In: *International Conference on Learning Representations*. 2020.

[45] Subhabrata Mukherjee and Ahmed Awadallah. “Uncertainty-aware self-training for few-shot text classification”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 21199–21212.

[46] Sangwoo Mo et al. “S-clip: Semi-supervised vision-language learning using few specialist captions”. In: *Advances in Neural Information Processing Systems* 36 (2024).

[47] Gengmo Zhou et al. “Uni-Mol: A Universal 3D Molecular Representation Learning Framework”. In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=6K2RM6wVqKu.

[48] Rémi Flamary et al. “POT: Python Optimal Transport”. In: *Journal of Machine Learning Research* 22.78 (2021), pp. 1–8. URL: http://jmlr.org/papers/v22/20-451.html.

[49] Alexandre Sablayrolles et al. “Spreading vectors for similarity search”. In: *arXiv preprint arXiv:1806.03198* (2018).

[50] Maxime Oquab et al. “Dinov2: Learning robust visual features without supervision”. In: *arXiv preprint arXiv:2304.07193* (2023).

[51] Liangliang Shi et al. “Understanding and generalizing contrastive learning from the inverse optimal transport perspective”. In: *International conference on machine learning*. PMLR. 2023, pp. 31408–31421.

[52] Maria Hauser, Martin Steinegger, and Johannes Söding. “MMseqs software suite for fast and deep clustering and searching of large protein sequence sets”. In: *Bioinformatics* 32.9 (2016), pp. 1323–1330.

[53] Michael Brocidiacono et al. “BigBind: learning from nonstructural data for structure-based virtual screening”. In: *Journal of Chemical Information and Modeling* 64.7 (2023), pp. 2488–2495.

[54] Xiangying Zhang et al. “Planet: a multi-objective graph neural network model for protein–ligand binding affinity prediction”. In: *Journal of Chemical Information and Modeling* (2023).

[55] Russell Spitzer and Ajay N Jain. “Surflex-Dock: Docking benchmarks and real-world application”. In: *Journal of computer-aided molecular design* 26 (2012), pp. 687–699.

[56] Andrew T McNutt et al. “GNINA 1.0: molecular docking with deep learning”. In: *Journal of cheminformatics* 13.1 (2021), p. 43.

[57] Zhenla Jiang et al. “A comprehensive comparative assessment of 3D molecular similarity tools in ligand-based virtual screening”. In: *Briefings in bioinformatics* 22.6 (2021), bbab231.

A Data Curation Details

The data we used are manually extracted from regularly published primary literature, and then further curated and standardized. The whole ChEMBL database of version 33 is downloaded and cleaned via the following steps. The data is selected with assay type of 'B', target type of 'SINGLE PROTEIN', molecule type of 'Small molecule', standard type of Ki, Kd, IC50, EC50 in nM, and standard relation of '=' and '<'. Moreover, data with abnormal concentration value is further deleted, such as negative and unreasonable large ones.

B Metrics

The BEDROC metric is designed to assess early retrieval performance, giving higher weights to active compounds that are ranked closer to the top. It is defined as:

$$\text{BEDROC}_{\alpha} = \frac{\sum_{i=1}^{n} e^{-\alpha r_i/N}}{R_{\alpha}\left(\frac{1-e^{-\alpha}}{e^{\alpha/N}-1}\right)} \times \frac{R_{\alpha}\sinh(\alpha/2)}{\cosh(\alpha/2)-\cosh(\alpha/2-\alpha R_{\alpha})} + \frac{1}{1-e^{\alpha(1-R_{\alpha})}}$$

where n is the total number of active compounds, N is the total number of molecules, r_i is the rank of the i-th active compound. Following previous work, we set $\alpha = 85$ to prioritize early retrieval.

We also use the Enrichment Factor (EF) to evaluate the effectiveness of virtual screening methods. The calculation formula for EF is as follows:

$$\text{EF} = \frac{n_a/N_{x\%}}{n/N}$$

where n represents the total number of active compounds in the database, N represents the total number of molecules, $N_{x\%}$ represents the top $x\%$of all molecules, and n_a represents the number of active compounds within the top $x\%$ of molecules.

C Proof for S-Molsearch Induced by Inverse Optimal Transport

First, we introduce a lemma to establish the relationship between IOT and contrastive learning from [51].

Lemma 3 *The optimization problem 3 and 9 are equivalent:*

$$\begin{aligned} &\min_{\theta} KL(\hat{P}||P^{\theta}) \\ &\textit{subject to} \quad P^{\theta} = \arg\min_{P\in U(a)} \langle C^{\theta}, P\rangle - \tau H(P) \end{aligned} \tag{8}$$

where $C^{\theta} \in R^{M\times N}, C^{\theta}(i,j) = c - s_{ij}(\theta)$ and $\hat{P}(i,j) = \frac{\delta_{ij}}{n}$, δ denotes the Kronecker delta function.

$$\min_{\theta} - \sum_{i=1}^{n} log(\frac{exp(s_{ii}(\theta)/\tau)}{\sum_{j\neq i} exp(s_{ij}(\theta)/\tau)}) \tag{9}$$

In addition, P^{θ} has the form as follows:

$$P^{\theta} = \frac{exp(s_{ii}(\theta)/\tau)}{\sum_{j\neq i} Nexp(s_{ij}(\theta)/\tau)} \tag{10}$$

C.1 Proof for Proposition 1

In the optimization problem 6, since the setting of $\mathcal{T}$ does not involve parameter optimization of θ or ψ, it is evident that the parameter optimization processes for θ and ψ are independent. According to lemma 3, we can transform the original optimization problem into the following problem:

$$\begin{aligned}
&\min_{\theta,\psi}\Big(KL(\Gamma^g||\Gamma^\theta)+KL(\hat{\Gamma}^\theta||\Gamma^\psi)+\mu Reg_1(\Gamma^\psi)+\nu Reg_2(\Gamma^\theta))\Big)\\
&\text{subject to}\quad \Gamma^\theta=\frac{exp(s_{ii}(\theta)/\tau)}{\sum_{j\neq i}Nexp(s_{ij}(\theta)/\tau)},\\
&\qquad\qquad\quad \Gamma^\psi=\frac{exp(s_{ii}(\psi)/\tau)}{\sum_{j\neq i}Nexp(s_{ij}(\psi)/\tau)},\\
&\qquad\qquad\quad \hat{\Gamma}^\theta=\mathcal{T}(f_\theta^{fixed},g_\psi^{fixed},X_{sup},X_{full})
\end{aligned}\tag{11}$$

Due to

$$KL(p||q)=H(p,q)-H(p)\tag{12}$$

where $H(p)$ represents the entropy of p, and $H(p,q)$ represents the cross-entropy between p and q. Given the selection of a specific $\mathcal{T}$ to derive $\hat{\Gamma}^\theta$, we can thus determine $\hat{\Gamma}^\theta(i,j)$ as a fixed quantity, making $H(p)$ constant. Furthermore, we can obtain

$$\begin{aligned}
\min_{\theta,\psi}\Bigg(&-\sum_{i=1}^{n}\log\left(\frac{\exp(s_{ii}(\theta)/\tau)}{\sum_{j\neq i}\exp(s_{ij}(\theta)/\tau)}\right)\\
&-\sum_{i=1}^{n}\sum_{j=1}^{n}\hat{\Gamma}^\theta(i,j)\log\left(\frac{\exp(s_{ij}(\psi)/\tau)}{\sum_{k\neq i}\exp(s_{ik}(\psi)/\tau)}\right)\\
&+\mu\,\mathrm{Reg}_1(\Gamma^\psi)+\nu\,\mathrm{Reg}_2(\Gamma^\theta)\Bigg)\\
\text{where}\quad&\hat{\Gamma}^\theta(i,j)=\mathcal{T}(f_\theta^{\text{fixed}},g_\psi^{\text{fixed}},X_{\text{sup}},X_{\text{full}})_{(i,j)}
\end{aligned}\tag{13}$$

C.2 Proof for Proposition 2

Since $C^{\theta^{fixed}}$ does not involve parameter optimization of θ, we can still calculate the value of $\hat{\Gamma}^\theta(i,j)$. According to the proof in C.1, we know that

$$\begin{aligned}
\min_{\theta,\psi}\Bigg(&-\sum_{i=1}^{n}\log\left(\frac{\exp(s_{ii}(\theta)/\tau)}{\sum_{j\neq i}\exp(s_{ij}(\theta)/\tau)}\right)\\
&-\sum_{i=1}^{n}\sum_{j=1}^{n}\hat{\Gamma}^\theta(i,j)\log\left(\frac{\exp(s_{ij}(\psi)/\tau)}{\sum_{k\neq i}\exp(s_{ik}(\psi)/\tau)}\right)\\
&+\mu\,\mathrm{Reg}_1(\Gamma^\psi))\Bigg)\\
\text{where}\quad&\hat{\Gamma}^\theta(i,j)=\arg\min_{\Gamma\in U(a,b),\,a=\mathbf{1}_N,\,b=\mathbf{1}_N}\left(\langle C^{\theta^{\text{fixed}}},\Gamma\rangle+\frac{\lambda}{2}\|\Gamma\|^2\right)_{(i,j)}
\end{aligned}\tag{14}$$

When we choose Reg_1 as the regularization term 4, we obtain the optimization formulation of S-MolSearch.

D Reproduce details

For the training of S-MolSearch , we use Adam optimizer at a learning rate of 0.001. The batch size is 128, and the training is conducted on 4 NVIDIA V100 32G GPUs. For backbone model, we use the same parameters as Uni-Mol. To enhance the training, we retain the masking and coordinate noise addition for atoms within molecules, as implemented in Uni-Mol. The parameters are identical to those in Uni-Mol, with a masking ratio of 0.15 and noise following a uniform distribution between -1 and 1 Å. We randomly sample data from the labeled and unlabeled datasets to form the validation set, and select checkpoints based on the loss. More detailed configurations can be found in the code repository.

E Extending to the Few-Shot Setting

Extending S-MolSearch from a zero-shot to a few-shot setting is feasible. Due to the lack of a universal few-shot setup standard in this scenario, we explore two few-shot settings. For data splitting in both settings, we randomly selected 70% of the active molecules from each target in DUD-E as the training set for few-shot learning, while the remaining 30% and all inactive molecules serve as test data. The initial training set includes approximately 16,000 molecules, while the test dataset contains around 1,418,000 molecules. From these 16,000 active molecules, we randomly selected 50,000 pairs as the contrastive learning training data. In the first setting, we simply add the query molecule to the active molecule set corresponding to the target and randomly sample pairs, denoted as R in Table 5. Pairs involving molecules bound to the same target are treated as positive, while those involving molecules bound to different targets are treated as negative. In the second setting, we fix one side of each contrastive learning pair as the query molecule, denoted as F in Table. If the other molecule in the pair is an active molecule bound to the same target, it is considered a positive pair; if bound to a different target, it is considered a negative pair. From Table 5, it can be seen that S-MolSearch performs better in the few-shot setting than in the zero-shot setting, indicating its potential in few-shot scenarios.

Table 5: Performance on DUD-E in two few-shot settings.

Method	AUROC (%)	EF 0.5%	EF 1%	EF 5%
$\text{R}_{\text{zero-shot}}$	85.38	79.08	47.12	11.82
$\text{R}_{\text{few-shot}}$	97.21	154.90	86.00	18.48
$\text{F}_{\text{zero-shot}}$	84.87	79.07	46.45	11.70
$\text{F}_{\text{few-shot}}$	98.32	165.09	89.98	19.07

F Qualitative Examples of Similarities

We provide qualitative examples to enhance understanding of S-MolSearch's capabilities. Specifically, we select two targets, hdac2 and csf1r, from DUD-E. In Figure 4, we present the query molecules along with the top-ranked molecules retrieved by S-MolSearch, all of which are active molecules. We provide the embedding similarity and the Tanimoto similarity of molecular fingerprints between these molecules and the query molecule. The results indicate that molecules with high Tanimoto similarity also tend to have high embedding similarity.

G Limitations

S-MolSearch falls short in terms of interpretability. Traditional 3D molecule search methods can capture shape and functional features required for biological interactions, providing scientists with mechanistic insights. S-MolSearch is currently unable to provide such intuitive insights for molecule search. Additionly, its use of two encoders also leads to higher memory consumption than single-encoder methods.

H Potential societal impacts

S-MolSearch performs well on LBVS and can help screen bioactive molecules from large molecule databases. This eases the workload of medicinal chemists and may accelerate the discovery of new drug. On the downside, S-MolSearch also runs the risk of being used inappropriately, such as when it is used to search for similar molecules to drugs that are addictive.

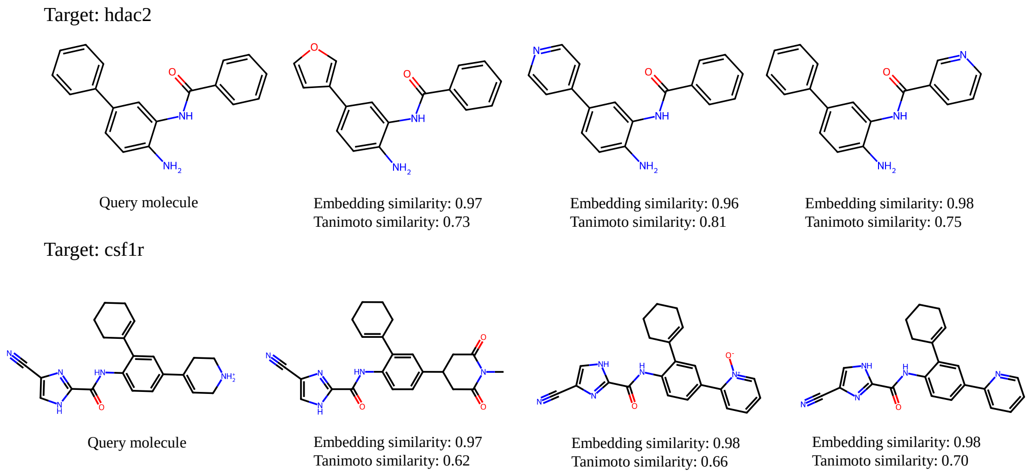

Target: hdac2
Query molecule
Embedding similarity: 0.97
Tanimoto similarity: 0.73
Embedding similarity: 0.96
Tanimoto similarity: 0.81
Embedding similarity: 0.98
Tanimoto similarity: 0.75
Target: csf1r
Query molecule
Embedding similarity: 0.97
Tanimoto similarity: 0.62
Embedding similarity: 0.98
Tanimoto similarity: 0.66
Embedding similarity: 0.98
Tanimoto similarity: 0.70

Figure 4: Qualitative examples of similarities for targets hdac2 and csf1r in DUD-E.

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: The main claims articulated in the abstract and introduction accurately mirror the paper's contributions and scope.

   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: The paper outlines the limitations of the research in appendix.

   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: [Yes]

   Justification: We give hypotheses and proofs in the main text and appendix.

   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: The paper provides comprehensive details necessary for reproducing the main experimental results.

   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: The code, model, and data are made publicly available upon acceptance.

   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: The paper meticulously details all aspects of the training and testing process. It specifies the data splits used for training, validation, and testing, along with a thorough description of the hyperparameters and their selection criteria.

   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: [No]

Justification: We verify the robustness of the algorithm by validating it on large-scale data.

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 provide this in appendix.

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 conducts 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 provide this in appendix.

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: The 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: The paper does not use existing assets.

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: [NA]

    Justification: The paper does not release new assets.

    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.