

REAP: Towards Effective Training-Free Chemical Reasoning with Explicit Atomic Priors

Mingxu Zhang¹, Dazhong Shen^{2*}, Qi Zhang³, Ying Sun^{4*}

¹The Hong Kong University of Science and Technology (Guangzhou)

²Nanjing University of Aeronautics and Astronautics ³Shanghai AI Laboratory

⁴The 63rd Research Institute, National University of Defense Technology, Nanjing
mzhang630@connect.hkust-gz.edu.cn, shendazhong@nuaa.edu.cn,
zhangqi.fqz@gmail.com, sunyinggilly@gmail.com

Abstract

Large Language Models (LLMs) exhibit strong general reasoning but struggle in molecular science due to the lack of explicit priors required for precise chemical reasoning. Current solutions inject priors into parameters, but this static coupling hinders rapid knowledge updates and often compromises the model’s general capabilities. To address this, we introduce **REAP**, a training-free framework that equips LLMs with an external knowledge base, enabling them to reason over retrieved chemical priors dynamically. REAP implements a structured reasoning pipeline that autonomously selects relevant priors from our constructed atom-level knowledge base, retrieves analogue exemplars, and synthesizes these information to guide the LLM’s decision-making. This architecture ensures interpretability and adaptability while preserving the LLM’s intrinsic general intelligence. Experiments show that REAP outperforms current reasoning methods and rivals state-of-the-art training-based models, demonstrating the effectiveness of our framework.

1 Introduction

LLMs exhibit strong reasoning capabilities and have already revolutionized logic-intensive domains like recommendation systems (Xin et al., 2025) and can also be applied to social science (Cui et al., 2026; Li and Li, 2026). Inspired by this success, there is a growing consensus that LLMs hold immense potential for molecular science. Since chemical reasoning inherently relies on vast domain knowledge and rigorous logical deduction, LLMs are naturally suited to tackle these challenges. Consequently, they are envisioned as powerful engines to accelerate discovery in drug design and material synthesis.

To translate this potential into accurate molecular predictions, the research community has ac-

tively explored strategies to instill effective chemical reasoning capabilities into LLMs. The most intuitive strategy is to directly adapt general reasoning paradigms to the chemical domain (Xian et al., 2025; Chen et al., 2025; Zhang et al., 2025b; Zheng et al., 2025). However, these methods often suffer from an information bottleneck; by relying on coarse string representations or simple molecule-level descriptors, they fail to provide the fine-grained, atom-level priors essential for precise reasoning. To address this deficit, researchers have pivoted toward training-based methods, which seek to inject these missing priors by encoding them directly into the model’s parameters through domain-specific training (Cao et al., 2023; Liu et al., 2023c; Zhang et al., 2024; Su et al., 2022; Luo et al., 2023). Yet, this paradigm introduces a fundamental dilemma: it implicitly bakes knowledge into static parameters, resulting in inherent rigidity where updating knowledge requires retraining and risks forgetting of the LLM’s general reasoning capabilities.

Consequently, it is better to equip frozen LLMs with explicit, externalized chemical priors, combining the precision of domain knowledge with the flexibility of generalist models. However, this idea faces two fundamental challenges. First, a representation gap: grounded reasoning relies on fine-grained atomic priors (e.g., charge, local topology) inherent to the molecular structure, yet no existing representation explicitly exposes these details. Second, a reasoning gap: LLMs struggle to distinguish chemically meaningful signals from irrelevant noise. Without a mechanism to autonomously select valid evidence, they fail to construct coherent logical chains from dense raw chemical data.

To address these challenges, we introduce REAP (**R**easoning with **E**xplicit **A**tomic **P**riors), an effective training-free framework that equips LLMs with an external knowledge base, enabling them to reason over retrieved chemical priors dynamically. To bridge the representation gap, REAP

*Corresponding authors.

constructs the first atom-level textual knowledge base by discretizing atoms into tokens, externalizing fine-grained physicochemical attributes into a human-readable format accessible to frozen LLMs. Crucially, to overcome the reasoning gap, we design a mechanism that helps LLMs select task-relevant evidence from dense priors. REAP autonomously selects task-relevant priors to focus on discriminative chemical features, and retrieves similar molecules to serve as empirical anchors. By integrating explicit priors with concrete examples, it enables the LLM to derive precise, evidence-based predictions. Empirically, we demonstrate that this strategy is effective: REAP outperforms existing reasoning frameworks and rivals or surpasses specialized training-based models, proving that externalized reasoning is a robust alternative to parametric learning.

In summary, our contributions are threefold:

- We introduce REAP, an effective training-free reasoning framework that equips LLMs with an external knowledge base. The framework enables frozen LLMs to perform precise, interpretable reasoning using explicit priors.
- We construct the first atom-level textual knowledge base, which externalizes fine-grained physicochemical priors into a human-readable format, effectively bridging the gap inherent in standard linear representation.
- Experiments demonstrate that REAP outperforms existing reasoning frameworks and rivals several training-based models, validating that explicit prior injection is a competitive alternative to implicit parameter updates.

2 Related Work

Chemical Large Language Models. Recent efforts to equip LLMs with chemical expertise generally follow two paradigms. The first involves training-based adaptation, ranging from domain-specific pre-training on chemical corpora (Zhao et al., 2024; Liu et al., 2023b) to multi-modal alignment where molecular graph features are projected into the LLM’s embedding space (Cao et al., 2023; Liu et al., 2023b). Some approaches further expand the vocabulary to include discrete chemical tokens (Zhang et al., 2024, 2025a; Wang et al., 2024). The second paradigm explores retrieval-augmented generation, where models retrieve analogues or utilize external tools to aid generation (Xian et al.,

2025; M. Bran et al., 2024; Liu et al., 2024; Lee et al., 2024). However, training-based methods suffer from computational costs and parameter rigidity, while existing RAG approaches often rely on coarse instance retrieval, lacking the evidence grounding necessary for precise reasoning.

Reasoning with LLMs. LLM reasoning has evolved significantly through diverse prompting and grounding strategies. Foundational paradigms like **Chain-of-Thought (CoT)** (Wei et al., 2022; Kojima et al., 2022) and **Tree of Thoughts (ToT)** (Yao et al., 2023; Long, 2023) empower models to decompose complex tasks into intermediate steps, while **In-Context Learning (ICL)** (Brown et al., 2020; Dong et al., 2024) leverages few-shot demonstrations to steer reasoning without parameter updates. To further enhance fidelity, **Retrieval-Augmented Generation (RAG)** (Lewis et al., 2020; Guu et al., 2020) and verification-based frameworks (Dhuliawala et al., 2024; Jiang et al., 2023) explicitly anchor inference to external knowledge. However, the direct application of these general-purpose strategies to chemistry is limited, as standard 1D molecular representations lack the explicit, fine-grained atomic priors necessary to support such rigorous, evidence-based deduction.

3 Method

We bridge two fundamental gaps that hinder LLMs in molecular science. To address the representation gap, we construct the first atom-level textualized knowledge base (Sec. 3.1) by discretizing physicochemical attributes into tokens for direct retrieval. To address the reasoning gap, we propose REAP (Fig. 1), which consists of two modules: a prior selection module (Sec. 3.3) that selects task-relevant fields, and an evidence acquisition module (Sec. 3.4) that retrieves priors and analogues. Together, these modules build a focused, high-quality input that grounds the LLM’s final inference.

3.1 Atomic Prior Knowledge Base

Effective chemical reasoning requires integrating priors—fundamental physicochemical attributes that govern molecular properties (e.g., aqueous solubility or membrane permeability)—at two granularities. While molecule-level priors are easily computed and provide macroscopic constraints, they fail to capture the local nuances driving reactivity. Precise prediction needs fine-grained atom-level priors such as hydrogen-bonding roles to identify

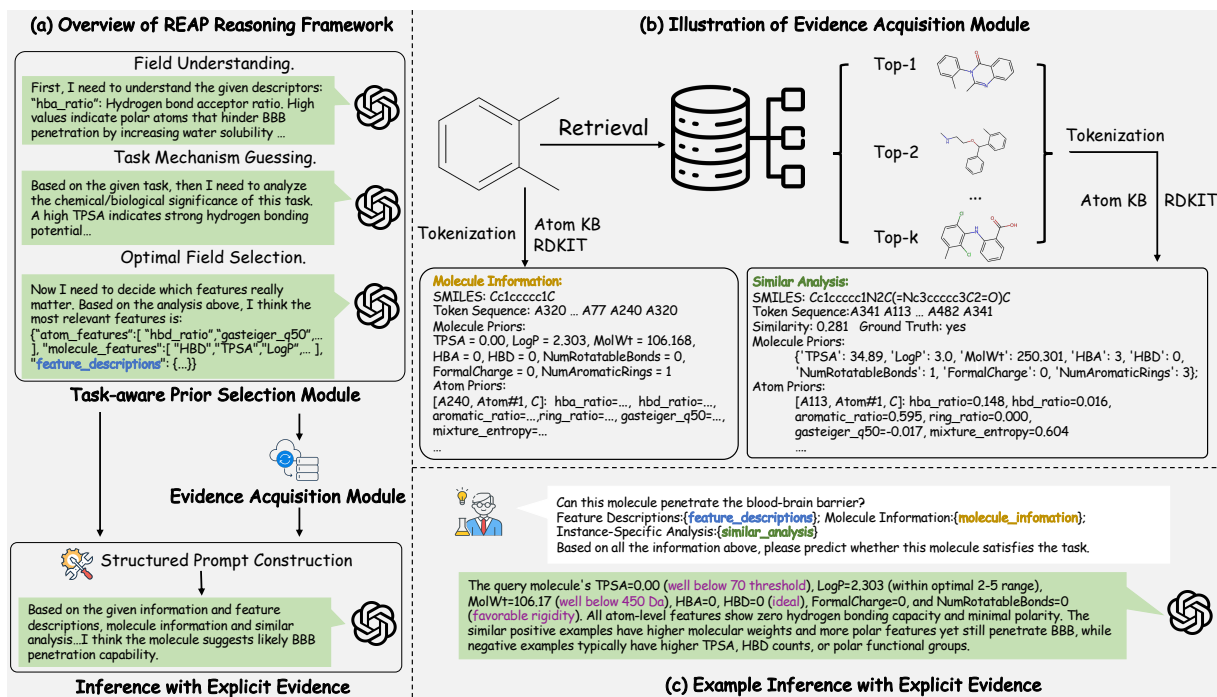


Figure 1: Overview of the REAP framework. **(a) The Reasoning Pipeline.** The workflow progresses from autonomous prior selection where the LLM determines critical features, to the retrieval module and final inference. **(b) Retrieval Module Illustration.** Shows how top- k analogues are retrieved and populated with fine-grained priors from the knowledge base to form the context. **(c) Integrative Inference Example.** Demonstrates the final structured prompt and the model’s explicit, evidence-based reasoning output for a specific task.

reactive centers and binding sites. However, providing these priors presents a challenge of information overload: a single molecule contains many atoms, each possessing multiple numerical attributes. Injecting these raw arrays directly into the context would overwhelm the LLM with noise. To address this, we use discrete tokens to represent clusters of similar atomic environments and construct a knowledge base that maps each token to a structured profile of its physicochemical statistics which enables the LLM to retrieve priors via discrete indices, converting raw data into a queryable format.

Atom Tokenization. To transform chemical environments into discrete, retrievable units, we first extract and tokenize atom-level representations. To this end, we utilize the AtomDisc tokenizer (Zhang et al., 2025a). Specifically, given a molecular graph $G = (V, E)$, we first employ the pre-trained molecule (Liu et al., 2023a) encoder to map each atom v to a context-aware embedding $\mathbf{r}_v \in \mathbb{R}^d$. To discretize these continuous vectors, a vector-quantized codebook $\mathcal{C} = \{\mathbf{c}_j\}_{j=1}^K$ is used to map each atom to the index of its nearest codeword:

$$l_v = \arg \min_{1 \leq j \leq K} \|\mathbf{r}_v - \mathbf{c}_j\|_2^2.$$

The resulting discrete identifier A_{l_v} serves as the atomic token, and the molecule is represented as a sequence of these tokens $\mathbf{A}(G) = (A_{l_{v_1}}, \dots, A_{l_{v_{|V|}}})$, which functions as the primary key for our textualized knowledge base.

Knowledge Base Construction. We build the atomic prior knowledge base from a large corpus of molecules sampled from PubChem (Kim et al., 2023). For each molecule, we derive its AtomDisc atom-token sequence $\mathbf{A}(G)$ and compute RDKit atom-level descriptors. For each atom token t , we aggregate all instances assigned the same index t in the corpus to form a token-level prior profile. We summarize numeric attributes with robust statistics (median and interquartile range) to represent the typical descriptor values of this atomic cluster, and summarize categorical/boolean attributes by their modes and empirical distributions. To capture typical local environments, we compute the Pointwise Mutual Information (PMI) between token t and its 1-hop neighbor tokens t' , $\text{PMI}(t, t') = \log \frac{P(t, t')}{P(t)P(t')}$, and keep neighbors with the highest PMI scores. Finally, we materialize the knowledge base as a JSON-style mapping from atom tokens to their prior fields: $\mathcal{D} = \{(t, V_t)\}$, where the key t is an AtomDisc

atom token, and the value $V_t = \{(\mathcal{P}_{all}, \mathcal{V}_t)\}$ contains the full prior field set \mathcal{P}_{all} and the corresponding values $\mathcal{V}_t = \{v_{t,p} \mid p \in \mathcal{P}_{all}\}$.

3.2 Overview of REAP Reasoning Framework

Building upon the Atomic Textualized knowledge base, REAP executes a structured pipeline to dynamically utilize this knowledge for prediction. This framework transforms raw instructions into verified predictions through a clear process:

Module 1: Task-aware Prior Selection. While the knowledge base provides a comprehensive dictionary of atomic priors, effective reasoning requires focusing on the specific field relevant to the task at hand (e.g., solubility depends on polarity, while toxicity involves reactive groups). This module addresses the need for task adaptation. Taking the task instruction \mathcal{T} and all priors fields \mathcal{P}_{all} as input, it analyzes which field are necessary for the specific task. Then, it will select a subset of priors $\mathcal{P}^* \subset \mathcal{P}_{all}$ and their explanation H_{field}^* .

Module 2: Evidence Acquisition. With the relevant priors identified, the model now requires actual evidence to perform the reasoning. It performs two specific actions: (1) Prior Retrieval: For the target molecule, we collect the values of the selected fields \mathcal{P}^* from two sources: (i) molecule-level priors are computed using RDKit, and (ii) atom-level priors are retrieved from the atomic knowledge base by querying with the molecule’s AtomDisc tokens as keys. These priors are assembled into the evidence packet E_q , providing the explicit facts used for reasoning. (2) Analogue Retrieval: It retrieves the top- k similar molecules from a reference dataset \mathcal{D} (e.g. training set) and populates them with the same priors to form the context $\mathcal{C}_{context}$. The output is the combination of the target’s explicit facts E_q and the retrieved analogue $\mathcal{C}_{context}$.

Inference with Explicit Evidence. Finally, to reason with explicit chemical priors, we synthesize the prepared evidence into a complete reasoning context. We construct a final input $\mathcal{I}_{final} = [\mathcal{T}, H_{field}^*, \mathcal{C}_{context}, E_q]$. The model is instructed to generate a reasoning chain R that compares the target packet E_q against the examples in $\mathcal{C}_{context}$ before predicting the label Y . The inference follows the probability maximization:

$$Y^* = \arg \max_Y P(Y \mid R, \mathcal{I}_{final}).$$

3.3 Task-aware Prior Selection Module

The knowledge base contains a large set of priors, but using all of them would introduce substantial noise and dilute the evidence needed for a specific task. Module 1 therefore performs task-aware prior selection: it reads the instruction and decides which fields are actually useful for the prediction. To make this selection reliable and interpretable, we implement the module as a three-step procedure:

Field Understanding. The model first reads the full prior list \mathcal{P}_{all} and produces a plain-language explanation H_{field} for each field:

$$H_{field} = \text{LLM}(p_f, \mathcal{P}_{all}),$$

where p_f asks the model to explain each field, and H_{field} provides the chemical meaning of priors. This step ensures the model understands what each available field represents.

Task Mechanism Guessing. Conditioned on H_{field} and the task instruction \mathcal{T} , the model analyzes which fields are likely relevant for this task and produces a reasoning summary:

$$H_{logic} = \text{LLM}(p_t, \mathcal{T} \mid H_{field}),$$

identifies which fields to prioritize for the task.

Optimal Field Selection. Finally, conditioned on H_{logic} , the model selects the task-relevant prior subset \mathcal{P}^* and the corresponding explanations H_{field}^* using prompt p_s :

$$\mathcal{P}^* = \text{LLM}(p_s \mid H_{logic}).$$

We provide the full field of available priors in Appendix B and the corresponding prompt templates for the automated pipeline in Appendix E.1.

3.4 Evidence Acquisition Module

This module provides explicit evidence for the final decision: it retrieves structurally similar molecules from a reference database and summarizes both the query and the retrieved molecules using the task-selected priors \mathcal{P}^* from last module.

Analogue Retrieval. Given a query molecule M_q and a reference database \mathcal{D} , we retrieve analogues to provide comparable evidence for prediction by structural similarity. We use Morgan fingerprints because they encode local circular substructures and bonding patterns, which are strong signals for property similarity. Let \mathbf{v}_q and \mathbf{v}_i denote the

Table 1: Comparison of REAP with baselines on classification benchmarks (AUROC). *DA* prompts LLMs to answer directly from the given molecule, whereas *CoT* prompts LLMs to explicitly reason about the molecule. Prompt templates for both settings are provided in Appendix E.3. All baseline results are taken as reported in their respective papers and entries marked “—” were not reported in the original papers.

Dataset	REAP	DA	CoT	MolRAG	LLM4SD	InstructMol	MoMu	GraphMVP	UniMol
BACE	0.8970	0.6428	0.6775	0.7673	0.8383	0.8230	0.7670	0.8120	0.8570
BBBP	0.8731	0.4918	0.6121	0.6291	0.7560	0.7000	0.7050	0.7240	0.7290
ClinTox	0.8856	0.5015	0.4860	—	0.9130	0.9150	0.7990	0.7910	0.9190
HIV	0.7270	0.6452	0.6266	0.6476	0.7901	0.6890	0.7590	0.7700	0.8080
SIDER	0.7179	0.6371	0.6279	—	0.6540	0.5780	0.6050	0.6390	0.6590
Tox21	0.8023	0.5685	0.5958	0.6642	0.7470	0.7470	0.7560	0.7590	0.7960
Avg	0.8172	0.5812	0.6043	0.6771	0.7831	0.7420	0.7318	0.7492	0.7947

Morgan fingerprint vectors of M_q and $M_i \in \mathcal{D}$, respectively. We compute the Tanimoto similarity:

$$\text{Sim}(M_q, M_i) = \frac{\mathbf{v}_q \cdot \mathbf{v}_i}{\|\mathbf{v}_q\|_1 + \|\mathbf{v}_i\|_1 - \mathbf{v}_q \cdot \mathbf{v}_i},$$

and retrieve the top- k nearest neighbors:

$$\mathcal{N}_k = \{M_i \mid M_i \in \text{Top-}k(\mathcal{D}, M_q)\}.$$

Task-Adaptive Evidence Packing. For the query molecule M_q and each retrieved analogue $M_i \in \mathcal{N}_k$, we build evidence packets by combining molecule-level priors computed on-the-fly and atom-level priors retrieved from our knowledge base. Molecule-level priors are computed using RDKit descriptors. For atom-level priors, we obtain the AtomDisc token sequence $\mathbf{A}(G)$ and use each atom token as the retrieval key to query the atomic prior knowledge base \mathcal{D} . For token index ℓ , we retrieve its key-value record $\mathcal{D}(\ell)$ and keep only the task-selected fields \mathcal{P}^* . To avoid excessive and uninformative details from molecules with many atom tokens, we apply an atom-selection function $\phi(\cdot)$ and keep only $\mathcal{A}_f = \{a \in \mathcal{A} \mid \phi(a) = 1\}$ (details in Appendix A.2). We denote the resulting packet construction as $E(\cdot; \mathcal{P}^*, \mathcal{A}_f)$. The query packet $E_q = (E(M_q; \mathcal{P}^*, \mathcal{A}_f), \mathbf{A}(G_q))$ provides the explicit facts of the target molecule for the final decision, and each analogue packet is $E_i = E(M_i; \mathcal{P}^*, \mathcal{A}_f)$ for $M_i \in \mathcal{N}_k$. The final example context for inference:

$$\mathcal{C}_{\text{context}} = [(E_i(\mathcal{P}^*, \mathcal{A}_f), y_i, s_i, \mathbf{A}(G_i))]_{i=1}^k,$$

where y_i is the ground-truth label of analogue M_i , s_i is the Tanimoto similarity value.

4 Experiments

We evaluate REAP on standard molecular property benchmarks to answer four research questions (RQs). **RQ1:** Does REAP significantly outperform existing training-free baselines, and can it rival the performance of state-of-the-art training-based models? **RQ2:** What are the individual contributions of atom-level and molecule-level priors? Are they complementary distinct sources of information essential for optimal performance? **RQ3:** What is the impact of retrieved analogues on performance, and does REAP maintain robustness in zero-shot settings? **RQ4:** Is REAP a model-agnostic framework that consistently enhances performance across different LLM backbones?

We address **RQ1** in Section 4.3; **RQ2** via component analysis in Section 4.4.1; **RQ3** via retrieval ablation in Section 4.4.2; and **RQ4** in Section 4.5.

4.1 Experimental Setup

We evaluate REAP on MoleculeNet (Wu et al., 2018) classification benchmarks and on widely used physicochemical/pharmacokinetic datasets: ESOL (Delaney, 2004; Wu et al., 2018), FreeSolv (Mobley and Guthrie, 2014; Wu et al., 2018), Lipophilicity (Lipo) (Wu et al., 2018), Caco2 permeability (Wang et al., 2016), LD50 (Zhu et al., 2009), and AqSol (Sorkun et al., 2019). These tasks span solubility (ESOL, FreeSolv, AqSol), lipophilicity (Lipo), absorption (Caco2), and toxicity (LD50). Unless otherwise noted, we report AUROC for classification and RMSE (\downarrow) for regression, following common practice; dataset details and splits are in Appendix A.1. The hyperparameters we used are provided in Appendix A.3.

Table 2: Ablation study of REAP along two axes: (i) the contribution of chemical priors (atom and molecule level) and (ii) the effect of the Top- k selection. Classification results (BACE, BBBP, ClinTox) are reported as AUROC (\uparrow), and regression results (Caco-2, FreeSolv, Lipophilicity) as RMSE (\downarrow).

Dataset	REAP	w/o Atom. Pr.	w/o Mol. Pr.	$k = 0$	$k = 2$	$k = 4$	$k = 6$	$k = 8$	$k = 10$
BACE	0.8042	0.7290	0.8207	0.6694	0.6594	0.8223	0.8222	0.8511	0.8042
BBBP	0.8246	0.8131	0.7176	0.7887	0.7712	0.8095	0.8094	0.8181	0.8246
ClinTox	0.8163	0.7974	0.7072	0.5902	0.7288	0.8018	0.7741	0.7664	0.8163
Avg. AUROC	0.8150	0.7798	0.7485	0.6828	0.7198	0.8112	0.8019	0.8119	0.8150
Caco-2	0.5018	0.5271	0.5028	2.2123	0.5825	0.5261	0.5018	0.4935	0.5041
FreeSolv	0.9008	1.0210	0.9776	2.8607	1.2379	1.0253	0.9008	1.0540	1.0177
Lipo	0.7022	0.7917	0.7844	1.1916	0.7679	0.7674	0.7022	0.7327	0.7143
Avg. RMSE	0.7016	0.7799	0.7549	2.0882	0.8628	0.7729	0.7016	0.7601	0.7454

4.2 Baseline

To comprehensively evaluate our REAP framework, we consider a broad set of baseline methods spanning training-free, LLM-based, graph-based, and 3D molecular modeling approaches. Specifically, we include training-free methods such as MolRAG (Xian et al., 2025) and LLM4SD (Zheng et al., 2025). For LLMs with task-specific training, we compare against InstructMol (Cao et al., 2023) and MoMu (Su et al., 2022). In addition, we incorporate representative graph-based baselines, including GraphMVP (Liu et al., 2022) and MoleculeSTM (Liu et al., 2023a), as well as the 3D molecular modeling methods UniMol (Zhou et al., 2023) and TokenMol (Wang et al., 2025).

4.3 Main Results

We summarize the overall performance against both training-free and training-based baselines in Table 1. These findings provide a definitive affirmative answer to **RQ1**. First, REAP consistently achieves the best performance among training-free methods, significantly outperforming baselines that rely on surface-level prompting. Second, and more notably, it successfully rivals and in several cases surpasses representative training-based approaches such as MoMu (Su et al., 2022), InstructMol (Cao et al., 2023). This demonstrates that explicitly injecting fine-grained priors is not merely a compromise for efficiency, but a robust strategy that allows frozen LLMs to match the predictive power of fully fine-tuned graph encoders.

Beyond numerical accuracy, REAP offers a qualitative advantage in transparency (example outputs in Appendix F.2). Unlike end-to-end neural base-

lines where decision boundaries are opaque, REAP provides traceable reasoning paths. By inspecting the model’s articulated reasoning regarding the selected priors and their stated implications for the prediction, domain practitioners can verify whether the decision is based on sound chemical principles or spurious correlations. This transparency fosters trust, ensuring that high performance is achieved for the right reasons.

We further evaluate REAP on regression tasks; complete results are provided in Appendix G.

4.4 Ablation Study

We conduct ablation studies to isolate the impact of two core design components: (i) the contribution of multi-level textualized priors, and (ii) the role of retrieved exemplars in grounding reasoning.

4.4.1 Contribution of Multi-Level Priors

To answer **RQ2**, we compare the full REAP against variants that strip away specific prior levels while keeping other settings fixed. *w/o Atom. Pr.* removes the atom-level textualized priors, and *w/o Mol. Pr.* removes the molecule-level descriptors.

Results in Table 2 reveal that the full REAP consistently achieves the best performance, confirming that atom- and molecule-level priors are not redundant but synergistic. Specifically, removing atom-level priors causes marked degradation on datasets driven by specific pharmacophores or local environments (e.g., lower AUROC on BACE, higher RMSE on FreeSolv), highlighting the necessity of fine-grained local information. Conversely, removing molecule-level priors also hurts performance (e.g., on BBBP and ClinTox), suggesting that global physicochemical context (e.g., solubil-

Table 3: Comparison across zero-shot REAP (ZS-REAP), few-shot MolRAG (FS-MolRAG), and chain-of-thought prompting in zero-shot (ZS-CoT) and few-shot (FS-CoT) settings. Classification performance is reported as AUROC (\uparrow), and regression performance as RMSE (\downarrow). The best results are shown in **bold**, and the runner-up results are underlined.

Dataset	ZS-REAP	FS-MolRAG	ZS-CoT	FS-CoT
<i>Classification (AUROC \uparrow)</i>				
BACE	0.6694	0.7673	0.6475	0.6373
BBBP	0.7887	0.6291	0.6121	0.6660
ClinTox	<u>0.5902</u>	—	0.4860	0.6283
<i>Regression (RMSE \downarrow)</i>				
ESOL	0.6315	3.2806	0.8944	0.9134
FreeSolv	2.8607	6.0827	3.4312	3.2863
Lipophilicity	1.1013	1.1251	1.2098	1.2479

ity, weight) provides essential macroscopic constraints. The aggregate trend demonstrates that integrating these two levels of granularity yields the most robust reasoning, validating our design of a comprehensive, multi-level knowledge base.

4.4.2 Top- k Sensitivity Analysis

To address **RQ3**, we examine the effect of analogue retrieval by varying $k \in \{0, 2, 4, 6, 8, 10\}$. Here, $k=0$ represents the Zero-Shot setting, where REAP relies solely on its internal knowledge base.

Table 2 shows a clear trend: moving from $k=0$ to a small positive k yields substantial gains. This confirms that retrieved analogues successfully activate the LLM’s capability for reasoning by analogy, allowing it to validate hypotheses against concrete examples. Notably, performance saturates around $k \approx 4-6$, suggesting that REAP is highly efficient—it does not require a large crowd of examples, but rather a handful of high-quality anchors to ground its decision.

We further evaluate whether REAP remains effective without any exemplars. As shown in Table 3, Zero-Shot REAP (ZS-REAP) is competitive with, and often superior to, standard few-shot baselines. Remarkably, ZS-REAP outperforms Few-Shot CoT on nearly all datasets and surpasses Few-Shot MolRAG on **4/5** comparable benchmarks. This result is particularly significant: it implies that access to a high-quality explicit knowledge base allows the model to reason effectively even without specific examples, validating our core premise that explicit grounding can effectively substitute or outperform for surface-level pattern matching from few-shot prompting.

4.5 Generalization across LLM Backbones

To verify the universality of our framework, we instantiate REAP across a diverse spectrum of LLM backbones—ranging from efficient distilled models (GPT-5-Nano) to frontier-class reasoning engines (DeepSeek-V3.2, GPT-5, GPT-4.1, Claude Sonnet-4). All experiments use identical prompts and retrieval pipelines without specific tuning.

Table 4 reveals a universal enhancement pattern. REAP consistently achieves the best average performance across all backbones compared to standard prompting strategies (DA, CoT, and their few-shot variants) for both classification and regression tasks. Crucially, we observe a democratizing effect: smaller models equipped with REAP frequently rival or even surpass larger models relying on standard prompting. This suggests that the bottleneck in chemical reasoning is often the lack of explicit domain priors rather than raw parametric capacity. These results affirmatively answer **RQ4**: REAP is a robust, backbone-agnostic framework. By decoupling knowledge from the model, it serves as a plug-and-play enhancement layer that upgrades any LLM—regardless of vendor, architecture, or scale—into a grounded chemical reasoner.

4.6 Case Study: From Surface Heuristics to Evidence-Based Verification

To demonstrate how REAP utilizes explicit priors to ground the reasoning process, Fig. 2 presents a side-by-side comparison on a BACE-1 inhibition task. This comparison illustrates the shift from implicit pattern matching to quantitative verification.

The SMILES-only baseline fails due to the opacity of the linear representation. While it correctly identifies a potentially active scaffold, it lacks the granular information necessary to verify the physicochemical compatibility. Constrained by this information deficit, the model relies on heuristic-driven inference, falsely associating the mere presence of a drug-like scaffold with biological activity.

In contrast, REAP equips the LLM with specific, textualized priors such as the precise aromatic ring count, TPSA, and LogP which serve as quantitative evidence for reasoning. This explicit grounding enables the model to validate the molecule against established structure-activity relationships. By shifting the paradigm from logical hallucination to fact-conditioned reasoning, REAP successfully rejects the superficially promising candidate based on rigorous physicochemical constraints.

Table 4: Results across different LLMs integrated with REAP on classification (AUROC \uparrow) and regression (RMSE \downarrow) tasks. To highlight the advantage of explicit chemical priors over single-SMILES prompting, we compare four inference strategies: DA+FS, CoT+FS, DA, and CoT. Here, *DA* denotes directly answering from the prompt, *CoT* denotes reasoning, and *FS* indicates few-shot examples.

Task (Avg over datasets)	LLM	REAP	DA+FS	CoT+FS	DA	CoT
Regression	Claude-Sonnet-4	0.7442	1.5560	1.6815	1.8262	1.8878
	Deepseek-V3.2	0.7941	1.6854	1.8159	1.6081	1.8451
	GPT-4.1	0.8214	2.5202	2.0661	3.7289	4.5305
	GPT-5	0.7352	1.5129	1.6619	2.0690	1.9961
	GPT-5-Nano	0.8263	1.9909	1.8948	2.8695	3.1598
	Avg. RMSE \downarrow	0.7842	1.8531	1.8240	2.4204	2.6839
Classification	Claude-Sonnet-4	0.7938	0.5950	0.6503	0.6359	0.6476
	Deepseek-V3.2	0.8150	0.6599	0.6439	0.6601	0.7266
	GPT-4.1	0.8010	0.6669	0.6927	0.5952	0.5823
	GPT-5	0.8852	0.7199	0.7159	0.7898	0.6889
	GPT-5-Nano	0.6972	0.4728	0.5429	0.3664	0.5090
	Avg. AUROC \uparrow	0.7985	0.6229	0.6492	0.6095	0.6309

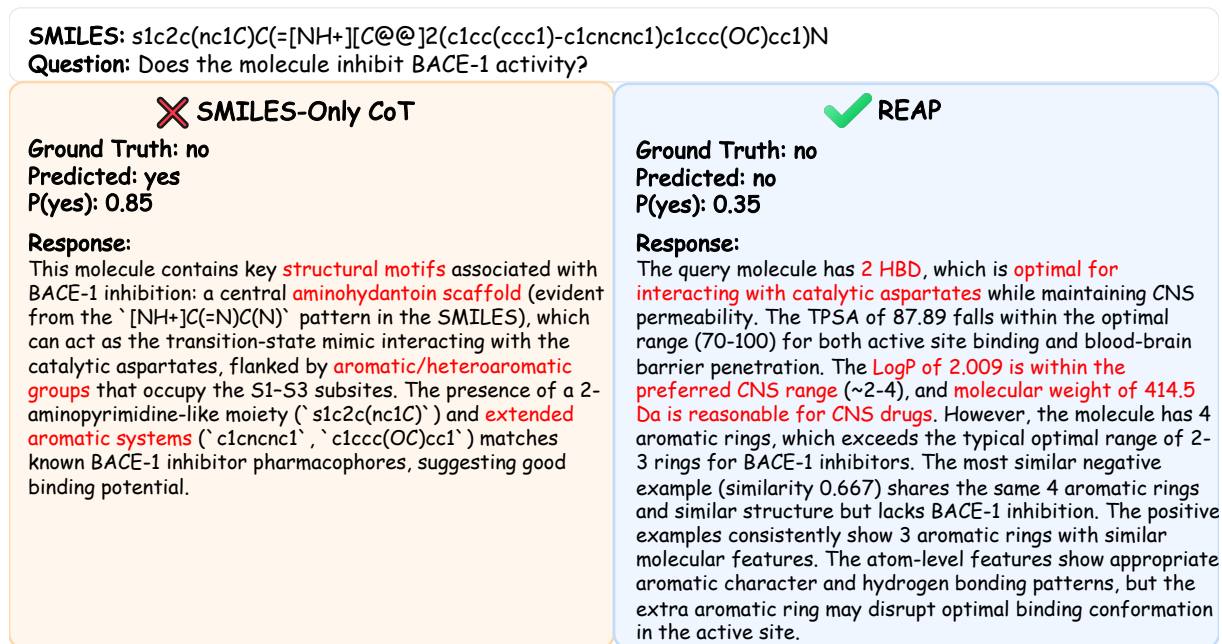


Figure 2: Comparison between SMILES-only CoT and REAP on a BACE-1 inhibition task. The left panel illustrates how the baseline falls into a hallucination trap, deriving an overconfident prediction from surface-level structural motifs without physical verification. In contrast, the right panel demonstrates REAP’s evidence-based reasoning: by explicitly citing quantitative priors (e.g., TPSA, LogP), it successfully rejects the candidate based on physicochemical constraints. Red highlights denote the verifiable attributes guiding the correct decision.

4.7 Inference Cost Analysis

As shown in Table 5, REAP requires only 1.01–1.02 average LLM calls per sample, which is comparable to the 1.00 calls used by all baselines. This is because Stage I (Feature Selection) involves three LLM rounds but is executed only once per

task, contributing only 3/ N calls per sample and thus becoming negligible in practice. Stage II (Retrieval) requires zero LLM calls, since it is implemented as pure CPU-based similarity search. Stage III (Inference) uses exactly one LLM call per sample, identical to the baselines. Therefore, the main

Method	Avg. LLM Calls	Avg. Input Tokens	Avg. Output Tokens	Avg. Total Tokens
DA	1.00	~147	~12	~159
CoT	1.00	~277	~139	~416
DA+FS	1.00	~198	~10	~208
CoT+FS	1.00	~334	~129	~463
MolRAG	1.00	~564	~600	~1164
REAP	1.01–1.02	~1400	~220	~1620

Table 5: Inference cost comparison across methods. REAP requires only 1.01–1.02 average LLM calls per sample, comparable to baselines (1.00). Its additional cost mainly comes from longer input context rather than more frequent model invocation.

Method	BACE	BBBP
MoMu	0.767	0.705
InstructMol	0.823	0.700
LLM4SD	0.838	0.756
MolRAG	0.767	0.629
REAP + Llama-2-7B	0.828	0.783

Table 6: Results on a smaller open-source backbone, Llama-2-7B. REAP remains effective even with a 7B model, outperforming the training-free baseline MolRAG and surpassing most training-based methods.

overhead of REAP comes not from more frequent LLM invocation, but from longer prompt context. As Table 5 shows, REAP uses more input tokens than simpler prompting strategies because it explicitly provides atom-level priors and retrieved analogue evidence, while keeping the number of LLM calls nearly unchanged.

4.8 Results on Smaller Open-Source LLMs

To examine whether REAP’s gains depend on large proprietary LLMs, we additionally evaluate it with Llama-2-7B, a much smaller open-source model. As shown in Table 6, REAP + Llama-2-7B achieves 0.828 AUROC on BACE and 0.783 AUROC on BBBP, substantially outperforming the training-free baseline MolRAG (0.767 / 0.629) and surpassing most training-based methods. These results suggest that the benefit of REAP mainly comes from its evidence-based inference framework rather than from relying on a particularly strong backbone.

5 Conclusion

In this work, we propose REAP, an effective reasoning framework that inference via explicit chemical priors. Instead of relying on the static parameters of training-based models, REAP externalizes knowledge into the first atom-level textual knowledge base. This architecture empowers LLMs to perform grounded reasoning by explicitly retrieving

fine-grained atomic evidence and molecule-level descriptors, ensuring both adaptability and interpretability. Empirically, REAP significantly outperforms existing reasoning framework and rivals state-of-the-art training-based methods across diverse benchmarks. By proving that explicit prior injection can effectively substitute for implicit parameter updates, REAP establishes a new path for transparent and robust AI in scientific discovery.

6 Limitations

Although REAP achieves strong performance on molecular property prediction and provides more interpretable predictions, several limitations remain. First, we evaluate REAP on property prediction benchmarks; extending it to more complex tasks such as molecule editing (Zhuang et al., 2025, 2026) is an important direction for future work. Second, although REAP exposes explicit intermediate evidence, the faithfulness and usefulness of such explanations still require further study (Ji et al., 2025). Finally, the current pipeline incurs a relatively large token budget, and more efficient evidence selection, constrained filtering, and compression strategies may further reduce inference cost (Zhang et al., 2026; Gong and Sun, 2025).

7 Acknowledgement

This work is partly supported by the National Natural Science Foundation of China (No. 62306255, 92370204), the National Key Research and Development Program of China (No. 2023YFF0725000), the Guangdong Basic and Applied Basic Research Foundation (No. 2024A1515011839), the Fundamental Research Project of Guangzhou (No. 2024A04J4233), the Guangzhou-HKUST(GZ) Joint Funding Program (No.2023A03J0008), and the Education Bureau of Guangzhou Municipality.

References

- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- He Cao, Zijing Liu, Xingyu Lu, Yuan Yao, and Yu Li. 2023. Instructmol: Multi-modal integration for building a versatile and reliable molecular assistant in drug discovery. *arXiv preprint arXiv:2311.16208*.
- Zhe Chen, Zhe Fang, Wenhao Tian, Zhaoguang Long, Changzhi Sun, Yuefeng Chen, Hao Yuan, Honglin Li, and Man Lan. 2025. Reactgpt: Understanding of chemical reactions via in-context tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 84–92.
- Shuting Cui, Ying Sun, Yuting Zhang, Qingxin Meng, and Hengshu Zhu. 2026. Llm-enhanced career knowledge graph understanding for job mobility prediction. *ACM Transactions on Management Information Systems*.
- John S Delaney. 2004. Esol: estimating aqueous solubility directly from molecular structure. *Journal of chemical information and computer sciences*, 44(3):1000–1005.
- Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the association for computational linguistics: ACL 2024*, pages 3563–3578.
- Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, and 1 others. 2024. A survey on in-context learning. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 1107–1128.
- Zheng Gong and Ying Sun. 2025. Outlier-aware post-training quantization for discrete graph diffusion models. In *Forty-second International Conference on Machine Learning*.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.
- Yang Ji, Ying Sun, Yuting Zhang, Zhigaoyuan Wang, Yuanxin Zhuang, Zheng Gong, Dazhong Shen, Chuan Qin, Hengshu Zhu, and Hui Xiong. 2025. A comprehensive survey on self-interpretable neural networks. *Proceedings of the IEEE*.
- Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992.
- Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, and 1 others. 2023. Pubchem 2023 update. *Nucleic acids research*, 51(D1):D1373–D1380.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.
- Seul Lee, Karsten Kreis, Srimukh Veccham, Meng Liu, Danny Reidenbach, Saeed Paliwal, Arash Vahdat, and Weili Nie. 2024. Molecule generation with fragment retrieval augmentation. *Advances in Neural Information Processing Systems*, 37:132463–132490.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.
- Shenghua Li and Yingwu Li. 2026. Severing the health-impairment chain: Psychological distress and cross-level safety climate associations with safety behavior. *Acta Psychologica*, 265:106773.
- Shengchao Liu, Weili Nie, Chengpeng Wang, Jiarui Lu, Zhuoran Qiao, Ling Liu, Jian Tang, Chaowei Xiao, and Animashree Anandkumar. 2023a. Multi-modal molecule structure–text model for text-based retrieval and editing. *Nature Machine Intelligence*, 5(12):1447–1457.
- Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. 2022. [Pre-training molecular graph representation with 3d geometry](#). In *International Conference on Learning Representations*.
- Shengchao Liu, Jiong Xiao Wang, Yijin Yang, Chengpeng Wang, Ling Liu, Hongyu Guo, and Chaowei Xiao. 2024. Conversational drug editing using retrieval and domain feedback. In *The twelfth international conference on learning representations*.
- Zejun Liu, Wei Zhang, Yingce Xia, Lijun Wu, Shufang Xie, Tao Qin, Ming Zhang, and Tie-Yan Liu. 2023b. Molxpt: Wrapping molecules with text for generative pre-training. *arXiv preprint arXiv:2305.10688*.
- Zhiyuan Liu, Sihang Li, Yan Chen Luo, Hao Fei, Yixin Cao, Kenji Kawaguchi, Xiang Wang, and Tat-Seng Chua. 2023c. Molca: Molecular graph-language modeling with cross-modal projector and uni-modal adapter. *arXiv preprint arXiv:2310.12798*.
- Jieyi Long. 2023. Large language model guided tree-of-thought. *arXiv preprint arXiv:2305.08291*.

- Yizhen Luo, Kai Yang, Massimo Hong, Xing Yi Liu, and Zaiqing Nie. 2023. Molfm: A multi-modal molecular foundation model. *arXiv preprint arXiv:2307.09484*.
- Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. 2024. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535.
- David L Mobley and J Peter Guthrie. 2014. Freesolv: a database of experimental and calculated hydration free energies, with input files. *Journal of computer-aided molecular design*, 28:711–720.
- Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Saucedo Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. 2017. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30.
- Murat Cihan Sorkun, Abhishek Khetan, and Süleyman Er. 2019. Aqsoldb, a curated reference set of aqueous solubility and 2d descriptors for a diverse set of compounds. *Scientific data*, 6(1):143.
- Bing Su, Dazhao Du, Zhao Yang, Yujie Zhou, Jiangmeng Li, Anyi Rao, Hao Sun, Zhiwu Lu, and Ji-Rong Wen. 2022. A molecular multimodal foundation model associating molecule graphs with natural language. *arXiv preprint arXiv:2209.05481*.
- Jike Wang, Rui Qin, Mingyang Wang, Meijing Fang, Yangyang Zhang, Yuchen Zhu, Qun Su, Qiaolin Gou, Chao Shen, Odin Zhang, and 1 others. 2025. Tokenmol 1.0: tokenized drug design with large language models. *Nature Communications*, 16(1):4416.
- Limei Wang, Kaveh Hassani, Si Zhang, Dongqi Fu, Baichuan Yuan, Weilin Cong, Zhigang Hua, Hao Wu, Ning Yao, and Bo Long. 2024. Learning graph quantized tokenizers for transformers. *arXiv preprint arXiv:2410.13798*.
- Ning-Ning Wang, Jie Dong, Yin-Hua Deng, Min-Feng Zhu, Ming Wen, Zhi-Jiang Yao, Ai-Ping Lu, Jian-Bing Wang, and Dong-Sheng Cao. 2016. Adme properties evaluation in drug discovery: prediction of caco-2 cell permeability using a combination of nsgaii and boosting. *Journal of chemical information and modeling*, 56(4):763–773.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. 2018. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530.
- Ziting Xian, Jiawei Gu, Lingbo Li, and Shangsong Liang. 2025. Molrag: unlocking the power of large language models for molecular property prediction. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15513–15531.
- Haoran Xin, Ying Sun, Chao Wang, and Hui Xiong. 2025. Llmcdsr: Enhancing cross-domain sequential recommendation with large language models. *ACM Transactions on Information Systems*, 43(5):1–33.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.
- Juzheng Zhang, Yatao Bian, Yongqiang Chen, and Quanming Yao. 2024. Unimot: Unified molecule-text language model with discrete token representation. *arXiv preprint arXiv:2408.00863*.
- Mingxu Zhang, Dazhong Shen, and Ying Sun. 2025a. Atomdisc: An atom-level tokenizer that boosts molecular llms and reveals structure–property associations. *Preprint*, arXiv:2512.03080.
- Mingxu Zhang, Huicheng Zhang, Jiaming Ji, Yaodong Yang, and Ying Sun. 2026. Enhance the safety in reinforcement learning by adrc lagrangian methods. *arXiv preprint arXiv:2601.18142*.
- Zimin Zhang, Qianli Wu, Botao Xia, Fang Sun, Ziniu Hu, Yizhou Sun, and Shichang Zhang. 2025b. Automated molecular concept generation and labeling with large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6918–6936, Abu Dhabi, UAE. Association for Computational Linguistics.
- Zihan Zhao, Da Ma, Lu Chen, Liangtai Sun, Zihao Li, Yi Xia, Bo Chen, Hongshen Xu, Zichen Zhu, Su Zhu, and 1 others. 2024. Chemdfm: a large language foundation model for chemistry. *arXiv preprint arXiv:2401.14818*.
- Yizhen Zheng, Huan Yee Koh, Jiaxin Ju, Anh TN Nguyen, Lauren T May, Geoffrey I Webb, and Shirui Pan. 2025. Large language models for scientific discovery in molecular property prediction. *Nature Machine Intelligence*, pages 1–11.
- Gengmo Zhou, Zhifeng Gao, Qiankun Ding, Hang Zheng, Hongteng Xu, Zhewei Wei, Linfeng Zhang, and Guolin Ke. 2023. Uni-mol: A universal 3d molecular representation learning framework.
- Hao Zhu, Todd M Martin, Lin Ye, Alexander Sedykh, Douglas M Young, and Alexander Tropsha. 2009. Quantitative structure- activity relationship modeling of rat acute toxicity by oral exposure. *Chemical research in toxicology*, 22(12):1913–1921.

Yuanxin Zhuang, Dazhong Shen, and Ying Sun. 2025. Moleditr1: Structure-preserving molecular editing via discrete diffusion and reinforcement learning. *arXiv preprint arXiv:2505.20131*.

Yuanxin Zhuang, Dazhong Shen, and Ying Sun. 2026. Ngtm: Substructure-based neural graph topic model for interpretable graph generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 29251–29259.

A Experimental Details

A.1 Task Descriptions

BBBP. This task assesses whether a compound can cross the blood-brain barrier (BBB), a crucial consideration for central nervous system (CNS) drugs. Molecules are classified into BBB-permeable or non-permeable categories based on experimental permeability data.

Tox21. A multi-label classification task involving 12 distinct toxicity pathways. These include nuclear receptor signaling and stress response pathways, reflecting the compound’s potential toxic effects across multiple biological processes.

ToxCast. This task covers a broader panel of in vitro toxicity assays than Tox21, measuring compound bioactivity across hundreds of biochemical and cellular targets. It enables high-throughput evaluation of toxicity at scale.

SIDER. The Side Effect Resource dataset includes marketed drugs annotated with known adverse drug reactions. The goal is to predict side effect profiles across 27 categories, making this a multi-label classification task.

ClinTox. This binary classification task distinguishes drugs approved by the FDA from those that failed clinical trials due to toxicity. It provides a stringent benchmark for modeling toxicity-related risk.

HIV. The HIV dataset involves binary classification of compounds based on their ability to inhibit HIV replication in a human cell assay. It evaluates antiviral activity and molecular efficacy.

BACE. This task aims to predict inhibitors of human β -secretase 1 (BACE-1), an enzyme linked to Alzheimer’s disease. Compounds are labeled based on their biochemical assay outcomes.

QM9. From the QM9 quantum chemistry dataset, we focus on predicting three scalar properties: the energies of the Highest Occupied Molecular Orbital (HOMO), the Lowest Unoccupied Molecular Orbital (LUMO), and their difference (the HOMO–LUMO gap). These properties are central to understanding molecular reactivity, stability, and electronic behavior.

ESOL. This task predicts the aqueous solubility of small molecules. ESOL provides experimental

log solubility values for a curated set of drug-like compounds.

FreeSolv. FreeSolv contains hydration free energy values for small neutral molecules. The task evaluates the model’s ability to predict solvation thermodynamics, crucial for drug absorption modeling.

Lipo. The Lipophilicity dataset measures the distribution coefficient ($\log D$) between octanol and water. It reflects how hydrophobic or hydrophilic a compound is, which is relevant to absorption and bioavailability.

Caco2. This task involves predicting the permeability of molecules across Caco-2 cell monolayers, a common in vitro assay used to estimate human intestinal absorption.

LD50. LD50 measures the lethal dose required to kill 50% of test subjects (typically rodents), serving as a proxy for acute toxicity prediction.

AqSol. AqSol contains experimentally measured solubility values for a broad set of compounds, aiming to improve solubility prediction beyond the chemical space covered by ESOL.

A.2 Setup for Filtering Atom Cards

We do *not* pass all atom cards to Stage III. For each molecule we keep only a small, decision-critical subset. Concretely, we first mark **must-keep** atoms: any RDKit-identified hydrogen-bond donor/acceptor (from the `ChemicalFeatures.BaseFeatures.fdef`), any atom with nonzero formal charge or $|\text{Gasteiger}| \geq 0.10$, all heteroatoms (N, O, S, P, F, Cl, Br, I), and ring bridgeheads/aromatic carbons. If must-keep atoms exceed a per-molecule budget M (default $M = 20$), we retain those with larger $|\text{Gasteiger}|$ and aromaticity first (stable ties by atom index). If they are fewer than M , we fill the remaining slots by ranking the rest with a light priority that favors aromatic/conjugated sites, ring/fused-ring membership, proximity to heteroatoms, and sp/sp² hybridization. The final selection is serialized as compact cards (token id, atom index, symbol, plus only the Stage I–selected fields), which focuses the prompt on functional centers instead of flooding the context with non-informative atoms.

A.3 Experimental Setup and Parameters

For all datasets, we follow the standard MoleculeNet evaluation protocol by holding out 10% of the data as the test set and *not* training any task-specific model; the training split is used exclusively as the retrieval pool in StageII. Following MoleculeNet(Wu et al., 2018), we use scaffold splits for the recommended tasks and random splits otherwise, with a fixed random seed of 42. Molecular similarity is computed using RDKit Morgan fingerprints and Tanimoto similarity, and we retrieve Top- k analogues (including both positive and negative examples) for each test molecule. We query LLMs with deterministic decoding (temperature =0) and a fixed maximum output length; all other decoding parameters are left at provider defaults. We evaluate classification tasks using AUROC and regression tasks using RMSE. Because logits are inaccessible for closed-source LLMs, we use the model-reported probability $P(\text{yes}) \in [0, 1]$ as the prediction score when computing AUROC. Full settings and hyperparameters are provided in Table 7.

Component	Hyperparameter	Value / Setting
Data split	Train:test ratio	90:10
	Random seed	42
Retrieval (Stage II)	Fingerprint type	RDKit Morgan (ECFP4)
	Radius	2
	nBits	2048
	Similarity	Tanimoto
	Top- k analogues	$k = 5$ or $k = 10$
LLM decoding	Temperature	0.0
	Max tokens	2000
	Top- p	Default
	Other decoding	Default
Metrics	Classification	AUROC
	Regression	RMSE

Table 7: Summary of experimental hyperparameters and implementation choices used in REAP.

A.4 Baselines

To comprehensively evaluate our REAP framework, we compare against both *training-free* and *training-based* baselines. For training-based methods, we further include *LLM-based*, *graph-based*, and *3D-based* approaches. All baseline results are reported as in their original papers.

Training-free methods. We include MolRAG (Xian et al., 2025) and LLM4SD (Zheng et al., 2025). MolRAG computes the similarity between a test molecule and the training set, then supplies the similarity scores and the correspond-

ing training labels to LLMs as in-context evidence, allowing the model to make decisions by analogy. LLM4SD prompts LLMs to induce classification rules; RDKit-derived, molecule-level chemical priors are converted into vector features, upon which a decision tree is trained for prediction.

LLM-based training methods. We consider InstructMol (Cao et al., 2023), TokenMol (Wang et al., 2025), and MoMu (Su et al., 2022). InstructMol trains an additional adapter that projects molecular representations into the LLM embedding space; the projected vector is concatenated with the original prompt and injected into the transformer, so the LLM learns to leverage these features for decision making. MoMu performs contrastive learning between text embeddings and molecular embeddings to learn a unified cross-modal representation. TokenMol discretizes 3D features—specifically torsion angles—into tokens, concatenates them with the prompt to form a token-only sequence, and trains the model via random-mask autoregression.

Graph-based methods. We adopt GraphMVP (Liu et al., 2022), which conducts contrastive learning between 2D (GIN-based) and 3D (SchNet-based (Schütt et al., 2017)) molecular representations to capture complementary structural information.

3D-based methods. We include UniMol (Zhou et al., 2023), which pretrains on molecular conformations and protein pockets, leveraging rotation-translation-invariant spatial positional encodings and achieving state-of-the-art performance on molecular property prediction.

A.5 knowledge base Construction Details

We construct the atomic-prior knowledge base from a subset of approximately 220,000 molecules sampled from PubChem, using only their SMILES strings. Specifically, we encode each molecule with froze MoleculeSTM checkpoint to obtain atom-level representations, and then map these representations to discrete token indices via AtomDisc. Importantly, this entire pipeline is performed in a fully unsupervised manner: we do not use any labels or annotations from downstream property-prediction tasks, nor do we tune any component on evaluation datasets. So we don’t need to train any models in our framework. For all tasks, we only use off-the-shelf models’ checkpoint to obtain the necessary information and textualize them using

rdkit and our methods.

To further avoid any potential data leakage, the knowledge base is built solely from the PubChem subset and is kept completely disjoint from all MoleculeNet evaluation datasets; no molecules from the test/validation splits are used during knowledge base construction. Atom-level priors are computed purely from molecular structures using RDKit-based descriptors and statistics, without referencing any task targets.

For reproducibility and community use, we publicly release the complete REAP framework, including the knowledge-base construction code and prompting scripts, at <https://github.com/mingxuZhang2/REAP>.

A.6 Evaluation Details

This section describes our evaluation protocol for classification and regression tasks.

Classification. For closed-source LLMs, token-level logits are not accessible. Therefore, instead of computing AUROC from logits, we prompt the model to output a calibrated confidence score for the positive class, i.e., $p(\text{yes}) \in [0, 1]$, in a structured format. Concretely, the model returns a numeric value wrapped by `<confidence>` and `</confidence>`, and we parse this value as the prediction score to compute AUROC. For multi-label (or multi-class) settings, we decompose each task into a set of binary one-vs-rest (or per-label) sub-tasks, and report AUROC for each label/item accordingly.

Regression. For regression tasks, we compute RMSE directly between the model predictions and the dataset-provided ground-truth labels. We do not apply any additional target transformations (e.g., log-space conversion) or target standardization/normalization. Since our approach is training-free, we evaluate in the original label space as provided by each benchmark dataset, and compute RMSE on this same scale. To avoid ambiguity, we will also explicitly state (per dataset) the target definition and units (when available) in the dataset description or appendix.

B Atomic Prior knowledge base field

Atomic Prior KB Schema — Fields & Example

Schema:

- token_id: discrete AtomDisc token index
- support_count: #atoms in corpus mapped to this token
- primary_symbol: most frequent element symbol among instances
- is_mixed: whether multiple element types map to this token
- symbol_distribution: per-element counts for this token
- mixture_entropy: entropy (01) of the element mixture
- env_type: modal local environment {chain|ring|fused_ring}
- env_distribution: counts of {chain, ring, fused_ring}
- aromatic_ratio: fraction of instances that are aromatic (01)
- conjugated_ratio: fraction with any conjugated bond (01)
- median_degree: median number of bonded neighbors
- median_ring_size: median size of the smallest ring (0 if non-ring)
- hybridization: modal hybridization {s, sp, sp2, sp3, sp3d, sp3d2, other}
- electrics.inductive: median inductive effect sign in {-1,0,+1}
- electrics.resonance: median resonance effect sign in {-1,0,+1}
- polarity.gasteiger_q50: median Gasteiger partial charge
- polarity.gasteiger_iqr: IQR of Gasteiger charge (variability)
- polarity.tpsa_contrib_q50: median per-atom TPSA proxy
- hbond.donor_ratio: fraction flagged as H-bond donors (01)
- hbond.acceptor_ratio: fraction flagged as H-bond acceptors (01)
- hetero_r1_median: median #heteroatom neighbors (1-hop)
- neighbors_top: top neighbor tokens by PMI; each item has:
 - {token: neighbor token id, pmi: $\log_2 P(A,B)/(P(A)P(B))$,
 - co_occur_ratio: fraction of this tokens atoms with such a neighbor}

EXAMPLE of Atomic Priors knowledge base:

```
{
  "token_id": 112,
  "support_count": 49708,
  "primary_symbol": "C",
  "is_mixed": true,
  "symbol_distribution": {"C": 26016, "O": 23689, "Hg": 3},
  "mixture_entropy": 0.9992911271,
  "env_type": "chain",
  "env_distribution": {"chain": 49689, "ring": 19},
  "aromatic_ratio": 0.0,
  "conjugated_ratio": 0.0001005874,
  "median_degree": 2.0,
  "median_ring_size": 6.0,
  "hybridization": "sp3",
  "electrics": {"inductive": 0, "resonance": 0},
  "polarity": {
    "gasteiger_q50": 0.0339449055,
    "gasteiger_iqr": 0.4922527119,
    "tpsa_contrib_q50": 0.0
  },
  "hbond": {"donor_ratio": 0.0014082240, "acceptor_ratio": 0.4765631287},
  "hetero_r1_median": 0.0,
  "neighbors_top": [
    {"token": 30, "pmi": 6.3621896439, "co_occur_ratio": 0.9395067192},
    {"token": 53, "pmi": 3.2824712421, "co_occur_ratio": 0.1208256216},
    {"token": 112, "pmi": 1.9914829505, "co_occur_ratio": 0.0966444033},
    {"token": 58, "pmi": 1.3526284184, "co_occur_ratio": 0.0445401143},
    {"token": 430, "pmi": 0.5000017521, "co_occur_ratio": 0.0427898930}
  ]
}
```

C Input Format Example for Retrieved Analogues

Input Retrieved Analogues Format

SMILES: Fc1cc(cc(F)c1)CC(NC(=O)C)C(O)C[NH2+]C1(CCCCC1)c1cc(ccc1)C1CCOC1

Discrete Token Sequence: A410 A188 A434 A434 A434 A188 A410 A434 A139 A17 A425 A505 A263 A56 A100 A332 A100 A296 A296 A389 A389 A215 A389 A389 A424 A107 A503 A107 A107 A107 A146 A146 A229 A229 A146

Similarity: 0.928

Ground Truth: yes

Selected Atom-Level Features:

[A410, Atom#0, C]: gasteiger_q50=-0.059, gasteiger_iqr=0.145, hba_ratio=0.479, hbd_ratio=0.006, aromatic_ratio=0.515, conjugated_ratio=0.517, ring_ratio=0.000, median_degree=2.000, neighbors_top=[{'token': 124, 'pmi': 6.908255240747004, 'co_occur_ratio': 0.2788412820795242}, {'token': 188, 'pmi': 6.555685343667383, 'co_occur_ratio': 0.5088445864081947}, {'token': 109, 'pmi': 4.684120859168923, 'co_occur_ratio': 0.13898006388368764}, {'token': 410, 'pmi': 4.239188136752083, 'co_occur_ratio': 0.4191651062892389}, {'token': 484, 'pmi': -1.0447276102119363, 'co_occur_ratio': 0.08201343760326027}]

...

Selected Molecule-Level Features:

{'TPSA': 75.17, 'LogP': 3.299, 'MolWt': 487.611, 'HBA': 3, 'HBD': 3, 'NumAromaticRings': 2, 'NumRotatableBonds': 9, 'NumHeteroatoms': 7, 'FormalCharge': 1}

D Input Format for Tested Molecules

Input Tested Molecule Format

SMILES: Fc1cc(cc(F)c1)CC(NC(=O)C)C(O)C[NH2+]C1(CCCCC1)c1cc(ccc1)C1CCOCOC1

Discrete Token Sequence: A410 A188 A434 A434 A434 A188 A410 A434 A139 A17 A425 A505 A263 A56 A100 A332 A100 A296 A296 A389 A389 A215 A389 A389 A424 A107 A503 A107 A107 A107 A146 A229 A229 A229 A312 A312 A229

Selected Atom-Level Features:

[A410, Atom#0, C]: gasteiger_q50=-0.059, gasteiger_iqr=0.145, hba_ratio=0.479, hbd_ratio=0.006, aromatic_ratio=0.515, conjugated_ratio=0.517, ring_ratio=0.000, median_degree=2.000, neighbors_top=[{'token': 124, 'pmi': 6.908255240747004, 'co_occur_ratio': 0.2788412820795242}, {'token': 188, 'pmi': 6.555685343667383, 'co_occur_ratio': 0.5088445864081947}, {'token': 109, 'pmi': 4.684120859168923, 'co_occur_ratio': 0.13898006388368764}, {'token': 410, 'pmi': 4.239188136752083, 'co_occur_ratio': 0.4191651062892389}, {'token': 484, 'pmi': -1.0447276102119363, 'co_occur_ratio': 0.08201343760326027}]

Selected Molecule-Level Features:

{'TPSA': 84.400, 'LogP': 3.274, 'MolWt': 517.637, 'HBA': 4.000, 'HBD': 3.000, 'NumAromaticRings': 2.000, 'NumRotatableBonds': 9.000, 'NumHeteroatoms': 8.000, 'FormalCharge': 1.000}

E Prompt Templates

E.1 Prompt Template for Stage I

Stage I — Round 1: Prior Schema Analysis

System:

You are a chemistry and molecular property prediction expert.
Your task is to understand the available features for molecular analysis.

User:

I will provide a list of features. For each, briefly state:
1) what it measures, 2) which property it relates to, 3) when it matters.

Atom-Level (AtomDisc KB): support_count, primary_symbol, is_mixed, mixture_entropy, gasteiger_q50, gasteiger_iqr, hba_ratio, hbd_ratio, aromatic_ratio, conjugated_ratio, ring_ratio, median_degree, neighbors_top (PMI)

Molecule-Level (RDKit): TPSA, LogP, MolWt, HBA, HBD, NumAromaticRings, NumRotatableBonds, NumHeteroatoms, FormalCharge

Stage I — Round 2: Task Understanding

System:

Continue as a chemistry expert. Focus on the task.

User:

Task:{instruction}

Please analyze:

- 1) the chemical/biological significance;
- 2) key molecular characteristics for this task;
- 3) structural cues separating positive vs. negative samples.

Stage I — Round 3: Feature Selection (JSON Output)

System:

Select the most relevant features and justify them concisely.

User:

Task:{instruction}

Output JSON:

```
{
  "atom_features": ["feature1", "feature2", ...],
  "molecule_features": ["feature1", "feature2", ...],
  "feature_descriptions": {
    "feature1": "1-2 sentences on meaning/relevance",
    "feature2": "...",
    "...": "..."
  }
}
```

IMPORTANT:

- 1) Use exact names from the lists;
- 2) Provide descriptions for ALL selected features;
- 3) Keep descriptions concise (12 sentences).

E.2 Prompt Template for Stage III

Stage III: Inference With Analogues and Chemical Priors

System:

You are a molecular property prediction expert.

Your task is to predict whether a molecule satisfies a given property based on:

1. Selected relevant features
2. Instance-specific analysis from similar training examples

You must provide:

1. A binary prediction (yes/no)
2. A confidence score P(yes) from 0 to 100

User:

Task:{instruction}

Feature Descriptions:{feature_descriptions}

SMILES:{smiles}

Discrete Token Sequence:{query_tokens}

Selected Atom-Level Features:{atom_features}

Selected Molecule-Level Features:{molecule_features}

Instance-Specific Analysis:{similar_analysis}

Based on all the information above, please predict whether this molecule satisfies the task.

Your response MUST follow this exact format:

<analysis>

Brief reasoning based on features and similar examples

</analysis>

<answer>yes**</answer>** or **<answer>**no**</answer>**

<confidence>XX**</confidence>**

Where:

- **<answer>**MUST be exactly "yes" or "no" (lowercase)
- **<confidence>**is P(yes) from 0-100 (integer)
- If you answer "yes", confidence should be > 50
- If you answer "no", confidence should be < 50

CRITICAL: You MUST include all three tags (**<analysis>**, **<answer>**, **<confidence>**) in your response.

E.3 Baseline Prompt

Directly Answer without Analogues

System:

You are an expert medicinal chemist. Directly answer the question with "yes" or "no".

User:

Task:{TASK_NAME}

SMILES:{SMILES}

Your response MUST follow this exact format:

`<answer>yes</answer>` or `<answer>no</answer>`
`<confidence>XX</confidence>`

Where:

- `<answer>`MUST be exactly "yes" or "no" (lowercase)
- `<confidence>`is P(yes) from 0-100 (integer)
- If you answer "yes", confidence should be > 50
- If you answer "no", confidence should be < 50

CRITICAL: You MUST include all three tags (`<answer>`, `<confidence>`) in your response.

Answer:

CoT without Analogues

System:

You are an expert medicinal chemist. Your task is to analyze a molecule given its SMILES representation and predict whether it satisfies the given condition.

User:

Task:{TASK_NAME}

Instructions:

1. Carefully analyze the SMILES structure
2. Consider molecular properties, functional groups, and structural features
3. Provide clear reasoning based on chemical principles
4. Give your final answer as "yes" or "no"

Format your response as:

- First, provide your analysis and reasoning (2-3 sentences)
- Then, provide your final answer on a new line starting with "Answer: " followed by "yes" or "no"

Question: {QUESTION}

SMILES:{SMILES}

Your response MUST follow this exact format:

`<analysis>`

Brief reasoning based on features and similar examples

`</analysis>`

`<answer>yes</answer>` or `<answer>no</answer>`

`<confidence>XX</confidence>`

Where:

- `<answer>`MUST be exactly "yes" or "no" (lowercase)
- `<confidence>`is P(yes) from 0-100 (integer)
- If you answer "yes", confidence should be > 50
- If you answer "no", confidence should be < 50

CRITICAL: You MUST include all three tags (`<analysis>`, `<answer>`, `<confidence>`) in your response.

Your analysis and answer:

F Output Example

F.1 Stage I Output Example

F.1.1 Stage I-Round 1 Example

Stage I — Round 3: Feature Selection Output

```
{
  "atom_features": [
    "primary_symbol", "gasteiger_q50", "gasteiger_iqr", "hba_ratio", "hbd_ratio", "aromatic_ratio",
    "conjugated_ratio", "ring_ratio", "median_degree", "neighbors_top"
  ],
  "molecule_features": [
    "TPSA", "LogP", "MolWt", "HBA", "HBD", "NumAromaticRings", "NumRotatableBonds", "
    NumHeteroatoms", "FormalCharge"
  ],
  "feature_descriptions": {
    "primary_symbol": "Element identity (e.g., N, O) indicates potential basic centers and H-
      bonding atoms needed to engage BACE-1s catalytic Asp residues.",
    "gasteiger_q50": "Median partial charge reflects local polarity/basicity, informing ionic and
      H-bond interactions with the Asp dyad.",
    "gasteiger_iqr": "Charge variability signals context sensitivity of electronics, affecting
      strength and orientation of binding in the active site.",
    "hba_ratio": "Fraction acting as H-bond acceptors; supports complementary interactions in
      BACEs polar pocket and conserved waters.",
    "hbd_ratio": "Fraction acting as H-bond donors; enables directional H-bonds but excess donors
      can reduce CNS permeability.",
    "aromatic_ratio": "Proportion of aromatic character; contributes -interactions and hydrophobic
      contacts in BACE subsites while influencing permeability.",
    "conjugated_ratio": "Fraction in conjugated systems; affects electron delocalization and
      polarizability, tuning binding and lipophilicity.",
    "ring_ratio": "Fraction of atoms in rings; increases rigidity to align pharmacophore and
      reduce binding entropy costs.",
    "median_degree": "Typical number of bonded neighbors; shapes steric bulk and substitution
      pattern for pocket fit.",
    "neighbors_top": "Most common neighboring atoms indicate functional group context that drives
      specific binding interactions.",
    "TPSA": "Topological Polar Surface Area; moderate values balance BBB penetration with
      sufficient polarity for binding.",
    "LogP": "Lipophilicity; needs to be tuned for CNS permeability while avoiding nonspecific
      hydrophobic binding.",
    "MolWt": "Molecular weight; size influences BBB penetration and the ability to span BACE
      subsites without being overly large.",
    "HBA": "Count of hydrogen bond acceptors; facilitates binding but high counts raise polarity
      and desolvation.",
    "HBD": "Count of hydrogen bond donors; enables key H-bonds yet excessive donors hinder
      permeability.",
    "NumAromaticRings": "Number of aromatic rings; provides /hydrophobic contacts and shape,
      though too many can hurt solubility/permeability.",
    "NumRotatableBonds": "Molecular flexibility; fewer rotatable bonds improve binding efficiency
      and CNS drug-likeness.",
    "NumHeteroatoms": "Total non-C/H atoms; correlates with polarity and specific interactions,
      with high counts potentially impairing BBB crossing.",
    "FormalCharge": "Net formal charge; 0 to +1 is typically favored for BBB penetration and ionic
      interaction with catalytic Asp residues."
  }
}
```

F.2 Classification Task Output Example

Example Output for Classification Task – BACE

SMILES: "Fc1cc(cc(F)c1)CC(NC(=O)C)C(O)C[NH2+]C1(CCC(=O)NC1)c1cc(ccc1)C(C)(C)C"

Ground Truth:yes

Response:

<analysis>The query molecule has very similar structural features to the top positive examples, particularly the first positive example which shares nearly identical molecular features (HBD =4, TPSA=95.04, LogP=2.04, MolWt=488.60, NumAromaticRings=2). The key functional atoms including the amide carbonyl (A263), hydroxyl group (A332), and ammonium group (A397) provide essential hydrogen bonding interactions with BACE-1 catalytic aspartates. The aromatic rings (A410, A188, A434 patterns) enable π -stacking with Tyr71. The TPSA (95.04) falls within the optimal range for CNS penetration, and LogP (2.04) is in the preferred range for brain exposure. The most similar negative example differs by having a ketone instead of an amide in the ring system, suggesting the amide group in the query molecule provides better binding interactions. The high structural similarity to multiple positive examples and optimal CNS drug-like properties strongly support BACE-1 inhibition. **</analysis>**

<answer>yes**</answer>**

<confidence>85**</confidence>**

Example Output for Classification Task – BBBP

SMILES: "C3=C(C2(C1=CC=CC=C1)C(NCN2)=O)C=CC=C3"

Ground Truth:yes

Response:

<analysis>The query molecule is small (MolWt 238) and neutral, with moderate lipophilicity (LogP 1.61) and low polarity (TPSA 41.1). TPSA < 60 is typically favorable for BBB penetration. Although HBD = 2 can be somewhat limiting, several top positive neighbors have similar HBD (2) and comparable LogP/TPSA ranges (e.g., TPSA ~58, LogP ~0.5\to 1.77), and share key functional motifs (amide carbonyl [A482] with adjacent nitrogens [A154/A151], and aromatic rings). The nearest positive examples have notably higher similarity (0.36\to 0.50) than the negatives (0.21), supporting a positive prediction. Overall, the feature profile and neighbor evidence favor BBB penetration. **</analysis>**

<answer>yes**</answer>**

<confidence>74**</confidence>**

Example Output for Classification Task – Clintox

SMILES: "CC1=C(C(C(=C(N1)C)C(=O)OCC[NH+](C)Cc2ccccc2)c3cccc(c3)[N+](=O)[O-])C(=O)OC",

Ground Truth:no

Response:

<analysis> The query shares the same nitro\to aromatic core and ester functionalities as multiple highly similar NEGATIVE examples (similarities 0.621, 0.446, 0.444), with very close atom-level features (A113/A20 aromatic carbons, A340/A302 oxygens, nitro group A469/A200). Its molecule-level descriptors (TPSA 112, LogP 2.26, HBA 7, HBD 2, MW 480) align closely with these negatives (TPSA ~108\to 117, LogP ~2.18\to 2.20, HBA 7\to 138). The POSITIVE set is less similar (max ~0.205) and shows more varied scaffolds and properties. Although the query\to formal positive charge (1) may modulate some properties, the nearest-neighbor pattern strongly favors the negative class. **</analysis>**

<answer>no**</answer>**

<confidence>34**</confidence>**

Example Output for Classification Task – HIV

SMILES: "Cc1cn(C(CO)OC(CO)CN(O)C2C3CC4CC(C3)CC2C4)c(=O)[nH]c1=O"

Ground Truth:no

Response:

<analysis>The query molecule shares a highly similar core structure with both positive and negative examples, featuring the characteristic uracil-like ring system (A457, A457, A154, A416, A412, A482 pattern) that appears in anti-HIV compounds. However, several concerning features suggest it may not exhibit anti-HIV activity: 1. Molecular weight (409.483) is significantly higher than most positive examples (224-314 range) and approaches the upper limit for drug-like molecules, potentially hindering binding pocket accommodation. 2. High rotatable bond count (8) indicates excessive flexibility, which typically reduces binding affinity to rigid enzyme pockets - this is higher than most positive examples (2-6 range). 3. The complex bicyclic substituent introduces substantial steric bulk that may not fit well into known anti-HIV enzyme binding sites. 4. While TPSA (128.02) and HBA/HBD counts are reasonable for anti-HIV agents, the most similar negative example (similarity 0.529) shares nearly identical molecular features (TPSA=128.02, HBA=8, HBD=4) and also contains the problematic high rotatable bond count. The structural similarity to negative examples, particularly the high molecular weight and flexibility, outweighs the shared core with positive compounds.**</analysis>**

<answer>no**</answer>**

<confidence>35**</confidence>**

Example Output for Classification Task – Tox21

SMILES: "COc1ccc2c3c1O[C@@H]1[C@@H](O)C=C[C@@H]4[C@@H](C2)NCC[C@]314"

Ground Truth:no

Response:

<analysis>The query molecule has key features that align with known ER-LBD activators: HBD=2 (including a phenolic -OH group at A423 with hbd_ratio=0.998), HBA=4, moderate LogP=1.16, low TPSA=50.72, and rigid structure (NumRotatableBonds=1). However, it only has one aromatic ring (NumAromaticRings=1), which is suboptimal compared to typical ER agonists that have 2-4 aromatic rings for optimal hydrophobic contact. The most similar negative example (similarity =0.448) shares the same steroid-like scaffold with only one aromatic ring and was inactive, while positive examples with higher similarity tend to have more aromatic character. The presence of the essential phenolic -OH group is favorable, but the limited aromatic system likely reduces binding efficiency to the hydrophobic ER pocket.**</analysis>**

<answer>no**</answer>**

<confidence>35**</confidence>**",

Example Output for Classification Task – Sider

SMILES: "CC1=C(C2=CC3=NC(=CC4=C(C(=C(N4)C=C5[C@@]6([C@@H](C(=CC=C6C(=N5)C=C1N2)C(=O)OC)C(=O)OC)C)C)CC(=O)OC)C(=C3C)CCC(=O)O)C=C"

Ground Truth:yes

Response:

<analysis>The query molecule shows several concerning features for carcinogenicity. It has a high aromatic ratio (0.595-0.939 across atoms) and extensive conjugation (0.951-1.000), which enables DNA intercalation. The presence of 3 aromatic rings and polycyclic aromatic structure is a classic structural alert for carcinogenicity. The high LogP (6.719) indicates significant lipophilicity for bioaccumulation, while the molecular weight (718.807) suggests potential bioavailability issues, though the molecule may still reach nuclear targets. The similar positive examples share key structural motifs - particularly the polycyclic aromatic systems with nitrogen heteroatoms and carboxylic acid/carboxylate groups. The top positive example has very similar atom-level features (same token patterns A113, A160, A162) and comparable molecular properties (TPSA 152.19 vs 173.56, similar aromatic ring count). The negative examples tend to have simpler aromatic systems, lower molecular complexity, and different functional group patterns. The combination of DNA-intercalating aromatic systems, metabolic activation potential from heteroatoms, and structural similarity to known carcinogens supports a positive prediction.**</analysis>**

<answer>yes**</answer>**

<confidence>78**</confidence>**

G Additional Experimental Results

Table 8: Regression benchmarks (RMSE ↓) across six datasets (Aqsl, Caco2, ESOL, FreeSolv, LD50, Lipophilicity). Entries marked “—” were not reported. The *Average* column is computed per method over the available datasets. The best results are shown in **bold**, and the runner-up results are underlined.

Method	Aqsl	Caco2	ESOL	FreeSolv	LD50	Lipophilicity	Average
CoT	2.8430	13.7336	0.8944	3.4312	737.9134	1.2098	126.6709
CoT+FS	1.3486	8.6642	0.9134	3.2863	12.7961	1.2479	4.7094
DA	1.4254	17.9927	1.0194	2.4534	297.4678	1.3517	53.6184
DA+FS	1.4581	0.6335	1.0497	2.5510	1.1385	1.4555	1.3810
MolRAG	—	—	3.2806	6.0827	—	1.1251	—
LLM4SD	—	—	0.5200	2.6200	—	0.6800	—
TokenMol	0.7350	0.6110	0.5930	<u>1.2250</u>	0.9400	0.6450	<u>0.7915</u>
MoleculeSTM	1.3293	<u>0.5409</u>	1.1770	1.2880	<u>0.8047</u>	0.6944	0.9724
REAP	<u>1.1266</u>	0.5041	<u>0.6504</u>	1.0177	0.7095	0.7143	0.7871

Table 9: Regression benchmarks (RMSE ↓). Each dataset is shown as a multirow block with per-LLM results. The bottom row reports the average across all regression datasets and LLMs.

Dataset	Model	REAP	DA+FS	CoT+FS	DA	CoT
ESOL	Claude-Sonnet-4	0.5854	0.9414	0.9043	0.8589	0.9654
	DeepSeek-V3.2	0.6504	1.0497	0.9134	1.0194	0.8944
	GPT-4.1	0.6940	1.1039	0.9846	1.0487	0.9363
	GPT-5	0.5900	0.8114	0.8529	0.8833	0.8668
	GPT-5-Nano	0.7729	1.4503	1.4259	1.3622	1.3085
FreeSolv	Claude-Sonnet-4	0.9366	2.5170	2.9579	3.3703	3.3429
	DeepSeek-V3.2	1.0177	2.5510	3.2863	2.4534	3.4312
	GPT-4.1	1.0364	5.2186	3.9521	8.9247	11.4176
	GPT-5	0.9930	2.6808	3.1213	4.2696	4.0424
	GPT-5-Nano	1.0009	3.0818	2.9089	5.8587	6.8408
Lipophilicity	Claude-Sonnet-4	0.7107	1.2095	1.1821	1.2493	1.3552
	DeepSeek-V3.2	0.7143	1.4555	1.2479	1.3517	1.2098
	GPT-4.1	0.7338	1.2383	1.2615	1.2134	1.2376
	GPT-5	0.6227	1.0466	1.0114	1.0542	1.0792
	GPT-5-Nano	0.7051	1.4405	1.3494	1.3876	1.3302
Avg. RMSE ↓		0.7842	1.8531	1.8240	2.4204	2.6839

Table 10: Regression benchmarks (RMSE ↓) across six datasets (Aqsol, Caco2, ESOL, FreeSolv, LD50, Lipophilicity). The *Average* column is computed per method over all six datasets. To demonstrate the effectiveness of the reasoning capability of LLMs, we further compare the KNN method with our REAP on regression tasks. As shown in Table 10, our REAP greatly outperform than the KNN or weighted-KNN methods. Importantly, these KNN baselines use exactly the same retrieval setup as REAP (the same similarity metric and the same top- k neighbors), and differ only in the prediction head: KNN directly averages the retrieved labels, while weighted-KNN performs similarity-weighted averaging. Therefore, the substantial gap between REAP and the two nearest-neighbor baselines indicates that the improvements cannot be explained by nearest-neighbor signals alone (i.e., simply copying/aggregating retrieved labels or priors). Instead, the LLM effectively integrates the retrieved evidence with task instructions and chemical context to perform non-trivial reasoning beyond local interpolation, which leads to consistently better performance across all benchmarks.

Method	Aqsol	Caco2	ESOL	FreeSolv	LD50	Lipophilicity	Average
KNN	1.6771	0.5779	1.5175	10.1398	0.8169	0.9199	2.6082
Weighted-KNN	1.6580	0.5623	1.5218	9.6172	0.8054	0.9067	2.5119
REAP	1.1266	0.5041	0.6504	1.0177	0.7095	0.7143	0.7871

Table 11: Classification benchmarks (AUROC \uparrow). Each dataset is shown as a multirow block with per-LLM results. The bottom row reports the average across all classification datasets and LLMs.

Dataset	Model	REAP	DA+FS	CoT+FS	DA	CoT
BACE	Claude-Sonnet-4	0.8028	0.6218	0.6694	0.6891	0.6390
	DeepSeek-V3.2	0.8042	0.6494	0.6373	0.6428	0.6775
	GPT-4.1	0.8443	0.7087	0.7608	0.7330	0.7012
	GPT-5	0.8970	0.6941	0.7106	0.7484	0.6975
	GPT-5-Nano	0.6729	0.5709	0.6773	0.5984	0.6421
BBBP	Claude-Sonnet-4	0.8173	0.7604	0.7730	0.7315	0.7686
	DeepSeek-V3.2	0.8246	0.5510	0.6660	0.4918	0.6121
	GPT-4.1	0.8074	0.7335	0.7584	0.7304	0.7570
	GPT-5	0.8731	0.8856	0.8340	0.8685	0.8435
	GPT-5-Nano	0.7259	0.4175	0.4439	0.3435	0.4081
ClinTox	Claude-Sonnet-4	0.7614	0.4030	0.5085	0.4870	0.5353
	DeepSeek-V3.2	0.8163	0.7792	0.6283	0.8459	0.8902
	GPT-4.1	0.7514	0.5584	0.5589	0.3223	0.2887
	GPT-5	0.8856	0.5799	0.6033	0.7524	0.5258
	GPT-5-Nano	0.6927	0.4301	0.5075	0.1574	0.4769
Avg. AUROC \uparrow		0.7985	0.6229	0.6492	0.6095	0.6309