

Automated Molecular Concept Generation and Labeling with Large Language Models

Zimin Zhang^{1*†}, Qianli Wu^{2*†}, Botao Xia^{2*†},
Fang Sun², Ziniu Hu³, Yizhou Sun², Shichang Zhang^{4†},

¹University of Illinois Urbana-Champaign, ²University of California Los Angeles

³California Institute of Technology, ⁴Harvard University

¹ziminz19@illinois.edu

²{qianliwu, xiabotao}@g.ucla.edu, {fts, yzsun}@cs.ucla.edu

³acgbull@gmail.com, ⁴shzhang@hbs.edu

Abstract

Artificial intelligence (AI) is transforming scientific research, with explainable AI methods like concept-based models (CMs) showing promise for new discoveries. However, in molecular science, CMs are less common than black-box models like Graph Neural Networks (GNNs), due to their need for predefined concepts and manual labeling. This paper introduces the **Automated Molecular Concept (AutoMolCo)** framework, which leverages Large Language Models (LLMs) to automatically generate and label predictive molecular concepts. Through iterative concept refinement, AutoMolCo enables simple linear models to outperform GNNs and LLM in-context learning on several benchmarks. The framework operates without human knowledge input, overcoming limitations of existing CMs while maintaining explainability and allowing easy intervention. Experiments on MoleculeNet and High-Throughput Experimentation (HTE) datasets demonstrate that AutoMolCo-induced explainable CMs are beneficial for molecular science research. The source code is available at <https://github.com/ziminz19/AutoMolCo>.

1 Introduction

Artificial intelligence (AI) has significantly advanced molecular science. A prime example is MIT Jameel Clinic’s use of deep learning to identify halicin – the first antibiotic discovered in three decades that is effective against a broad spectrum of 35 bacteria (Stokes et al., 2020). Deep learning models, such as Graph Neural Networks (GNNs), excel at learning complex atomic structures and predicting molecular properties (Wu et al., 2018). However, a major challenge with such deep-learning-based models like GNNs is their “black

boxes” nature and lack of explainability (Yuan et al., 2022). Despite their high predictive performance, black-box models fail to provide insights into the underlying reasoning behind their predictions, making it difficult for scientists to interpret and intervene in the model’s decision-making process, which hinders scientific understanding and limits the potential for knowledge discovery.

In contrast, concept-based models (CMs) (Lampert et al., 2009; Koh et al., 2020; Yeh et al., 2020; Wu et al., 2023a) offer a promising explainable AI (XAI) approach by providing insights that can drive scientific discoveries. Unlike black-box models, CMs first predict human-interpretable concepts and then use them to predict task labels, providing both predictions and rationales. For example, in computer vision, CMs predict bird species by identifying concepts like “wing color” (Koh et al., 2020). In molecular science, CMs interpret predictions through concepts like functional groups and molecular descriptors. As shown in Figure 1, a CM predicts molecular solubility using descriptors like *# of nitrogen atoms* and *TPSA*, allowing researchers to refine molecules based on these key features.

Despite their promise, CMs have seen limited application in molecular science due to challenges in concept generation and labeling. Existing CM methods either rely on predefined concepts and manual labeling by experts (Koh et al., 2020) or are limited to simple, qualitative concepts inadequate for molecular problems (Oikarinen et al., 2023). While feasible in computer vision (Koh et al., 2020; Oikarinen et al., 2023), molecular concepts are more complex and require precise quantitative labels, like *TPSA* in Figure 1, which reflects absorption and permeability relevant to solubility prediction. Identifying such concepts demands domain expertise and computational methods beyond current CM capabilities, posing a significant challenge for their effective use in molecular science.

In response to the effectiveness of CMs and the

*Equal Contribution.

†Work done when authors were at University of California Los Angeles.

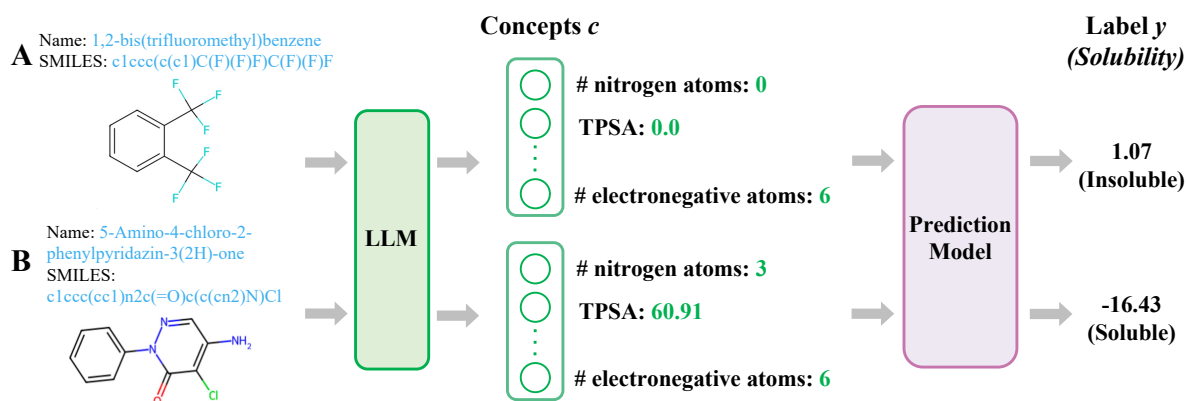


Figure 1: The prediction process of molecule properties is greatly illuminated with AutoMolCo. First, concepts are generated and labeled with an LLM. Then, a simple prediction model (e.g., linear regression) is fitted to achieve explainable predictions. LLM-relevant pieces are highlighted in green.

challenges of applying them to molecular science, we propose **Automated Molecular Concept (AutoMolCo)** generation and labeling. AutoMolCo leverages Large Language Models (LLMs) to generate molecular concepts that are predictive for the task and label these concepts for each molecule instance. AutoMolCo also repeats these procedures through iterative interactions with LLMs to refine concepts, enabling simple linear models on the refined concepts to outperform GNNs and LLM in-context learning (ICL) on several molecular benchmarks. The whole framework is automated and does not require human knowledge inputs in either concept generation, labeling, or refinement, thus surpassing the limitations of extant CMs.

The motivation behind AutoMolCo is the idea that LLMs can serve as extensive knowledge bases (Petroni et al., 2019; Alkhamissi et al., 2022), with their effectiveness for solving molecular science problems demonstrated through ICL (Guo et al., 2023b). We leverage LLMs for XAI by integrating them into CMs. For concept generation, we prompt LLMs with the task description to suggest relevant concepts. For concept labeling, we explore three methods: direct LLM prompting, function code generation, and external tool calling. We then build simple prediction models on these concepts. Additionally, we iteratively refine concepts by running feature selection and prompting LLMs to generate improved concepts, ensuring the CM remains up-to-date with the most relevant concepts and enhances performance.

In this work, we first show that AutoMolCo can produce meaningful concepts and accurate labels, which lead to CMs with simple prediction models to achieve surprisingly good performance

for molecular science problems. Then we perform a systematic study of AutoMolCo on MoleculeNet (Wu et al., 2018) and High-Throughput Experimentation (HTE) (Ahneman et al., 2018; Reizman et al., 2016) datasets to answer five research questions. In summary, our contribution includes:

- **Automated framework:** We propose AutoMolCo, which leverages LLMs for automated concept generation and labeling, eliminating the need for human domain knowledge and labor-intensive data collection, thereby streamlining the development of CMs.
- **Accuracy and explainability:** AutoMolCo produces meaningful molecular concepts that, when combined with simple prediction models for CMs, can achieve superior or comparable accuracy to powerful black-box models while providing greater explainability.
- **LLM-driven XAI for science:** Our work highlights the potential of LLMs in addressing complex molecular science problems, introduces a novel perspective on CMs with LLMs, and paves the way for future research to exploit the LLMs’ capabilities in molecular science and beyond.

2 Related work

Concept-based Models A well-known example of CMs is the Concept Bottleneck Model (CBM) (Koh et al., 2020), which predicts through an intermediate layer of human-specified concepts, like "wing color" in bird classification. While transparent, CBMs are limited by predefined concepts and label requirements. Several variations of CBMs target specific tasks (De Fauw et al., 2018; Yi et al., 2018; Bucher et al., 2019; Losch et al., 2019; Chen et al., 2020), with a notable one,

label-free CBM (Oikarinen et al., 2023), bypasses predefined concepts by using GPT-3 for concept generation and CLIP-Dissect for matching concepts with images. However, it focuses on vision tasks and generates simple, qualitative concepts (e.g., "yellow") that are insufficient for molecules, which demand deeper chemical knowledge and precise quantitative labels.

Explainable Learning on Scientific Graphs Data

Explainable learning on graph data is getting popular, especially for scientific problems like particle identifying (Mokhtar et al., 2022), whether prediction (Jeon et al., 2024), material design (Wang et al., 2020; Li et al., 2024), and in particular, molecular science (Yuan et al., 2021; Zhang et al., 2022; Wu et al., 2023b). One line identifies graph motifs as concepts through counting or sampling (Milo et al., 2002; Wernicke, 2006) and builds GNNs on top of them (Zhang et al., 2020; Yu and Gao, 2022). However, motif identification cannot be comprehensive as it is NP-complete. Another line tries to use concept-based explanations for GNNs with human-in-the-loop (Magister et al., 2021). Subsequent works have refined this idea with k-means clustering and similarity scoring algorithms to neuron-level grouping within activation layers (Magister et al., 2022; Xuanyuan et al., 2023). These methods exemplify the attempt to extract and interpret salient features in graph data, yet they often face challenges in fully capturing the nuanced complexity of molecular structures.

LLMs for Molecular Science Recently, there are some benchmarking papers on LLMs for molecular science. GPT4Graph (Guo et al., 2023a) prompts LLMs to explain the format or to summarize a raw molecule graph input, where the graph is represented by the Graph Modelling Language (GML) (Himsolt, 1997) or Graph Markup Language (GraphML) (Brandes et al., 2013). Graph-ToolFormer (Zhang, 2023) lets LLMs generate API calls to use external graph reasoning tools, which can be applied to molecule function reasoning problems. (Guo et al., 2023b) studies solving molecular problems with LLMs ICL. We show AutoMolCo outperforms ICL and enjoys better explainability. Some survey papers discussing LLMs' potential for molecular science include: (Zhang et al., 2023) from a scientific research perspective and (Jin et al., 2023) from an LLM for graph perspective, and (Yu et al., 2024) for fine-tuning LLMs.

3 AutoMolCo: automated molecular concept generation and labeling

In this section, we describe AutoMolCo for concept generation, labeling, and refinement, where the concepts are used to build an explainable CM. Figure 2 depicts the three major steps of AutoMolCo: 1) concept generation, 2) concept labeling, and 3) CM fitting and concept selection.

Step 1: Concept Generation Given a particular task on molecules, e.g., predicting the hydration-free energy of small molecules in water, the first step is to prompt LLMs to propose a diverse list of concepts that are potentially relevant to the task. This step is analogous to a brainstorming process. Concepts range from counting-based ones, like *# nitrogen atoms*, to more complicated ones that require precise calculation, like TPSA. Without LLMs, coming up with meaningful concepts requires domain experts. The underlying intuition for concept generation is founded on the idea that LLMs can be treated as extensive and integrated knowledge bases. Their capacity to comprehend and output meaningful concepts is pivotal in this phase, yielding a wide spectrum of potentially relevant concepts for our analysis. The prompt for this step is shown in Figure 7 Step 1. The LLM-suggested concepts might be less relevant initially, but they will be refined later.

Step 2: Concept Labeling Following the concept generation step, we then label the generated concepts for each data instance. Compared to human labeling, which requires domain knowledge and can be labor-intensive. Labeling with LLMs is streamlined to a process of interaction with a single LLM interface, which can be easily scaled and minimizes human error. This automation with LLMs is crucial for efficiently processing large volumes of data encountered in molecular studies. In this step, we consider three different labeling strategies to enhance labeling quality.

Labeling Strategy 1: Direct LLM prompting. We prompt LLMs directly to assign each data instance numerical or categorical labels for the generated concepts from Step 1. Similar to concept generation, this strategy relies on that LLMs can be treated as integrated knowledge bases for retrieving useful information. For each data instance, we provide LLMs with the molecule names or SMILES strings. The prompt is shown in Figure 7 Step 2.

Labeling Strategy 2: Function code generation with LLMs. Since LLMs are particularly

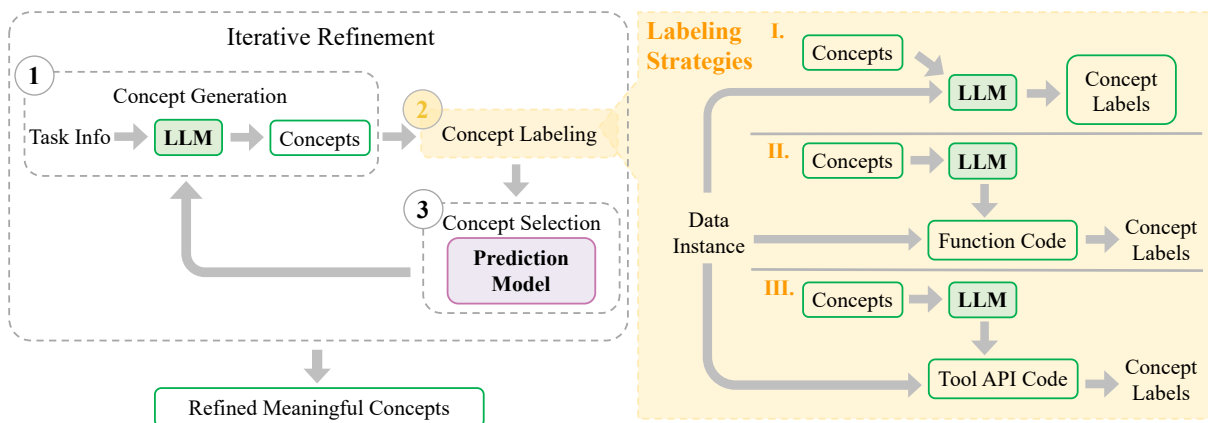


Figure 2: **The AutoMolCo framework.** Step 1: concept generation. Step 2: concept labeling with three different strategies. Step 3: fitting a prediction model and perform concept selection. These three steps are repeated for multiple iterations to achieve a refined list of meaningful concepts, where the selected concepts in each iteration are feedback to the LLM through prompting. LLM outputs are highlighted as green boxes.

skilled in code generation, we explore a second approach for concept labeling: generate functions in Python code for computing the concept labels. The function generation approach has two advantages. Firstly, it greatly reduces the need for repeated LLM API calls. Only a single API call is required for each concept to obtain the function code, as opposed to making a separate call for each data instance in the direct label prompting case. Secondly, the generated functions can utilize pre-processed dataset features as function arguments, such as atom types in terms of node features and molecule structures in terms of adjacency matrix. These features provide more direct information beyond molecule names or SMILES strings. Leveraging these features, the LLM-generated functions can offer more nuanced and accurate concept labels, enhancing the effectiveness of AutoMolCo. The prompt is shown in Figure 5 in Appendix A.

Labeling Strategy 3: External tool calling with LLMs. We also utilize LLMs to call external tools like RDKit (Landrum, 2010) for labeling, which combines the LLM generation with the specialized tool reliability. This strategy enjoys the same efficiency advantage as the function generation approach, meaning it requires only a single API call of the LLM per concept to get the API code for calling the tool. Moreover, the use of labeling tools ensures that labels for all the tool-calculable concepts are accurate and reliable. One disadvantage of this strategy is that not all generated concepts are calculable by the external tool, in which case we can only turn to the first two strategies. The prompt is shown in Figure 6 in Appendix A.

Step 3: CM Fitting and Concept Selection

After getting the generated concepts and their labels, we utilize them to fit prediction models for the molecular task. Since the concept labels can be treated as tabular data, any model from the off-the-shelf ones in Scikit-learn (Pedregosa et al., 2011) to sophisticated deep learning models can be applied. However, we found that explainable models like linear models and decision trees or simple two-layer multi-layer perceptions (MLPs) are often sufficient for achieving competitive performance. We attribute the credit to the high-quality concepts and their labels. We will discuss our model choice and perform a systematic study of different prediction models in Section 4. While fitting the model, we also run feature selection methods like Akaike Information Criterion (AIC) (Akaike, 1973, 1974) and Recursive Feature Elimination (RFE) (Guyon et al., 2002a) to determine the useful concepts. Feature selection not only boosts the model performance but also leads to automated iterative refinement for identifying the most useful concepts.

Iterative Concepts Refinement After all three steps. We do an iterative refinement of the generated concepts by prompting LLMs again with the empirical performance of our prediction model and the concept selection results from Step 3. We include such information in an updated prompt to make LLMs generate new concepts to replace the less useful ones from the previous iteration. Using the empirical results as feedback, we ensure that our CM remains adaptable and up-to-date with the most relevant molecular concepts. Through this iterative refinement process, we guarantee that the model performance improves over iterations and prune the irrelevant concepts generated in previous

iterations. The prompt for this step is shown in Appendix A: Figure 7.

4 Experiments

4.1 Experiment settings

Datasets We include four datasets from MoleculeNet (Wu et al., 2018): two regression datasets (FreeSolv and ESOL) and two classification datasets (BBBP and BACE). FreeSolv provides hydration-free energy for 642 molecules, while ESOL contains water solubility data for 1128 small organic molecules. BBBP (2,039 molecules) assesses blood-brain barrier penetration, and BACE (1,513 molecules) predicts β -secretase 1 inhibitors for Alzheimer’s research. All datasets use the scaffold splits from the Open Graph Benchmark (OGB) (Hu et al., 2020). Additionally, we include two HTE datasets, Buchwald-Hartwig (BH) (Ahne-man et al., 2018) and Suzuki-Miyaura (SM) (Reiz-man et al., 2016), for reaction yield prediction, with BH covering 3,957 molecules and SM covering 5,650. These datasets use the same splits as (Guo et al., 2023b).

Metrics We follow the standard evaluation metrics for these datasets. For FreeSolv and ESOL, results are measured with Root Mean Square Error (RMSE). For BBBP and BACE, we mainly evaluate these datasets using AUC-ROC and report results in our main Table 1. Since (Guo et al., 2023b) evaluates them with accuracy, we also report accuracy comparison in Table 6 in Appendix E. For BH and SM, we evaluate with accuracy.

Baselines Our baselines include GNNs, LLM ICL, and GNN + CBM. Specifically, we use the GIN and GCN. For LLM ICL, we refer to the findings in (Guo et al., 2023b) and use their prompts. For GNN + CBM, we use GIN and use GPT-3.5 Turbo for generating concept labels for CBM. For HTE datasets, we only consider LLM ICL baselines as graphs are not provided for the test set.

Models We employ GPT-3.5 Turbo as our primary LLMs for generating concepts and direct labeling. Additionally, we utilize GPT-4 for labeling strategy 2: function code generation, and strategy 3: external tool calling. For strategy 3, the LLM will create code snippets for invoking RDKit (Landrum, 2010). We don’t use GPT-4 for direct labeling due to the high cost of per-instance labeling. After collecting the concept labels, we explore four types of prediction models to cover a broad spectrum of tasks and performance levels. As a basic setting, we

use linear models like linear regression and logistic regression, we also consider more advanced models including decision trees and 2-layer MLPs. We use off-the-shelf prediction models from sklearn (Pedregosa et al., 2011). We do ablation on LLMs with Claude-2 in Appendix D. Since we call LLMs through their APIs and the prediction models are light and off-the-shelf, there is no specially requirements, like GPUs, for our framework.

Concept Selection We employ AIC (Akaike, 1973, 1974) for regression and RFE (Guyon et al., 2002b) for classification. These selection methods are specifically applied to linear models, and we use the selection results for multi-iteration performance with decision trees and MLPs.

4.2 AutoMolCo-induced CM performance

In Table 1, we compare the performance of the AutoMolCo-induced CM to baselines. Compared to GNNs, our CM achieves better results on MoleculeNet regression tasks and HTE tasks and competitive results on MoleculeNet classification tasks. In comparison to the results presented by ICL, our models have demonstrated a substantial performance advantage on all tasks. Our best-performing model is the culmination of multiple iterations of refinement and a combination of labeling strategies. Specifically, the results presented in Table 1 are achieved using the following approaches: **1.** A combination of all three labeling strategies for concept labeling, with further details provided in Appendix D.3; **2.** The optimal CMs from linear models, decision trees, and MLPs; **3.** Concepts refinement over three iterations, as discussed in Section 4.6. An in-depth exposition of these techniques is discussed in detail in the RQs below through experiments on MoleculeNet datasets. More details on experiment results for HTE datasets are shown in Appendix G.

4.3 RQ1: Can AutoMolCo generate meaningful molecular concepts?

The effectiveness of the CM relies on meaningful concepts, traditionally provided by domain experts (Koh et al., 2020). In RQ1, we evaluate AutoMolCo’s concept generation through iterative refinement and expert consultation. Figure 3 illustrates how concepts selected by a linear regression model for predicting solubility (FreeSolv) evolve from a broad initial set to a more focused, chemically relevant set. Experts noted that eliminated concepts, such as counts of specific atoms and rotat-

	FreeSolv (\downarrow)	ESOL (\downarrow)	BBBP (\uparrow)	BACE (\uparrow)	BH (\uparrow)	SM (\uparrow)
GIN	2.151	0.998	69.710	73.460	-	-
GCN	2.186	1.015	67.800	68.930	-	-
GIN + CBM	2.412	1.373	54.500	68.457	-	-
GPT-3.5 Turbo (zero-shot)	5.450	2.039	49.256	48.765	0.320	0.473
GPT-3.5 Turbo (4-shot)	4.852	1.161	51.580	41.871	0.640	0.630
GPT-3.5 Turbo (8-shot)	4.491	1.128	56.632	47.757	0.706	0.693
AutoMolCo-CM (ours)	2.065	0.843	65.278	70.744	0.810	0.800

Table 1: Performance comparison of the AutoMolCo-induced CM with baselines. MoleculeNet regression tasks (FreeSolv and ESOL) are measured in RMSE (\downarrow). MoleculeNet classification tasks (BBBP and BACE) are measured in AUC-ROC (\uparrow). HTE datasets (BH and SM)¹ are measured in accuracy (\uparrow). Ours achieve better results on MoleculeNet regression and HTE tasks and competitive results on MoleculeNet classification tasks.

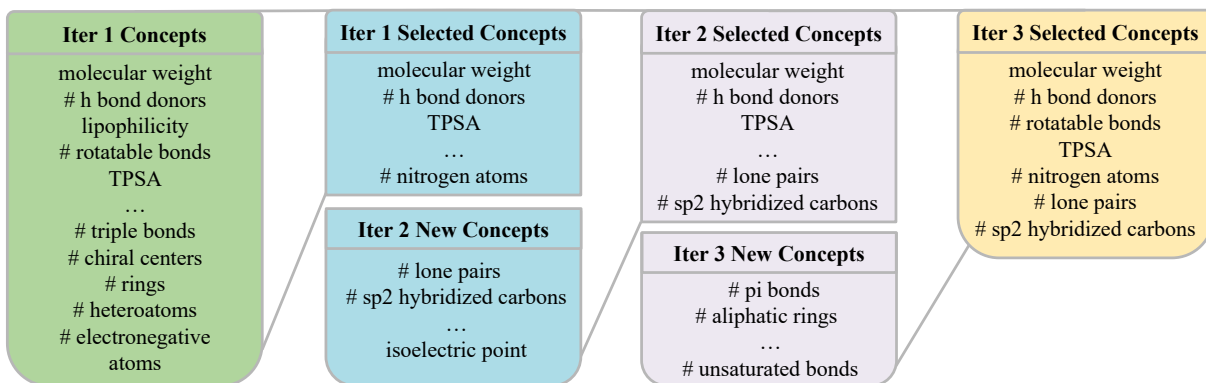


Figure 3: RQ1: Concepts selected by AutoMolCo in three refinement iterations on FreeSolv. A detailed version in Appendix B Figure 8.

able bonds, while informative, contributed less to predictive power or were correlated with retained concepts. The selected concepts, including molecular weight, # hydrogen bond donors, and TPSA, are fundamental properties influencing molecular interactions and solubility. For example, hydrogen bond donors relate directly to hydrogen bonding, and TPSA quantifies polar interaction surfaces crucial for solubility in polar solvents (Pajouhesh and Lenz, 2005). These final concepts align well with domain knowledge.

AutoMolCo also mitigate the potential LLM hallucination issue when generating concepts. Our experiments show LLMs perform better on straightforward, well-studied tasks but struggle with complex, experimental, or sparsely documented topics. However, in both cases, LLM hallucination and framework performance can be mitigated by employing combined labeling strategies and iterative refinement, resulting in high quality molecular con-

cepts.

4.4 RQ2: Can AutoMolCo assign molecules reasonable concept labels using each strategy?

Accurate concept labels are another critical component of CM performance. In this RQ2, we evaluate AutoMolCo labeling results. We collect ground truth labels for concepts where labels are available, either through calculation (e.g., for molecular weights), or manual lookup (e.g., for melting points). We evaluate labels produced by our direct prompting strategy and function generation labeling strategy using the Pearson correlation coefficient (r) with the ground truth, due to the scale-invariant nature of the metric. The external tool calling strategy is excluded from this evaluation as tools will always provide correct labels, and the downside of this strategy is that not all the concepts are tool-calculable (e.g., melting points). Results in Table 2 show strong correlations can be achieved on most datasets with AutoMolCo labeling. Nonetheless, variations in correlation underscore the potential for method improvement.

¹The performance of GNN-based methods on BH and SM is not reported because we could not obtain the correct train and test splits for their graph datasets from either the original paper or subsequent papers that utilized BH and SM.

It’s worth notice that correlations in Table 2 do not have a direct connection to the performance of each labeling strategy. It serves as a sanity check for labeling strategies but an incomplete picture of prediction performance. This is because some concept correlations are non-computable due to missing ground-truth labels, which also contribute to the prediction performance results in Table 3.

In addition to this benchmarking effort, we also discuss several challenges we encountered and overcame in each labeling strategy, including imputation for missing values, dictionary for unit inconsistency, and Chain-of-Thoughts (CoT) prompts for syntax errors in function code:

Direct LLM prompting For labeling with direct LLM prompting, we encountered two key issues: missing labels and unit inconsistency.

Missing labels One issue we found for concept labeling with direct LLM prompting is that it is challenging to have LLMs generate some concept labels for certain molecules. For instance, LLMs identified *acid dissociation constant* (pK_a) as a crucial concept for predicting water solubility. However, pK_a is a quantity only apply to acids, and thus the model will output “Unknown” for the label. For the ESOL dataset, this results in a 13.03% missing rate for this concept. The missing-label issue underscores AutoMolCo’s limitation in recognizing concept applicability across molecules. To mitigate this, we apply various imputation methods, including mean value imputation and domain-knowledge-driven imputation. For the latter, we set missing pK_a labels to 100—significantly above water’s pK_a of 14—to denote weak or non-acidity, which enhances the CM’s performance.

Unit inconsistency For concepts with multiple possible units, labels generated by LLMs can exhibit inconsistent units across molecules. For example, in our experiments on the ESOL dataset, the LLM suggested "melting point" as relevant for predicting water solubility but used inconsistent units—Celsius ($^{\circ}\text{C}$), Fahrenheit (F), or Kelvin (K)—for the melting point values.

Similar issues arose with other concepts like *molecular volume* and *molecular surface area* due to randomness in LLM context generation when processing different data instances. Our initial attempts to fix this by specifying units in the prompt were ineffective. To address this, we introduced an intermediate step: the LLM generates a concept-to-unit dictionary for proposed concepts, which

is then integrated into Step 2’s prompt to ensure consistent units in the generated labels.

Function code generation When generating labeling functions in Python code, we find it is non-trivial to prompt LLMs for executable functions with no errors. We made two efforts to increase the likelihood of producing executable functions with LLMs. We first perform prompt engineering to clearly specify atom types, adjacency matrices, and node and edge features, which enhances the function quality. Through careful prompt engineering, most generated functions for simpler concepts become executable. However, functions for labeling complex concepts like “number of rings” are still unlikely to be error-free due to their intricate nature. We thus adopt a chain of thought (CoT) approach to generate functions. For the CoT prompt, we first ask the LLM to describe the function in natural language, which can best leverage the LLM’s strength in generating natural language. Then, the CoT prompt asks the LLM to turn the natural language description of the function into Python code, which we found increases the likelihood of generating accurate and executable functions. An example of the CoT function-generation prompt is shown in Figure 5.

External tool calling Given there are external tools for molecular science with API access, we prompt LLMs to generate code snippets for calling the tool API. We observe that LLMs are adept at obtaining callable APIs for a majority of our generated concepts from step 1, which we successfully employed to calculate the concept labels for each molecule in our dataset. The example prompts and generated API calls can be found in Figure 6. The drawback of this strategy is that the external tool cannot cover all the concepts generated by the LLM, especially for those measured concepts like melting point. For these cases, we turn to the first two strategies for labeling.

4.5 RQ3: Can AutoMolCo-generated concepts and labels be utilized to build an effective CM?

In RQ1, we have verified that the generated concepts are meaningful according to domain experts. In RQ2, we have shown that concept labels are relatively accurately assigned after properly handling potential issues like missing labels and unit inconsistency. In this RQ3, we compare the performance of the AutoMolCo-induced CMs when

Labeling Strategy	LLM	Molecule Format	FreeSolv	ESOL	BBBP	BACE
Str-1 Direct Prompt	GPT-3.5	Name	0.82	0.63	0.06	-
Str-1 Direct Prompt	GPT-3.5	SMILES	0.82	0.75	0.69	0.22
Str-2 Function	GPT-4	-	1.00	0.79	0.69	0.67

Table 2: RQ2: Percentage of concepts with a high correlation (r score ≥ 0.7) with the ground-truth.

different prediction models and labeling strategies are adopted. Results in Table 3 show that AutoMolCo can give reasonable performance even with the most basic direct prompting labeling strategy and the simplest linear model. The good performance of different prediction models demonstrates the quality of the concepts and the effectiveness of AutoMolCo.

4.6 RQ4: Does iterative refinement boost the performance of AutoMolCo-induced CM?

As one of the most important designs of our AutoMolCo framework, concept refinement helps to identify meaningful important concepts through iterative interactions with LLMs. The concept relevance has been shown to improve in RQ1, but that does not necessarily mean CM performance will also improve. In this RQ4, we run AutoMolCo with three iterative concept refinements on the MoleculeNet datasets with linear prediction models. We show the results in Figure 4 (a) and (b), and we observe that the CM prediction performance indeed improved through concept refinement, especially for classification tasks. The improvement for regression tasks is marginal, partially because the performance is already good for regression.

4.7 RQ5: Does the AutoMolCo-induced CM facilitates explainable molecular science?

One of the key advantages of CMs over black-box models is their explainability. In this section, we evaluate this aspect of AutoMolCo-induced CMs through three experiments on all three types of the prediction models: coefficient interpretation of linear models, split interpretation of decision trees, and concept label intervention of MLPs.

Coefficient Interpretation of Linear Models

Using linear model in the AutoMolCo-induced CM offers excellent explainability through direct interpretation of the model coefficients. We plot the coefficients of the linear model on FreeSolv for predicting hydration free energy in Figure 4 (c), highlighting three significant concepts: *# hydrogen bond donors*, *TPSA*, and *# rotatable bonds*. According to domain experts, the *# hydrogen bond donors* relates to a molecule’s ability in hydrogen bond-

ing, reflecting its potential to interact with solvents and other molecules. Therefore, an its increment typically leads to a more favorable (more negative) hydration free energy (Chung and Park, 2015). *TPSA* quantifies the surface area of a molecule that can engage in polar interactions, providing insights into a molecule’s permeability characteristics. Thus, higher *TPSA* also leads to more favorable (more negative) hydration free energy (Pajouhesh and Lenz, 2005). Conversely, the *# rotatable bonds* positively correlated with hydration free energy. More rotatable bonds increase molecular flexibility, allowing the molecule to adopt conformations that enhance interactions with water molecules. This increased flexibility can lead to less favorable hydration free energy (less negative), as it reduces the stability of the solvation shell around the molecule (Guimarães and Cardozo, 2008). Our linear model interpretation aligns with domain knowledge without requiring any human knowledge input into the model. We show results on BBBP in Appendix C.

Splits Interpretation of Decision Trees Complementing to the coefficients of the linear model, decision tree enhances the understanding of model’s decision process. In Figure 11, we show the 3-layer decision tree for BBBP dataset. In the first two layers, the model uses *TPSA* to categorize the molecules into four categories, where molecules with *TPSA* less than 26.99 are likely to penetrate the BBB while molecules with *TPSA* greater than 176.22 are rarely penetrative. The decision tree further differentiates molecules with *TPSA* between 26.99 and 107.1 by whether or not it contains a hydrogen bond in its ring structure, where molecules without this property are more likely to penetrate the BBB. On the other hand, the model splits molecules with *TPSA* between 107.1 and 176.22 using the number of carbon atoms, illustrating that molecules containing more than 23 carbon atoms are very likely to be penetrative. Figure 13 shows the details of the decision tree.

Concept Label Intervention Besides analyzing interpretable prediction models like linear models and decision trees, we also conduct a case

Labeling Strategy	Prediction Model	FreeSolv(↓)	ESOL (↓)	BBBP (↑)	BACE (↑)
Str-1 Direct Prompt	Linear/Logistic	2.685	1.250	52.836	56.894
Str-1 Direct Prompt	Decision Tree	2.791	1.272	56.887	68.632
Str-1 Direct Prompt	MLP	2.338	1.194	51.794	60.059
Str-2 Function	Linear/Logistic	3.284	1.254	55.671	56.624
Str-2 Function	Decision Tree	2.569	1.238	54.167	55.573
Str-2 Function	MLP	2.805	1.034	58.738	56.894
Str-3 Tool	Linear/Logistic	3.142	1.011	57.350	63.154
Str-3 Tool	Decision Tree	3.750	1.027	55.903	65.658
Str-3 Tool	MLP	1.981	0.911	58.449	60.772

Table 3: RQ3: Model performance with different labeling strategies and prediction models.

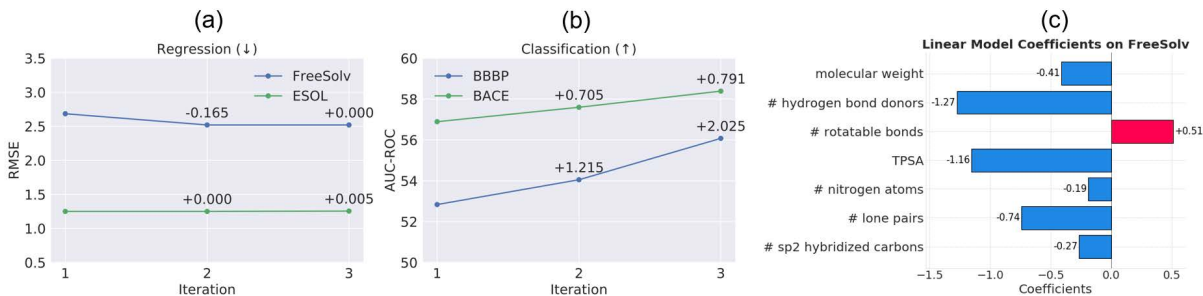


Figure 4: RQ4: Iterative refinement improves CM performance for (a) regression and (b) classification tasks; (c) RQ5: Coefficients of the linear regression model from AutoMolCo after 3 iterations on FreeSolv.

study of concept label interventions with MLPs. Our goal is to identify molecules with similar concept labels except for the one we intend to intervene on (e.g., similar molecular weights, # aromatic rings, etc., except logP) but different task labels (e.g., soluble vs. insoluble). Two examples we identify from the ESOL dataset are: *Diphenylamine* (N(c1ccccc1)c2ccccc2) and *RTI 17* (CCN2c1ccccc1N(C)C(=S)c3ccccc23). After three iterations of refinement these two molecules have the same labels for three out of the four remaining concepts, except for their logP labels, which differ by 0.275 (standardized to have mean 0 and standard deviation 1). Diphenylamine is predicted to be insoluble (-3.648), whereas RTI 17 is predicted to be soluble (-4.079), based on a conventional solubility threshold of -4 (Sorkun et al., 2019). These predictions proved to be quite accurate, with the ground truth solubility of these two molecules being -3.857 and -4.227, respectively. By intervening on diphenylamine’s logP value (0.209) to match RTI 17’s logP value (0.484) through interpolation, we observe a linear change in solubility. This study highlights the significant impact of logP on solubility predictions, which is consistent with expert conclusions (Lipinski et al., 1997; Avdeef, 2012), providing insights beyond black-box models. We show

the intervention plot in Appendix C Figure 10.

4.8 Ablation studies

We conduct ablation studies of AutoMolCo. We found that AutoMolCo can perform consistently with different LLMs and is robust to molecule input formats. Also, properly combining the labeling strategies can enhance model performance. These results are in Appendix D.

5 Conclusion

We propose the AutoMolCo framework that automates the generation and labeling of molecular concepts, overcoming challenges of existing CMs and enhancing explainability through iterative refinement of useful concepts. We demonstrate that, for molecular property prediction tasks, simple linear prediction model on our generated concepts can perform competitively or even better than GNNs and LLM ICL. Our work paves the way for future research to further exploit the capabilities of LLMs for XAI in molecular science and beyond.

6 Limitations

The AutoMolCo framework’s performance and explainability rely on the quality of LLM-generated concepts and labels. Although we conducted rigor-

ous experiments and verified the LLM-generated concepts and labels with domain experts to ensure the quality of our experiment results, limitations of LLM, such as potential hallucination and rare occurrence of certain chemical formula representations during pre-training stage, may effect the performance and reliability of our framework in other instances.

As a general framework, we expect the performance and reliability of AutoMolCo to be further improved with newer and more advanced LLMs. A thorough investigation on mechanistic interpretability of LLM and a more powerful LLM dedicated to molecular science may address these issues and could be considered as future directions.

Another limitation is that the evaluation of the generated concepts and labels often requires validation by human experts, introducing subjectivity and dependency on domain knowledge. Developing automated evaluation methods is another potential direction for improvement.

Acknowledgments

This work was partially supported by NSF 2211557, NSF 1937599, NSF 2119643, NSF 2303037, NSF 2312501, NASA, SRC JUMP 2.0 Center, Amazon Research Awards, and Snapchat Gifts.

References

- DT Ahneman, JG Estrada, S Lin, SD Dreher, and AG Doyle. 2018. Predicting reaction performance in c–n cross-coupling using machine learning. *Science*.
- Hirotsugu Akaike. 1973. Information theory and an extension of the maximum likelihood principle. In *2nd International Symposium on Information Theory*, pages 267–281. Akademiai Kiado.
- Hirotsugu Akaike. 1974. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.
- Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona Diab, and Marjan Ghazvininejad. 2022. A review on language models as knowledge bases. *arXiv preprint arXiv:2204.06031*.
- Alex Avdeef. 2012. *Absorption and drug development: solubility, permeability, and charge state*. John Wiley & Sons.
- Ulrik Brandes, Markus Eiglsperger, Jürgen Lerner, and Christian Pich. 2013. Graph markup language (graphml).
- Maxime Bucher, Stéphane Herbin, and Frédéric Jurie. 2019. Semantic bottleneck for computer vision tasks. In *Computer Vision—ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part II 14*, pages 695–712. Springer.
- Zhi Chen, Yijie Bei, and Cynthia Rudin. 2020. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782.
- Kee-Choo Chung and Hwangseo Park. 2015. Accuracy enhancement in the estimation of molecular hydration free energies by implementing the intramolecular hydrogen bond effects. *Journal of Cheminformatics*, 7:1–12.
- Jeffrey De Fauw, Joseph R Ledsam, Bernardino Romera-Paredes, Stanislaw Nikolov, Nenad Tomasev, Sam Blackwell, Harry Askham, Xavier Glorot, Brendan O’Donoghue, Daniel Visentin, et al. 2018. Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nature medicine*, 24(9):1342–1350.
- Cristiano RW Guimarães and Mario Cardozo. 2008. Mm-gb/sa rescoring of docking poses in structure-based lead optimization. *Journal of chemical information and modeling*, 48(5):958–970.
- Jiayan Guo, Lun Du, and Hengyu Liu. 2023a. Gpt4graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking. *arXiv preprint arXiv:2305.15066*.
- Taicheng Guo, Kehan Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. 2023b. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. *arXiv preprint arXiv:2305.18365*.
- Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. 2002a. Gene selection for cancer classification using support vector machines. *Machine learning*, 46:389–422.
- Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. 2002b. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422.
- Michael Himsolt. 1997. Gml: Graph modelling language. *University of Passau*.
- Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133.
- Hyeon-Ju Jeon, Jeon-Ho Kang, In-Hyuk Kwon, O Lee, et al. 2024. Cloudnine: Analyzing meteorological observation impact on weather prediction using explainable graph neural networks. *arXiv preprint arXiv:2402.14861*.

- Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. 2023. Large language models on graphs: A comprehensive survey. *arXiv preprint arXiv:2312.02783*.
- Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. 2020. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR.
- Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. 2009. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE conference on computer vision and pattern recognition*, pages 951–958. IEEE.
- Greg Landrum. 2010. **RDKit: Open-source cheminformatics**. <https://www.rdkit.org>. Accessed: Nov 22, 2023.
- Haoyu Li, Shichang Zhang, Longwen Tang, Mathieu Bauchy, and Yizhou Sun. 2024. **Predicting and interpreting energy barriers of metallic glasses with graph neural networks**. In *Forty-first International Conference on Machine Learning*.
- Christopher A Lipinski, Franco Lombardo, Beryl W Dominy, and Paul J Feeney. 1997. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced drug delivery reviews*, 23(1-3):3–25.
- Max Losch, Mario Fritz, and Bernt Schiele. 2019. Interpretability beyond classification output: Semantic bottleneck networks. *arXiv preprint arXiv:1907.10882*.
- Lucie Charlotte Magister, Pietro Barbiero, Dmitry Kazhdan, Federico Siciliano, Gabriele Ciravegna, Fabrizio Silvestri, Mateja Jamnik, and Pietro Liò. 2022. Encoding concepts in graph neural networks. *arXiv preprint arXiv:2207.13586*.
- Lucie Charlotte Magister, Dmitry Kazhdan, Vikash Singh, and Pietro Liò. 2021. Gcexplainer: Human-in-the-loop concept-based explanations for graph neural networks. *arXiv preprint arXiv:2107.11889*.
- Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. 2002. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827.
- Farouk Mokhtar, Raghav Kansal, and Javier Duarte. 2022. Do graph neural networks learn traditional jet substructure? *arXiv preprint arXiv:2211.09912*.
- Tuomas Oikarinen, Subhro Das, Lam Nguyen, and Lily Weng. 2023. Label-free concept bottleneck models. In *International Conference on Learning Representations*.
- H. Pajouhesh and G.R. Lenz. 2005. **Medicinal chemical properties of successful central nervous system drugs**. *NeuroRx*, 2(4):541–553.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.
- Brandon J. Reizman, Yi-Ming Wang, Stephen L. Buchwald, and Klavs F. Jensen. 2016. **Suzuki–miyaura cross-coupling optimization enabled by automated feedback**. *React. Chem. Eng.*, 1:658–666.
- Murat Cihan Sorkun, Abhishek Khetan, and Süleyman Er. 2019. Aqsolddb, a curated reference set of aqueous solubility and 2d descriptors for a diverse set of compounds. *Scientific data*, 6(1):143.
- Jonathan M Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M Donghia, Craig R MacNair, Shawn French, Lindsey A Carfrae, Zohar Bloom-Ackermann, et al. 2020. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702.
- Qi Wang, Jun Ding, and Evan Ma. 2020. **Predicting the propensity for thermally activated β events in metallic glasses via interpretable machine learning**. *Preprint*, arXiv:2006.13552.
- Sebastian Wernicke. 2006. Efficient detection of network motifs. *IEEE/ACM transactions on computational biology and bioinformatics*, 3(4):347–359.
- Zhengxuan Wu, Karel D’Oosterlinck, Atticus Geiger, Amir Zur, and Christopher Potts. 2023a. Causal proxy models for concept-based model explanations. In *International conference on machine learning*, pages 37313–37334. PMLR.
- 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.
- Zhenxing Wu, Jike Wang, Hongyan Du, Dejun Jiang, Yu Kang, Dan Li, Peichen Pan, Yafeng Deng, Dongsheng Cao, Chang-Yu Hsieh, et al. 2023b. Chemistry-intuitive explanation of graph neural networks for molecular property prediction with substructure masking. *Nature Communications*, 14(1):2585.
- Han Xuanyuan, Pietro Barbiero, Dobrik Georgiev, Lucie Charlotte Magister, and Pietro Liò. 2023. Global concept-based interpretability for graph neural networks via neuron analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10675–10683.

- Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. 2020. On completeness-aware concept-based explanations in deep neural networks. *Advances in neural information processing systems*, 33:20554–20565.
- Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. 2018. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. *Advances in neural information processing systems*, 31.
- Botao Yu, Frazier N. Baker, Ziqi Chen, Xia Ning, and Huan Sun. 2024. [Llasmol: Advancing large language models for chemistry with a large-scale, comprehensive, high-quality instruction tuning dataset](#). *Preprint*, arXiv:2402.09391.
- Zhaoning Yu and Hongyang Gao. 2022. Molecular representation learning via heterogeneous motif graph neural networks. In *International Conference on Machine Learning*, pages 25581–25594. PMLR.
- Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. 2022. Explainability in graph neural networks: A taxonomic survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(5):5782–5799.
- Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. 2021. On explainability of graph neural networks via subgraph explorations. In *International conference on machine learning*, pages 12241–12252. PMLR.
- Jiawei Zhang. 2023. Graph-toolformer: To empower llms with graph reasoning ability via prompt augmented by chatgpt. *arXiv preprint arXiv:2304.11116*.
- Shichang Zhang, Ziniu Hu, Arjun Subramonian, and Yizhou Sun. 2020. Motif-driven contrastive learning of graph representations. *arXiv preprint arXiv:2012.12533*.
- Shichang Zhang, Yozen Liu, Neil Shah, and Yizhou Sun. 2022. Gstarx: Explaining graph neural networks with structure-aware cooperative games. *Advances in Neural Information Processing Systems*, 35:19810–19823.
- Xuan Zhang, Limei Wang, Jacob Helwig, Youzhi Luo, Cong Fu, Yaochen Xie, Meng Liu, Yuchao Lin, Zhao Xu, Keqiang Yan, et al. 2023. Artificial intelligence for science in quantum, atomistic, and continuum systems. *arXiv preprint arXiv:2307.08423*.

Appendices

A Example prompts

We show an example prompt for generating the labeling functions in Python code in Figure 5 and an example prompt for generating code snippet to call external tools in Figure 6.

We also show Prompts for concept generation and labeling on the FreeSolv dataset for GPT 3.5-Turbo in Figure 7.

B RQ1 supplement: the full version of concept refinement

In Figure 8, we show the full list of concepts selected by AutoMolCoon FreeSolv in three refinement iterations. The result corresponds to the RQ1.

C RQ5 supplement: BBBP linear model coefficients and ESOL intervention visualization

Coefficient Interpretation of Linear Models Additional to the results in Section 4.7, we evaluating the linear model on the BBBP dataset. We focused on the top positive coefficient lipophilicity ($\log P$) and the top negative coefficient hydrogen bond acceptors shown in Figure 9. Notably, $\log P$ has a coefficient of 1.97 and hydrogen bond acceptors has a coefficient -4.36 . These finding aligns with domain knowledge, as higher $\log P$ enhances a molecule’s ability to cross lipid-rich biological membranes. Conversely, a lower number of hydrogen bond acceptors generally enhances a molecule’s permeability through the BBB. These findings validate the CM’s alignment with established biochemical principles, demonstrating its potential utility in predictive modeling for molecular properties.

Concept Label Intervention Complementing to the intervention study in Section 4.7, we plot the results in Figure 10.

Coefficients of decision tree Complementing to the intervention study in Section 4.7, we plot the coefficients of decision tree in Figure 11.

D Ablation Studies

D.1 Different LLMs

We study the performance of AutoMolCo with different LLMs. Table 4 compares the performance of GPT-3.5 Turbo and Claude-2 using the direct LLM prompting labeling strategy with linear prediction models. While both GPT-3.5 Turbo and Claude-2 exhibit slightly inferior performance compared

LLM	FreeSolv(↓)	ESOL (↓)	BBBP (↑)	BACE (↑)
GPT-3.5	2.685	1.250	52.84	56.89
Claude-2	2.804	1.327	52.78	56.11

Table 4: Ablation on LLMs (GPT vs. Claude-2).

to GNNs across four datasets, they maintain competitive results, emphasizing simplicity and interpretability. Specifically, Claude-2 underperforms GPT-3.5 Turbo after first iteration, potentially due to its less consistent and accurate response. This inconsistency, partly attributed to more frequent issues with missing values and unit inconsistencies observed in Claude-2, suggests GPT-3.5 Turbo’s superior ability to generate reliable ground truth knowledge. Additionally, GPT-3.5 Turbo’s better prompt comprehension and domain knowledge in chemistry might contribute to its enhanced performance in predicting target concepts.

D.2 Direct LLM prompting with molecule names vs. with SMILES strings

Building on insights from (Guo et al., 2023b) regarding LLMs’ challenges with long molecular representations, we examined LLM’s capability in labeling concepts using SMILES strings and molecule names across our datasets. Our findings indicate that LLMs perform reasonably well in identifying basic concepts like molecular weight and atom counts using either molecule names or SMILES strings, with a strong correlation to ground truth labels ($r > 0.9$). However, LLMs struggle with complex concepts requiring detailed structural knowledge, such as the *number of chiral centers*. Moreover, our analysis reveals a notable decline in LLM’s performance with molecule names in the larger datasets like BBBP, suggesting LLM’s familiarity with common molecular names improves its performance on smaller datasets, but this advantage diminishes with less familiar names in larger datasets. In contrast, the structural specificity of SMILES strings maintains more consistent performance across dataset sizes, highlighting their utility in representing unique molecular concepts. Furthermore, we compared the performance difference between the two representations over 3 iterations. As demonstrated in Table 5, the model performance using SMILES strings matched the model performance using molecule names on most datasets, but a notable improvement in performance with SMILES strings is observed on the BBBP dataset.

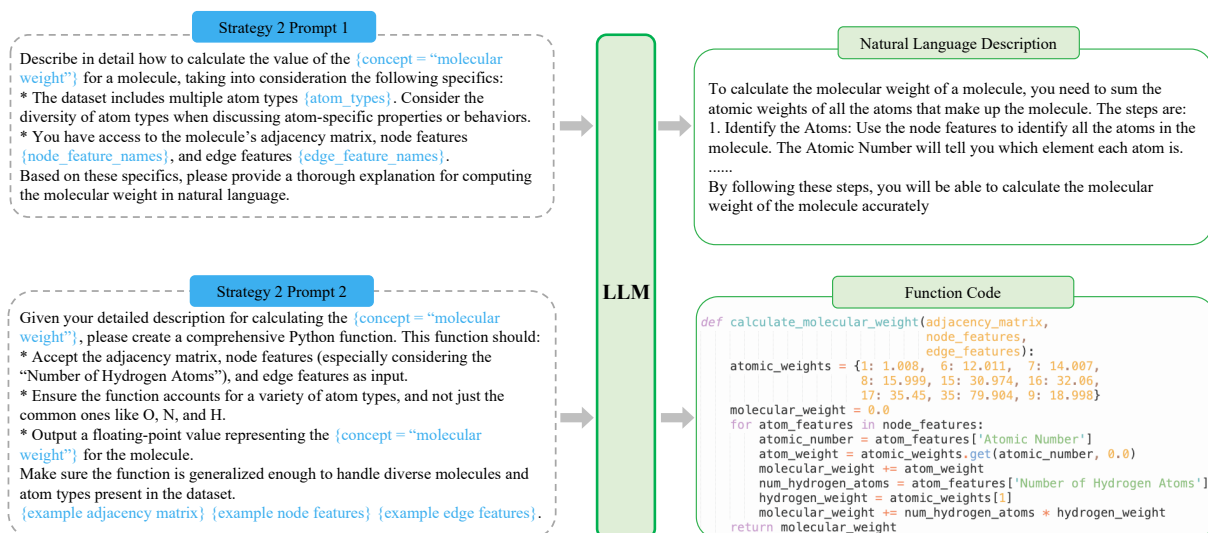


Figure 5: Prompts for generating concept labeling functions in Python code for the FreeSolv dataset.

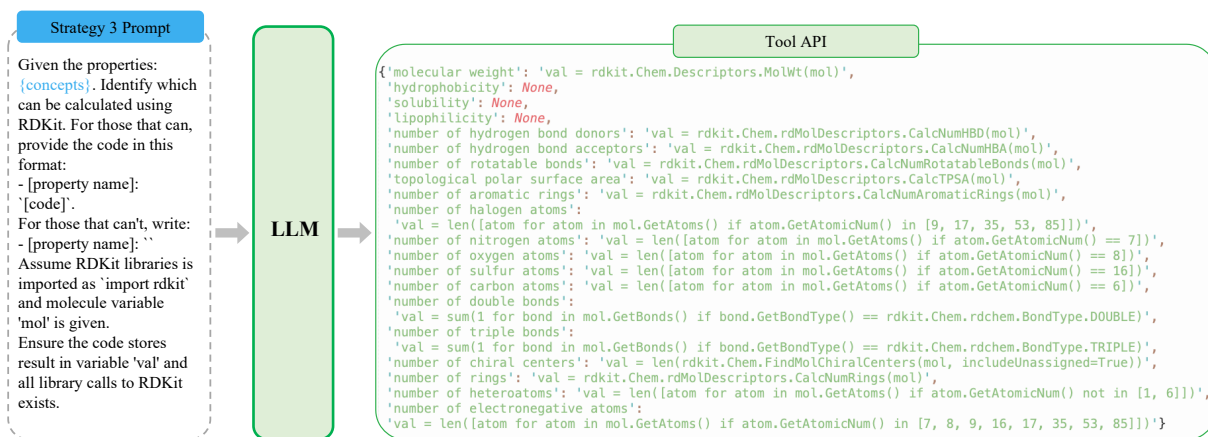


Figure 6: Prompts for calling the external tool RDKit to label concepts on the FreeSolv dataset.

We also present a comparative analysis of the quality of concept labels generated using molecular names vs. SMILES strings. The comparison is visualized through a series of heatmaps, as illustrated in Figure 12.

D.3 Combine different labeling strategies

The AutoMolCo framework includes three labeling strategies and allows easy extension to new ones. We consider combinations of the labeling strategies and study their impact on model performance, where we adopt a simple priority heuristic where strategy 3 > strategy 2 = strategy 1. Specifically, whenever the external tool is available for a suggested concept, we get the accurate concept labels from calling it. Otherwise, two concept labels are derived from both direct prompting and function code generation, and they are both considered in step 3 for the selection. This combined-strategy

labeling turns out to outperform most of the standalone strategies as shown in Table 8. These findings demonstrate that the three labeling strategies have their own strengths and weaknesses for different concepts, and they can be complement to each other to maximize the model performance. We leave exploration of new strategies and more sophisticated strategy combinations as future work.

E Accuracy comparison with LLM ICL on BBBP and BACE

In our experiments, we follow the standard and widely-used evaluation metrics for all datasets. For classification tasks on BBBP and BACE, we mainly evaluate these datasets using the AUC-ROC metric and report results in our main Table 1. Since (Guo et al., 2023b) provides the ICL prompts but evaluates BBBP and BACE with accuracy, we also report a comparison in accuracy in Table 6 for a

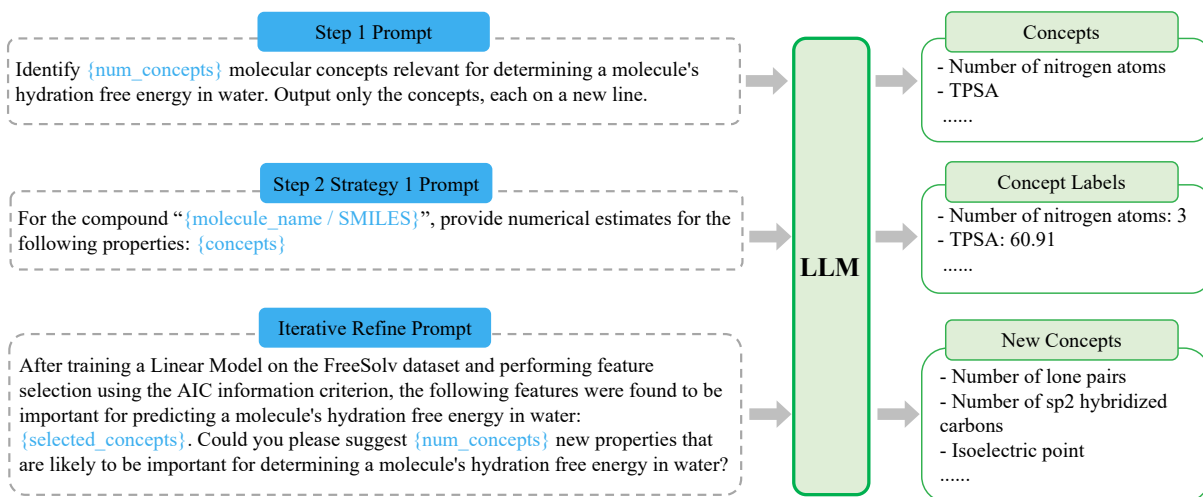


Figure 7: Prompts for concept generation and labeling on the FreeSolv dataset. Hyperparameters, molecule instance information, and re-used LLM responses from a previous step are in blue.

Input Format	FreeSolv (\downarrow)		ESOL (\downarrow)		BBBP (\uparrow)		BACE (\uparrow)
	SMILES	Names	SMILES	Names	SMILES	Names	SMILES
GPT-3.5 iter 1	2.854	2.685	1.401	1.250	53.88	52.84	56.89
GPT-3.5 iter 2	2.662	2.520	1.262	1.250	56.08	54.05	57.60
GPT-3.5 iter 3	2.763	2.520	1.262	1.255	60.41	56.08	58.38

Table 5: Ablation on input formats (SMILES vs. molecule names).

fair comparison.

F Decision trees visualization

As discussed in 4.7, we visualize the decision tree which makes the prediction process explainable. Figure 13 shows the impurity details of the decision tree shown in Figure 11 and Figure 14 shows a sample decision tree for the BACE dataset.

G More results on Buchwald-Hartwig and Suzuki-Miyaura

The GPT ICL performance from (Guo et al., 2023b) are measured on 100 data samples. To compare AutoMolCo's performance with their numbers, for each dataset we picked the best model performance from the logistic regression models or MLP models trained on either 200 or 500 sampled training data. The performance details are presented in Table 7 with best performance reported in Table 1.

	BBBP (\uparrow)	BACE (\uparrow)
GPT-4 (zero-shot)	0.476	0.499
GPT-4 (Scaffold, k= 8)	0.614	0.679
GPT-3.5 (Scaffold, k= 8)	0.463	0.496
Ours	0.657	0.704

Table 6: Performance comparison of the AutoMolCo-induced CM vs. LLM ICL (results are taken from (Guo et al., 2023b)). Results are evaluated with Accuracy (\uparrow).

	BH (\uparrow)	SM (\uparrow)
GPT-4 (random, k = 8)	0.800	0.764
GPT-3.5 + logistic (200 samples)	0.800	0.770
GPT-3.5 + MLP (200 samples)	0.800	0.780
GPT-3.5 + logistic (500 samples)	0.790	0.780
GPT-3.5 + MLP (500 samples)	0.810	0.800

Table 7: Performance comparison of the best AutoMolCo-induced CM vs. LLM ICL in accuracy (\uparrow) (GPT results are taken from (Guo et al., 2023b)).

Labeling Strategy	Model	FreeSolv(\downarrow)	ESOL (\downarrow)	BBBP (\uparrow)	BACE (\uparrow)
Direct Prompt	Linear/Logistic	2.685	1.250	52.836	56.894
Direct Prompt	Tree Model	2.791	1.272	56.887	68.632
Direct Prompt	MLP	2.338	1.194	51.794	60.059
Direct Prompt + Function	Linear/Logistic	2.697	1.254	56.134	56.712
Direct Prompt + Function	Tree Model	2.540	1.364	55.150	59.998
Direct Prompt + Function	MLP	2.211	0.971	57.697	65.797
Direct Prompt + Function + Tool	Linear/Logistic	3.002	1.136	55.845	64.032
Direct Prompt + Function + Tool	Tree Model	3.752	1.107	56.250	65.658
Direct Prompt + Function + Tool	MLP	2.122	0.791	58.391	62.624

Table 8: Combine labeling strategies.

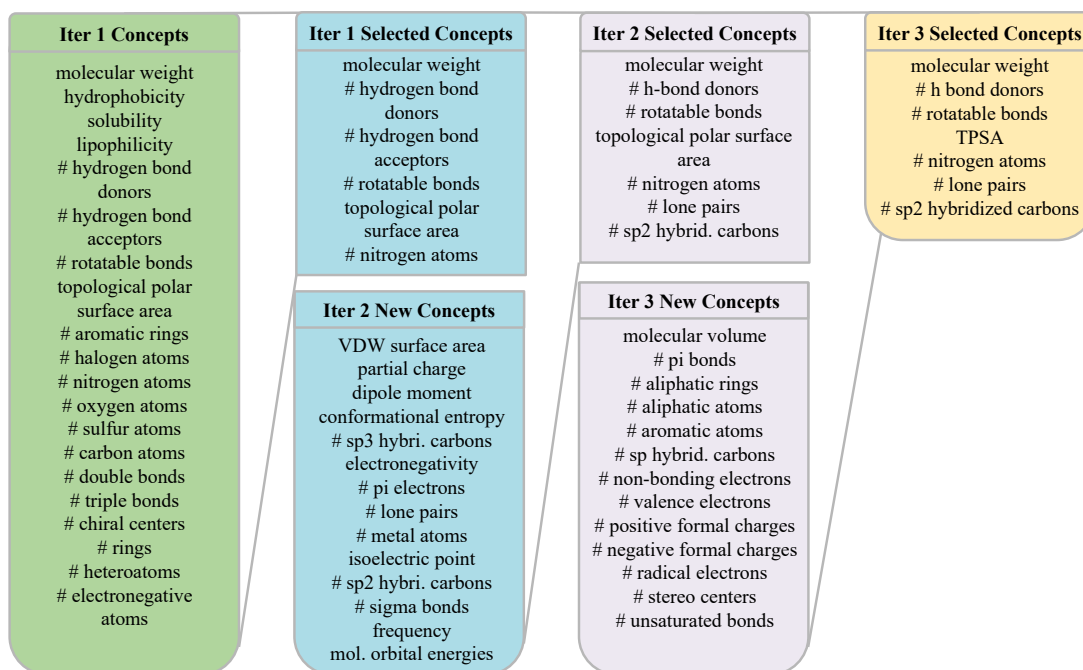


Figure 8: Concepts selected by AutoMolCo in three refinement iterations on FreeSolv. Full version.

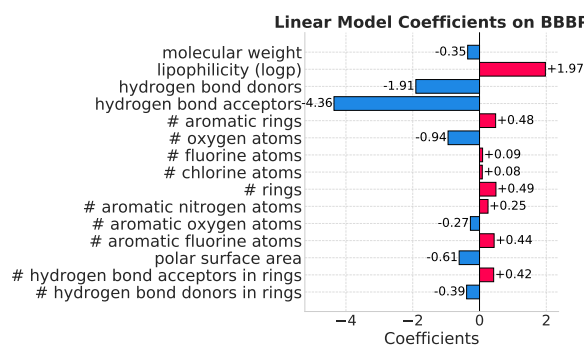


Figure 9: RQ5: Coefficients of the logistic regression model on BBBP with concepts refined by AutoMolCo after three iterations

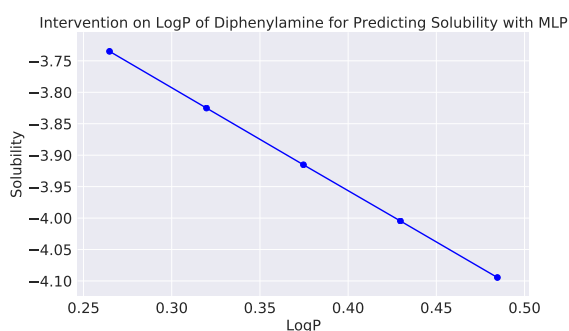


Figure 10: Intervention on logP of diphenylamine for predicting solubility with MLP

Decision Tree for BBBP

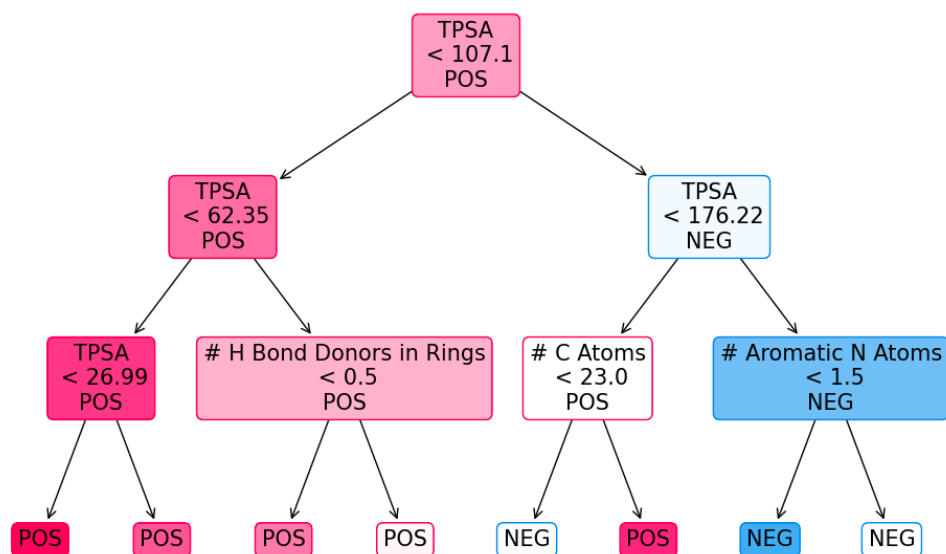


Figure 11: RQ5: coefficients of decision tree on BBBP from AutoMolCo after three iterations.



Figure 12: Correlation (r) between the ground truth labels and concept labels generated using molecule names or SMILES strings. Red indicates a higher correlation.

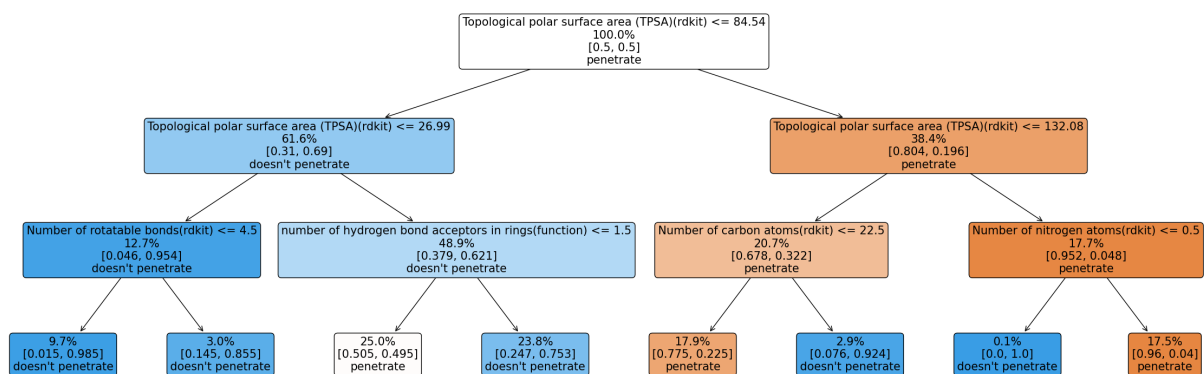


Figure 13: The decision trees for classification on BBBP.

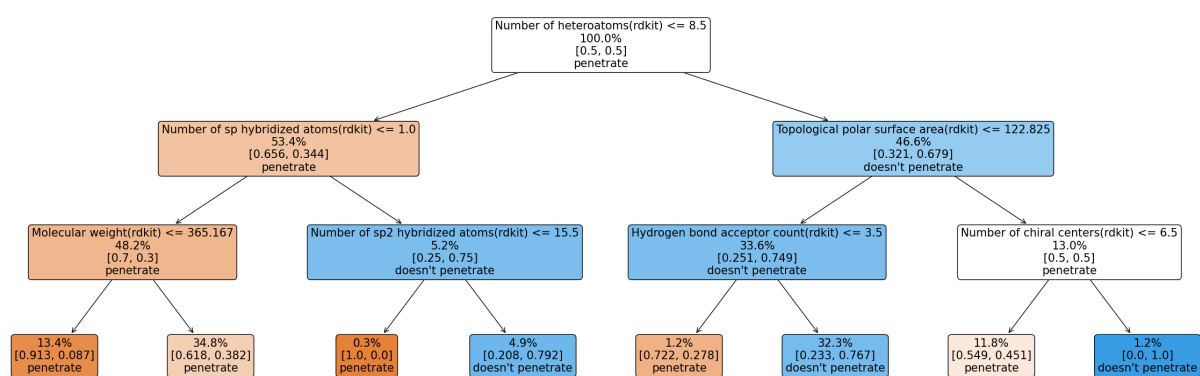


Figure 14: The decision trees for classification on BACE.