KGG: Knowledge-Guided Graph Self-Supervised Learning to Enhance Molecular Property Predictions

Van-Thinh To[1], Phuoc-Chung Van-Nguyen[1], Gia-Bao Truong[1], Tuyet-Minh Phan[1], Tieu-Long Phan[2,3*], Rolf Fagerberg[3], Peter F. Stadler[2,4,5,6,7,8], Tuyen Ngoc Truong[1*]

[1]Faculty of Pharmacy, University of Medicine and Pharmacy at Ho Chi Minh City, 41 Dinh Tien Hoang, District 1, Ho Chi Minh City, 700000, Vietnam.

[2]Bioinformatics Group, Department of Computer Science & Interdisciplinary Center for Bioinformatics & School for Embedded and Composite Artificial Intelligence (SECAI), Leipzig University, Härtelstraße 16–18, D-04107 Leipzig, Germany.

[3]Department of Mathematics and Computer Science, University of Southern Denmark, DK-5230 Odense M, Denmark.

[4]Max Planck Institute for Mathematics in the Sciences, Inselstraße 22, D-04103 Leipzig, Germany.

[5]Department of Theoretical Chemistry, University of Vienna, Währingerstraße 17, A-1090 Wien, Austria.

[6]Facultad de Ciencias, Universidad National de Colombia, Bogotá, Colombia.

[7]Center for non-coding RNA in Technology and Health, University of Copenhagen, Ridebanevej 9, DK-1870 Frederiksberg, Denmark.

[8]Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, NM 87501, USA.

*Corresponding author(s). E-mail(s): long.tieu_phan@uni-leipzig.de; truongtuyen@ump.edu.vn;
Contributing authors: tvthinh@ump.edu.vn; nvpchung.d19@ump.edu.vn; tgbao.d18@ump.edu.vn; tuyetminh.work@gmail.com; rolf@imada.sdu.dk; studla@bioinf.uni-leipzig.de;

Abstract

Molecular property prediction has become essential in accelerating advancements in drug discovery and materials science. Graph Neural Networks have recently demonstrated remarkable success in molecular representation learning; however, their broader adoption is impeded by two significant challenges: (1) data scarcity and constrained model generalization due to the expensive and time-consuming task of acquiring labeled data, and (2) inadequate initial node and edge features that fail to incorporate comprehensive chemical domain knowledge, notably orbital information. To address these limitations, we introduce a Knowledge-Guided Graph (`KGG`) framework employing self-supervised learning to pre-train models using orbital-level features in order to mitigate reliance on extensive labeled datasets. In addition, we propose novel representations for atomic hybridization and bond types that explicitly consider orbital engagement. Our pre-training strategy is cost-efficient, utilizing approximately 250,000 molecules from the ZINC15 dataset, in contrast to contemporary approaches that typically require between two and ten million molecules, consequently reducing the risk of potential data contamination. Extensive evaluations on diverse downstream molecular property datasets demonstrate that our method significantly outperforms state-of-the-art baselines. Complementary analyses, including `t-SNE` visualizations and comparisons with traditional molecular fingerprints, further validate the effectiveness and robustness of our proposed `KGG` approach.

Keywords: Drug discovery, graph neural networks, knowledge graph, self-supervised learning, orbital information.

1 Introduction

Significant advancements in Artificial Intelligence (AI) have profoundly influenced drug discovery sector, primarily through the implementation of machine learning and deep learning techniques. These computational methods, renowned for their ability to process and analyze large volumes of data with remarkable speed and precision, have demonstrated significant potential to enhance efficiency and reduce costs across various stages of the drug development pipeline, such as uncovering drug-target interactions [1, 2], designing and optimizing drug structures [3–5], and predicting 3D structures of proteins [6]. The identification of molecules with desired properties, in particular bioactivity and toxicity, remains one of the major interests in the field of Computer-Aided Drug Design (CADD) and Drug Development [7–9].

Molecular representations play a pivotal role in accurately predicting molecular properties, serving as the foundation for computational models to capture the essential structural and chemical features of molecules [10–12]. The choice of molecular representation profoundly affects the performance of predictive algorithms, as a well-chosen representation helps ensuring that the model can generalize across diverse chemical spaces while maintaining interpretability. Molecular descriptors [13–15], including chemical and physical properties of compounds, and molecular fingerprints [16–18] that encode the structure and properties of molecules into binary vectors are frequently used as input features in predictive models.

Graph-based deep learning has garnered significant attention within the artificial intelligence community [19–22], driven primarily by the ubiquity of graph-structured data across various domains, including e-commerce [23], transportation [24], and chemistry [25]. Chemical structures inherently adopt graph representations, making graph-based models particularly promising for molecular representation learning [26, 27]. Despite their notable successes in supervised and semi-supervised learning scenarios, these models heavily depend on manually labeled data, leading to several limitations: (1) the acquisition and annotation of large-scale labeled datasets can be prohibitively expensive, particularly in specialized fields such as chemistry and medicine [28], as well as in fields where datasets are very extensive, such as in the study of social and citation networks [29]; (2) supervised models frequently suffer from limited generalization and increased susceptibility to overfitting, particularly when labeled data is scarce [30]; and (3) the accuracy and reliability of labels significantly affect model performance, making these methods vulnerable to label noise and uncertainty [31]. These inherent challenges highlight the necessity for developing alternative methodologies capable of reducing dependency on labeled data while maintaining robust and generalizable performance.

Despite the persistent scarity of labeled data, the abundance of large-scaled unlabled dataset, particularly in chemistry [32], represents an invaluable resource, contingent upon effective utilization. Self-Supervised Learning (SSL) emerges as a promising paradigm in situations where extensive unlabeled data, but only limited labeled data, exist. In practice, SSL models are pre-trained using sizable unlabeled datasets through various pretext tasks, thereby capturing general representations of the underlying data manifold. Subsequently, these pre-trained models undergo fine-tuning using much smaller labeled datasets to optimize task-specific performance. Recent investigations into SSL methodologies, for molecular representation learning [33–41], which are summarized in Supporting Section 2.2, have demonstrated impressive performance and robustness across diverse benchmark datasets, detailed in Tables S1 and S2. Notably, the *Hierarchical Molecular Graph Self-Supervised Learning (HiMol)* framework introduced by Zang et al. [33], which leverages hierarchical graph structures to facilitate integrated representation learning, and the *Knowledge-guided Pre-training of Graph Transformer (KPGT)* model by Li et al. [41], which employs line graph representations complemented by a central node aggregating and propagating chemical properties, have been identified as state-of-the-art approaches exhibiting remarkable robustness and predictive power.

Although such significant advancements have been achieved in molecular representation learning, several critical challenges persist: (1) molecular graphs predominantly encode atom-level information within nodes and edges but generally omit orbital details, which are a crucial factor underlying chemical valence bonds, as well as other chemical properties, thereby limiting representation expressivity; (2) one-hot encoding strategies are computationally demanding for large-scale datasets [42] and insufficient in capturing meaningful relations among categorical variables [43], such as bond types or orbital hybridization; (3) the excessive reliance on the pre-training datasets utilized by SSL models may lead to data contamination [44]. Such contamination can inadvertently incorporate test instances into the training process, thereby artificially enhancing the measured generalization performance.

To address the aforementioned challenges, we introduce a novel graph-based SSL framework, Knowledge-Guided Graph (KGG), which explicitly integrates orbital information into molecular graphs. Our approach is inspired by the work of Benkö et al. [45], which employed *orbital graphs* where hybridization-aware atomic orbitals are represented as vertices and where the corresponding edges, interpreted as orbital overlaps, depict localized chemical bonds responsible for chemical reactions. The proposed KGG model consists of two essential components: (1) the Knowledge Representation Graph (KRG) architecture, serving as an encoder to extract hierarchical graph representations enriched with orbital-level chemical insights; and (2) the Knowledge Self-Supervised Pre-training (KSSP) multi-task pretext module, designed for comprehensive pre-training. This module encompasses tasks ranging from adjacency matrix reconstruction to the prediction of molecular attributes, notably orbital hybridization and bond characteristics. Taken together, this enables KGG to address the abovementioned drawbacks of coventional molecular graph represenations. We furthermore restrict our pre-training data to mitigate the risk of data contamination, as suggested by Jiang et al. [44], thereby ensuring more robust capabilities across diverse datasets.

2 Results and Discussion

2.1 KGG Framework

The KGG model is an SSL framework designed to encode molecular representations by utilizing knowledge vectors as initial features (Figure 1). Its architecture comprises two primary components: the KSSP and the KRG. The KSSP operates as a pre-training decoder driven by pretext tasks aimed at reconstructing orbital knowledge vectors, adjacency matrices, and two fundamental molecular properties, using a Multi-Layer Perceptron (MLP) that receives graph embeddings extracted by KRG. By jointly optimizing these tasks, the model effectively captures rich orbital and molecular information, allowing the KRG encoder to learn meaningful representations that enhance the overall predictive performance (Figure 1c). The KRG, built upon the Graph Isomorphism Networks (GIN) architecture [19], functions as a graph embedding extractor by processing hierarchical graphs and incorporating *orbital* knowledge vectors as inputs to produce graph embeddings (Figure 1d).

2.2 Molecular Property Predictions

Our KGG model demonstrated superior performance over thirteen state-of-the-art (SOTA) SSL approaches, details of which are provided in Supporting Section 2.1 and 2.2 for classification and regression tasks on datasets from MoleculeNet, respectively.

Regarding the classification tasks, our KGG model attained the highest average ROC-AUC score (75.5 ± 0.3) across six datasets, as depicted in Figure 2a. Detailed

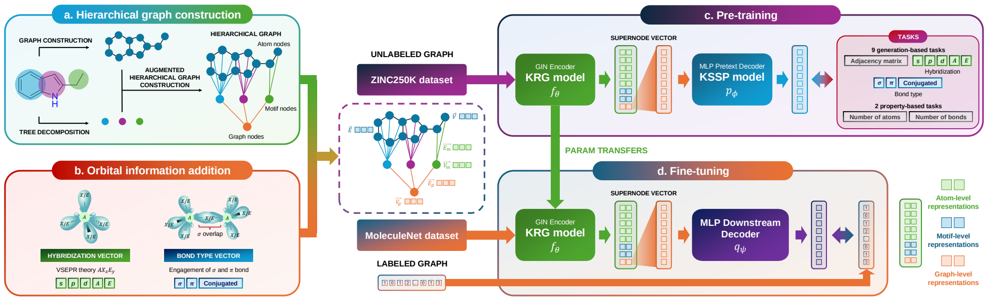


Fig. 1 Overview of the KGG Framework. (a) The hierarchical molecular graph construction involves three critical steps: (1) establishing an atom-level foundational graph, (2) decomposing this graph into motif-level nodes, and (3) integrating a supernode that encapsulates the entire molecular structure, thereby creating a comprehensive hierarchy with atom-, motif-, and graph-level representations [33]. **(b)** Initial feature embedding highlights the unique integration of orbital-level chemical information. Atom features are derived based on *hybridization* and Valence Shell Electron Pair Repulsion (VSEPR) theory [46], while bond features distinctly represent relationships characterized by the count of σ and π bonds, and existence of conjugation system (δ). Training procedure of KGG comprises two stages: pre-training **(c)** with eleven pretext tasks in the KSSP module, followed by fine-tuning **(d)** of the pre-trained KRG model for downstream evaluations on MoleculeNet benchmarks [47].

results, obtained from three independent experiments using distinct random seeds, are presented in Table S1. We further validated the superior performance of KGG through statistical testing, *T-test* in particular, which confirmed a statistically significant improvement (p-value < 0.01) relative to the second-best method (see Figure 2d). For a more comprehensive view of individual dataset performance and corresponding statistical assessments, we refer to Figures S1 and S2, respectively. Specifically, KGG delivered superior results on BACE and SIDER, with ROC-AUC scores of 86.3 ± 0.2 and 64.9 ± 1.0, respectively. On BACE, KGG significantly outperformed the second-best model (HiMol, 84.3 ± 0.3) at a p-value < 0.001. Likewise, on SIDER, KGG (64.9 ± 1.0) significantly exceeded the second-best approach (MGSSL, 61.8 ± 0.8) with p-value < 0.05. Although KGG ranked second on BBBP (72.5 ± 0.7) after HiMol (73.2 ± 0.8), the difference did not reach statistical significance (p-value > 0.05). Overall, these findings highlight the predictive capability of KGG compared to extant SSL methods, underscoring the practical utility of orbital information in molecular representation learning. Compared to the previously state-of-the-art HiMol [33], KGG achieved an average ROC-AUC improvement of 2.4% and outperformed in four out of six datasets (BACE, SIDER, ToxCast, ClinTox).

Among the spectrum of high-performing SSL methods, motif-based architectures, including KGG, HiMol [33], G_Motif [34], and MGSSL [36], have consistently exhibited superior predictive ability. Indeed, five of the six top-performing models in our experiments employ explicit motif representations, thereby emphasizing the pivotal role of motif information in molecular representation learning. Motifs, conceived as functional fragments encapsulating key chemical features, enable these models to detect and exploit domain-specific patterns inherent in molecular structures. Within this elite subset of motif-based approaches, KGG surpassed its counterparts in four of the six benchmark datasets, underscoring the utility of explicitly incorporating orbital information into motif-based designs. Departing from previous SSL paradigms [33–36, 40, 41, 48–54], KGG encodes molecular representations via a KRG "backbone," which merges motif structures with orbital knowledge vectors in the initial feature space for both nodes and edges. These enhanced descriptors, incorporating hybridization states and bond types (see Section 4.2), strengthen the model's expressive power, ultimately leading to more accurate molecular structure learning and property predictions.

Turning to regression tasks, Figures 2b, c illustrate the average root mean square error (RMSE) and mean absolute error (MAE) achieved across six datasets in MoleculeNet. As recommended by MoleculeNet [47], MAE serves as the evaluation metric for quantum-mechanical datasets (QM7, QM8, QM9), while RMSE is used for physicochemical datasets (ESOL, FreeSolv, Lipophilicity). Table S2 and Figure S3 provide a detailed comparison between KGG and other SSL architectures on each respective regression dataset. Notably, KGG delivered the best overall performance, achieving an average RMSE of 0.78 ± 0.15 and an average MAE of 27.076 ± 43.86. While we do not report T-test results here due to the limited number of datasets (three in each category), KGG nevertheless outperformed its peers on ESOL, QM8, and QM9, achieving scores of 0.731, 0.665, and 77.684, respectively (see Figure S3).

Compared to HiMol, which is the prior SOTA model, KGG reduced the average RMSE by 39% and outperformed on two out of three individual RMSE benchmarks. The average MAE was likewise reduced by 14%, with KGG again surpassing HiMol on two of the three MAE datasets. Notably, KGG attained a 68% lower RMSE than HiMol on FreeSolv and yielded a 15% reduction in MAE on QM7, a quantum-mechanical dataset often considered among the most challenging (see Table S2). These findings underscore the effectiveness of integrating orbital knowledge with motif-centric representations, thereby enabling KGG to capture intricate chemical relationships and deliver robust predictive performance.

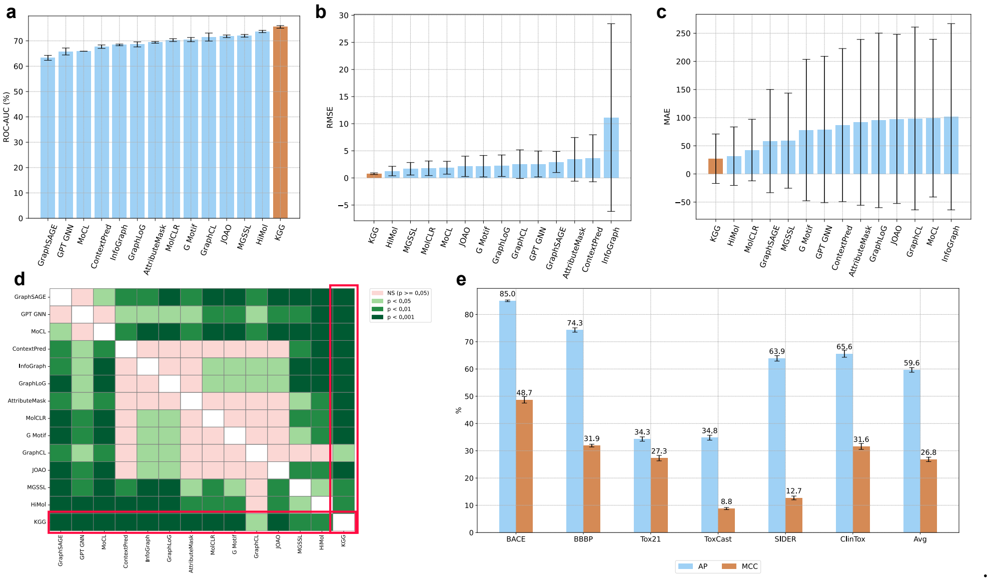


Fig. 2 **(a)** The average results of KGG and other SSL approaches on six classification datasets from MoleculeNet is measured in terms of ROC-AUC (%). The results, presented as mean ± standard deviation, were obtained through three independent runs, each utilizing a different random seed. **(b)** The average RMSE values across three regression physicochemical datasets ESOL, FreeSolv, and Lipophilicity of KGG model and previous SSL studies. **(c)** The average MAE values on three regression quantum datasets QM7, QM8, QM9 of KGG model and other SSL methods. **(d)** The T-test analysis compares the performance of KGG model against other SSL methods upon six classification datasets. The T-test results indicate that the performance of our KGG model is statistically significant superior to other SSL methods, with p value at three levels: 0.05, 0.01 and 0.001. **(e)** The Average Precision (AP) and Matthews Correlation Coefficient (MCC) values on six classification datasets from MoleculeNet of KGG model.

2.3 Comparisons with traditional fingerprints

We performed a comparative analysis to evaluate the representational capacity of the KGG fingerprint, derived from the *graph-level representations* produced by the KGG model, against three widely used conventional fingerprints: MACCS, ECFP4, and RDK7. Our evaluation revealed two key findings: firstly, a kNN classifier, selected for its simplicity, trained on KGG fingerprints, consistently achieves higher predictive accuracy compared to classifiers utilizing conventional fingerprints; secondly, t-distributed stochastic neighbor embedding (t-SNE) visualization clearly demonstrates that KGG fingerprints yield significantly improved clustering of data points, highlighting their enhanced discriminative ability.

Specifically, KGG fingerprints outperform traditional fingerprints on the BACE and BBBP datasets for *classification* tasks (Figure 3a) as well as on the ESOL, FreeSolv, Lipophilicity, and QM7 datasets for *regression* tasks (Figure 3b). Importantly, these improvements are consistently observed under both *random* and *scaffold* splitting strategies, indicating that the observed superiorities are fundamentally attributable to our knowledge vectors, particularly the incorporation of orbital information. Moreover, the notable predictive performance achieved even with a simple algorithm such as kNN suggests that KGG fingerprints inherently capture richer and more chemically meaningful information than traditional fingerprints, independent of the complexity of the predictive model. This inference is further substantiated by comparable predictive results when employing either kNN or MLP classifiers on KGG fingerprints, as presented in Figure S4.

Furthermore, we applied t-SNE visualization to directly compare KGG fingerprints with conventional alternatives, as illustrated in Figure 3c. The visualization distinctly demonstrates superior class separation in the BACE dataset when using KGG fingerprints, as opposed to conventional fingerprints. A similar pattern is consistently observed across other datasets, as further detailed in Figure S5, S6, S7, S8, S9. This clear separation reaffirms the strong capability of our KGG model in generating effective graph neural network fingerprints for molecular representation, enabling various downstream tasks.

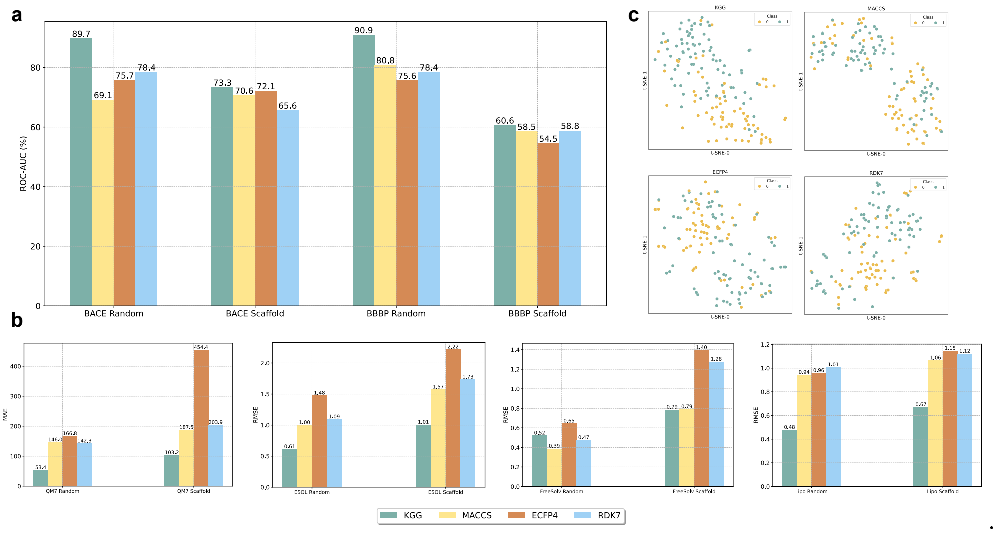
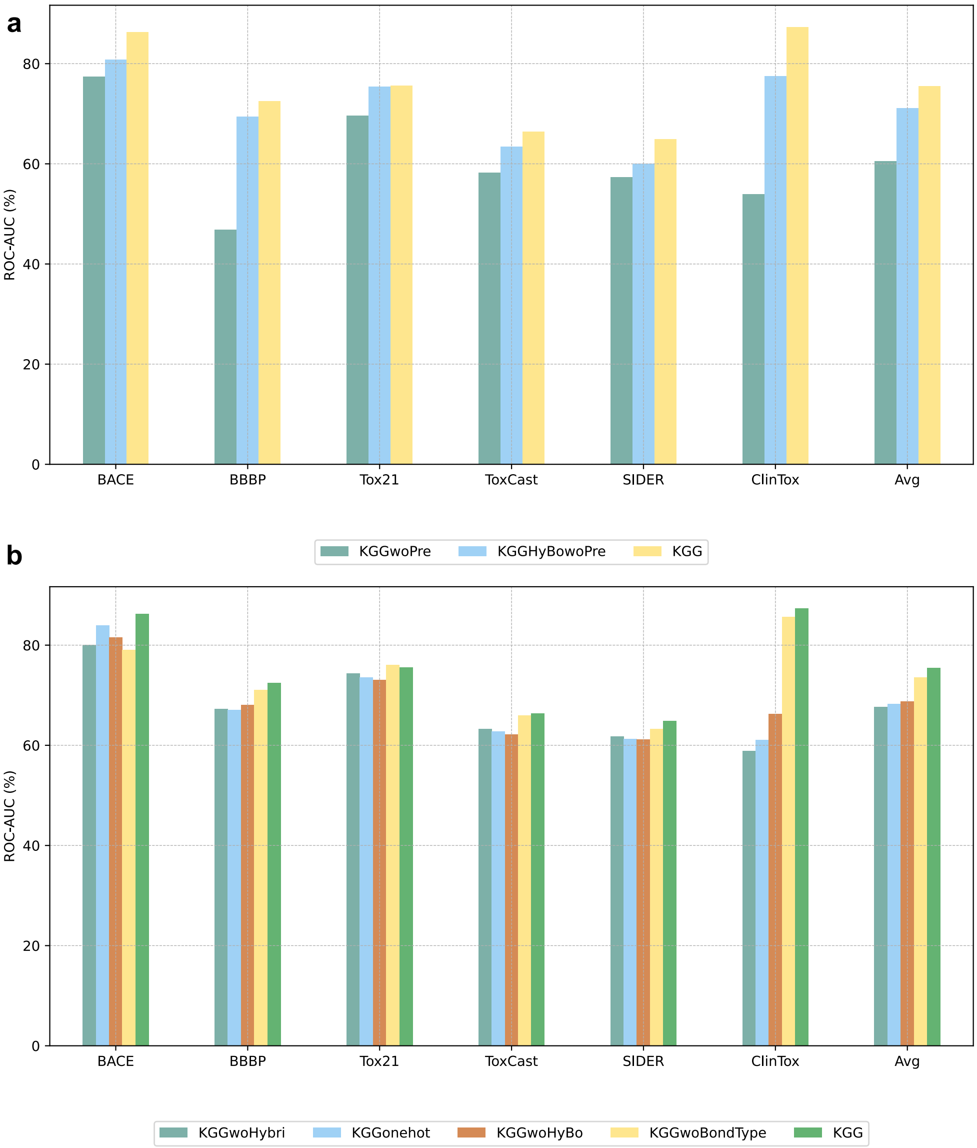


Fig. 3 A comparison of classification and regression performance of KGG, MACCS, ECFP4, RDK7 fingerprints using random and scaffold splitting strategies, as well as t-SNE visualizations. **(a)** The ROC-AUC values for two classification datasets (BACE and BBBP). **(b)** RMSE and MAE values for four regression datasets (ESOL, FreeSolv, Lipophilicity, QM7). **(c)** A t-SNE visualization of the validation BACE test set for each fingerprint type.

2.4 Ablation Study

To assess the effectiveness of each component in our proposed KGG model, we performed a series of ablation experiments. By selectively removing key modules from the KGG architecture, we created several model variants, as detailed in Supporting

Section 3.3, that highlight the contribution of each component [55]. The outcomes of these experiments are summarized in Figure 4, Table S4, and Table S5.



.

Fig. 4 Ablation study of KGG variants. **(a)** ROC-AUC values for six classification datasets across different KGG pre-training configurations, with the final three columns indicating their overall average. **(b)** performance of six classification datasets under KGG variants that use different featurization approaches, where the last five columns show the average ROC-AUC values for these variants.

We performed a detailed investigation of the impact of the pre-training on the performance of our KGG model, as depicted in Figure 4a. Specifically, we compared three configurations: without pre-training (KGGwoPre), with partial pre-training excluding orbital-oriented tasks (KGGHyBowoPre, retaining only reconstruction tasks for adjacency matrices, number of atoms, and number of bonds), and the fully pre-trained KGG. Our results indicate that KGGwoPre consistently yields the lowest performance across six classification datasets, underscoring the necessity of effective pre-training for robust molecular representations. Meanwhile, KGGHyBowoPre improves significantly over KGGwoPre, yet the complete KGG model demonstrates superior performance, emphasizing the critical contribution of orbital-oriented tasks. Thus, integrating these tasks in self-supervised learning enhances weight initialization and facilitates more effective model adaptation.

To evaluate the influence of graph representations on downstream fine-tuning performance, we systematically examined four variants of the KGG model (Figure 4b): KGGonehot (utilizing only one-hot encodings), KGGwoHyBo (omitting both hybridization and bond type vectors), KGGwoBondType (omitting bond type vectors), and KGGwoHybri (omitting hybridization vectors). Our analyses underscore the pivotal role of atomic hybridization, evidenced by KGGwoBondType's consistent outperformance over KGGwoHybri, thereby highlighting that *hybridization* provides more predictive utility compared to bond type alone, which may introduce unnecessary noise. Notably, the complete KGG model, incorporating both *hybridization* and *bond type* vectors, demonstrated superior performance relative to both KGGwoBondType and KGGwoHybri. This finding aligns chemically with the fundamental principle that molecular bonding involves hybrid orbital overlaps, underscoring the synergy between *hybridization* and *bond type* descriptors. Additionally, reliance exclusively on simplistic categorical encodings (KGGonehot) resulted in suboptimal performance compared to other models, highlighting their inadequacy in capturing nuanced chemical information. Ultimately, our fully-integrated KGG model, combining comprehensive orbital-related features, consistently delivered the strongest performance, validating the significant advantages of embedding detailed chemical knowledge into molecular graph representations.

2.5 Data contamination

Data contamination has artificially boosted the performance of SSL models on downstream tasks [44]. This phenomenon arises because the model merely memorizes the structures of test sets within the pre-training dataset, rather than exhibiting robust generalization. Our KGG model outperformed other SSL methods while maintaining a low rate of data contamination and utilizing a significantly smaller pre-training dataset comprising 250,000 samples, in contrast to the 2 to 10 million data points employed by others.

To investigate this issue, we selected the KPGT model [41], which has exhibited superior performance across several MoleculeNet datasets, as a benchmark for assessing the extent of contamination. Our findings, summarized in Table 1 and Figure 5, reveal two principal observations: (1) KGG exhibits an exceptionally low contamination rate (0.2%), contrasting sharply with the substantially higher rate observed in KPGT (81.7%); (2) despite this significantly lower contamination, KGG achieves competitive performance on diverse tasks, including classification (e.g., BACE, SIDER, ClinTox; see Figure 5b) and regression (e.g., ESOL, FreeSolv, Lipo; see Figure 5a). Notably, even when KPGT contamination is relatively moderate, such as 37.8% in ClinTox or 19.7% in BACE, the KGG model consistently matches or surpasses KPGT's performance. This observation suggests that KPGT's superior results on certain datasets may be influenced by memorization of test-set structures during pre-training, undermining fair comparative evaluation. Consequently, our results demonstrate that KGG maintains robust generalization capability and competitive downstream performance, despite a markedly lower level of dataset contamination.

Table 1 Data contamination comparison between KGG and KPGT models. (↓) denotes that a smaller number is better.

Task type	**Contaminated Data Points**		**Test Data Points**		**Cont. ratio (%) (↓)**	
	KGG	KPGT	KGG	KPGT	KGG	KPGT
Classification	6	1799	2289	2289	**0.3**	78.6
Regression	0	559	598	598	**0**	93.5
Overall	6	2358	2887	2887	**0.2**	81.7

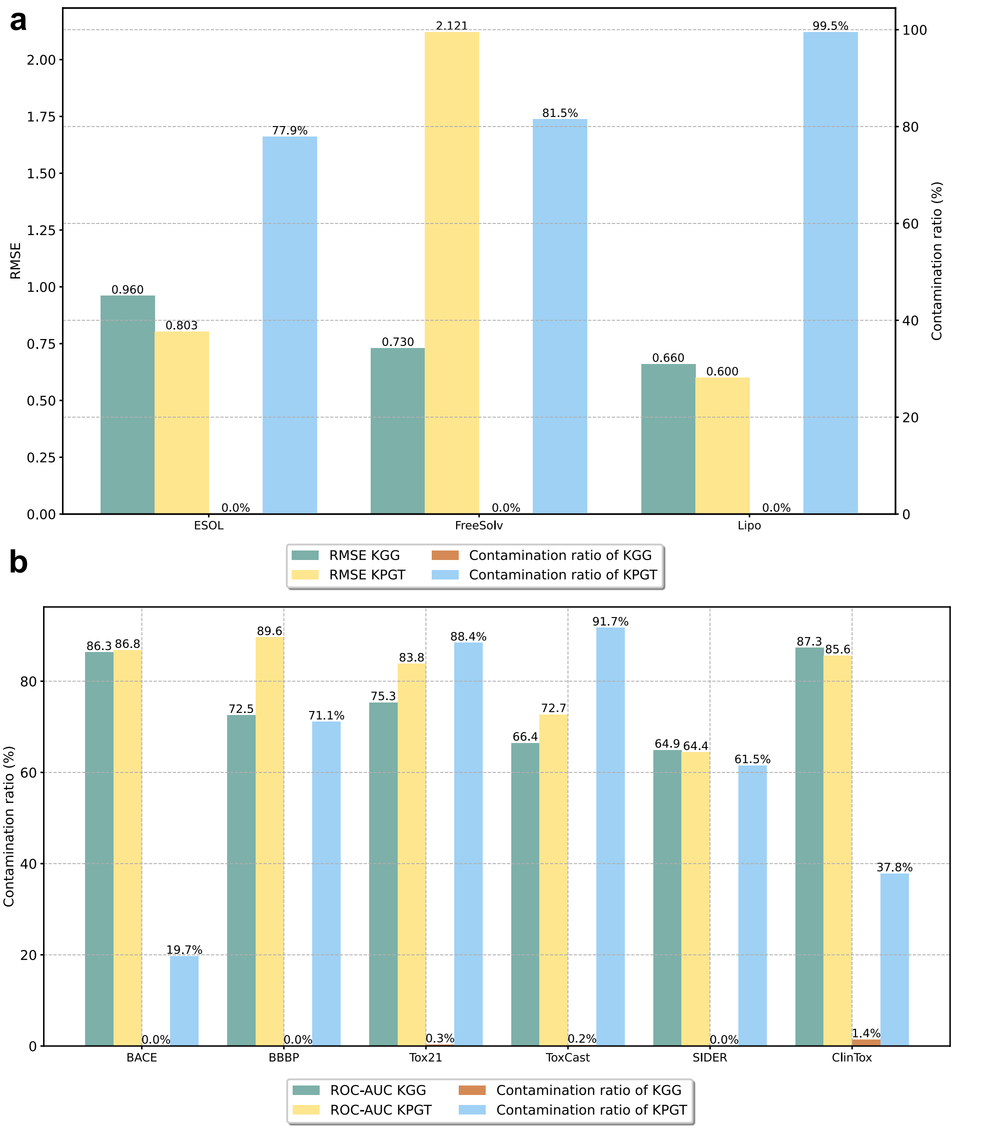


Fig. 5 An analysis of data contamination for KGG and KPGT is presented. **(a)** The RMSE performance of the two models is assessed on three regression datasets, also including the corresponding contamination ratios. **(b)** The ROC-AUC performance of both models is evaluated on six classification datasets, with their respective contamination ratios reported alongside.

2.6 Limitations of Benchmarking Metrics

Binary classification constitutes a fundamental task of molecular property prediction in general, and SSL in particular. Here, the area under the ROC curve (ROC-AUC) has conventionally been employed as the primary evaluation metric, recommended by MoleculeNet [33–36, 40, 41, 48–54]. Despite its widespread adoption, ROC-AUC captures only sensitivity and false positive rate, neglecting positive and negative predictive values. This can lead to misleadingly optimistic performance estimates, particularly in the context of class-imbalanced datasets. Furthermore, an individual coordinate in the ROC space does not uniquely specify a confusion matrix nor a set of matrices with an equivalent MCC [56, 57], thus raising additional concerns about the reliability of ROC-AUC for benchmarking SSL classification models. In contrast, the MCC accounts for all four quadrants of the confusion matrix (i.e., sensitivity, specificity, precision, and negative predictive value), thus requiring the classifier to excel across multiple dimensions in order to achieve a high MCC score. Indeed, while a high MCC value (e.g., 0.9) is invariably associated with a high ROC-AUC, the reverse does not necessarily hold [58]. The discrepancy between these metrics is evident in Figure 2e, where the MCC for KGG remains below 50% on certain datasets, whereas its ROC-AUC values consistently surpass 60% (Figure 2a). For imbalanced binary classification tasks, the average precision (AP) metric further complements MCC by jointly considering recall and precision [59]. Recognizing the inherent limitations of ROC-AUC and the need for more robust performance measures, we benchmarked KGG using both MCC and AP (Figure 2e). This approach represents the first instance of a SSL framework advocating MCC and AP as primary metrics for molecular classification, thereby offering a more comprehensive and reliable assessment of classification performance. Although the adoption of new metrics can introduce reproducibility challenges in large-scale benchmarking, we encourage future studies to consider MCC and AP alongside ROC-AUC to ensure robust and meaningful evaluations. To facilitate future comparative analyses, we present the MCC and AP performance of KGG in Table S3, thereby encouraging broader adoption of more robust and informative evaluation metrics in molecular classification tasks.

3 Conclusions

In this study, we propose an extended graph representation that explicitly incorporates orbital information, demonstrating superior performance over conventional approaches across a range of chemical tasks within a SSL framework. By embedding orbital characteristics into molecular graphs, our proposed methodology effectively captures detailed structural and chemical nuances, thereby surpassing conventional state-of-the-art SSL techniques relying solely on one-hot encodings. The principal contributions of this research are: (1) introducing two chemically informed knowledge vectors — hybridization states and bond types to enrich molecular representations; (2) designing a novel pretext task focused on reconstructing the orbital information encapsulated in these vectors; and (3) conducting a data contamination analysis to evaluate the generalizability and robustness of the proposed model.

Despite the encouraging performance of KGG in molecular property prediction, certain limitations warrant further exploration. First, the framework currently lacks an analysis of model uncertainty, which future studies should incorporate to quantify prediction confidence more effectively. Second, while chemical knowledge was utilized to encode select features, critical descriptors such as atom types remain represented via one-hot encoding, yielding high-dimensional feature spaces prone to overfitting. Addressing this challenge, subsequent research should focus on systematically transforming chemical domain knowledge (such as periodic groupings and periods, lone-pair contributions in conjugated systems, electron dynamics, and electron density distributions) into graph-structured representations, thereby enhancing both interpretability and predictive performance. Moreover, pre-training tasks are pivotal for the success of downstream applications. Accordingly, the selection of samples in the pre-training dataset must be designed not only to capture the intrinsic diversity of the data but also to prevent the introduction of bias or contamination, which can undermine generalization.

4 Methods

The overall framework of KGG is depicted in Figure 1 and comprises two distinct training stages: (1) pre-training with KSSP and (2) fine-tuning with KRG. Both stages follow a similar processing pipeline: (1) construction of a hierarchical graph (see Section 4.1); (2) integration of atom-level knowledge vectors (see Section 4.2); and (3) application of a representation extractor to encode the hierarchical graph into a feature vector (see Section 4.3). The resulting representation is then employed during both the pre-training and fine-tuning phases.

4.1 Hierarchical graph

4.1.1 Hierarchical structure

Given a SMILES string, a molecular graph $G = (V, E)$ is constructed using RDKit [60]. The vertex set V is defined as: $V = \{v_1, v_2, \ldots, v_n\}$, where n is the number of atoms in the molecule. The edge set E is given by pair of atoms (v_i, v_j) such that v_i and v_j connected by a chemical bond.

Subsequently, KGG identifies chemically meaningful substructures (motifs) as subgraphs $M'_i = (V'_i, E'_i)$ of G. The motif decomposition makes use of BRICS, a system of drug-like chemical fragments [61], and an additional rule from HiMol [33]. The decomposition will select a vertex set of G that matches the BRICS and HiMol rules. Complete details on the motif decomposition procedure are provided in Supporting Section 3.1 and Algorithm S1. Each motif M_i is associated with a motif node $v_{M_i} \in V_m$. Motif-atom edges E_m link motif nodes v_{M_i} to their constituent atom nodes v_j, i.e., we introduce the edge (v_{M_i}, v_j) whenever $v_j \in V(M_i)$ where $V(M_i)$ is the set of atoms forming the motif subgraph M_i.

To incorporate global structural information, a *graph-level node* or *supernode* $v_g \in V_g$ is introduced. Thus, the augmented hierarchical graph $\tilde{G}$ consists of three layers (*atom-level*, *motif-level*, *graph-level*) and is formally represented as: $\tilde{G} = (\tilde{V}, \tilde{E})$ with $\tilde{V} = V \cup V_m \cup V_g$ and $\tilde{E} = E \cup E_m \cup E_g$. The edges E_g connect the supernode to all motif nodes. Figure 6a illustrates the hierarchical graph construction process.

4.1.2 Hierarchical encoding

Let V, V_m, and V_g denote the sets of atoms, motifs, and the supernode, respectively. The *full representation* H^0 of the hierarchical graph is defined as the concatenation of the atom-level, motif-level, and graph-level feature representations: $H^0 = H \| H_m \| H_g$, where $\|$ denotes the vertical concatenation of matrices and vectors. The individual feature representations, using a d-dimensional embedding space, are defined as follows:

$H = \{\mathbf{h}_v \mid v \in V\} \subseteq \mathbb{R}^{|V| \times d}$ is the atom-level feature matrix, where each atom feature vector $\mathbf{h}_v$ encodes *atom-type* and *bond-type* information, such as atomic number, hybridization state, and bond orders.
$H_m = \left\{\mathbf{h}_{M_i}^{M} \mid M_i \in V_m\right\} \subseteq \mathbb{R}^{|V_m| \times d}$ is the motif-level feature matrix. Here, each motif feature vector $\mathbf{h}_{M_i}^{M}$ captures the structural and chemical properties of the motif M_i.
$H_g = \mathbf{h}_g \in \mathbb{R}^d$ is the graph-level feature vector $\mathbf{h}_g$, which aggregates global information from all motifs in $\tilde{G}$.

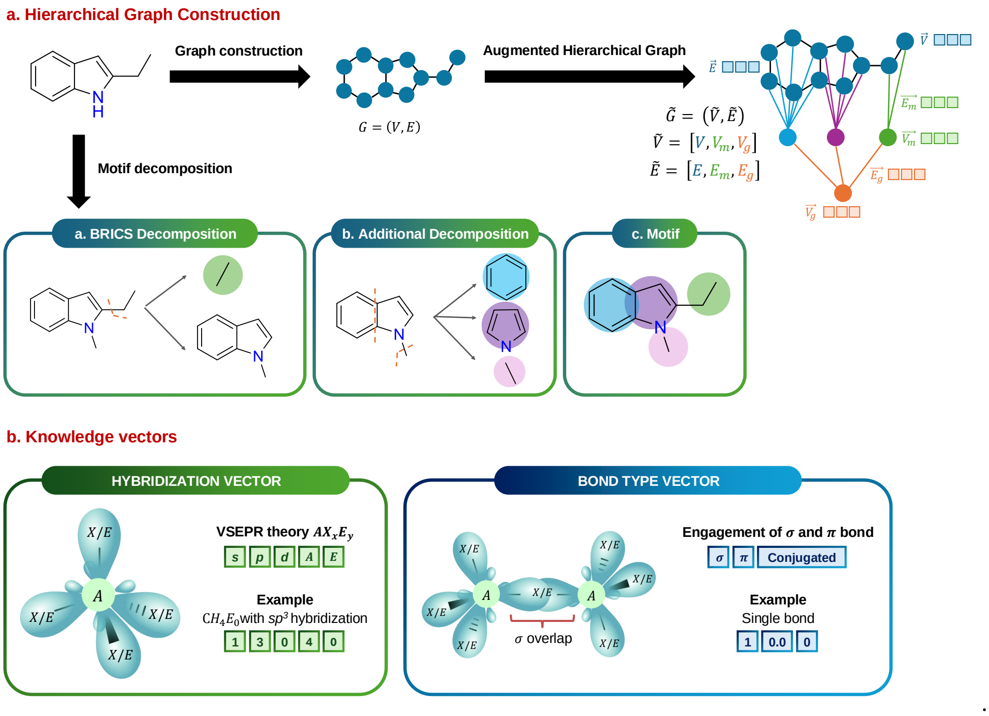


Fig. 6 Molecular Representation using KGG. (a) illustrates the three distinct levels of encoding. (b) demonstrates the knowledge vector concept applied to atoms and bonds.

4.2 Knowledge vectors

4.2.1 Node Representation

The input features of atoms are atomic number, degree, and hybridization, as described in Supporting Section 3.2.1. However, conventional one-hot encodings often overlook the inherent relationships among distinct hybridization states. In response, a knowledge vector-based encoding scheme can enrich molecular representations by incorporating both hybrid orbital compositions and features from VSEPR theory [46], summarized in Figure 6b.

Let $\mathcal{H}$ denote the set of possible hybridization states for a given atom:

$$\mathcal{H} = \{\text{UNSPECIFIED}, s, sp, sp^2, sp^3, sp^3d, sp^3d^2\}.$$

We define the *hybridization function* as $\phi_{\text{hybrid}} : \mathcal{H} \to \mathbb{N}^3$ that maps each state to (n_s, n_p, n_d), where n_s, n_p, and n_d denote the numbers of s, p, and d orbitals.

VSEPR theory correlates molecular shape with the arrangement of electron pairs around a central atom. The *VSEPR* descriptors, denoted AX_xE_y, can be formally defined by the *VSEPR function*: $\phi_{\text{VSEPR}} : \mathcal{V} \to \mathbb{N}^2$, that maps a molecular geometry to (x, y), where x is the number of bonded atoms X to the central atom A, and y is the number of lone pairs E on A.

To integrate hybridization with *VSEPR* descriptors into a single structural representation, we define the *composite function*: $\phi_{\text{comp}} : \mathcal{H} \times \mathcal{V} \to \mathbb{N}^5$, where $\phi_{\text{comp}}(\text{state, geometry}) = \big(n_s, n_p, n_d, x, y\big)$. and $(n_s, n_p, n_d) = \phi_{\text{hybrid}}(\text{state})$ and $(x, y) = \phi_{\text{VSEPR}}(\text{geometry})$.

As an example, consider the the carbon atom in methane CH_4E_0, illustrated in Figure 6b. Using our scheme, we represent this as: $\phi_{\text{comp}}(\text{C}) = (1, 3, 0, 4, 0)$. Here, 1 corresponds to the s orbital, 3 to the p orbitals, 0 to any d orbitals, 4 indicates the total number of sp^3 orbitals, and 0 indicates no lone pairs on carbon.

4.2.2 Edges Representation

The initial bond features comprise the bond type, ring state, and a knowledge vector that encodes distinct orbital contributions (see Supporting Section 3.2.2). This knowledge vector arises from quantum mechanical orbital interactions and is given by

$$\vec{e}_{ij} = \begin{pmatrix} \sigma_{ij} \\ \pi_{ij} \\ \delta_{ij} \end{pmatrix},$$

where σ_{ij}, π_{ij}, and δ_{ij} represent the sigma, pi, and conjugation contributions, respectively, each determined by the relevant orbital overlap integrals. Figure 6b provides a visual representation of these bond contributions. For example, the single bond in ethane, Figure 6b (ethane), is represented as $\vec{e}_{ij} = (1, 0.0, 0)^\top$ where 1 denotes the single σ-bond contribution, 0.0 indicates no π-bond component, and 0 denotes the absence of any conjugated state.

4.3 Molecular representation extractor

The objective of this stage is to encode the molecular graph features of an individual molecule into a one-dimensional vector for training purposes.

Specifically, consider the hierarchical graph $\tilde{G} = (\tilde{V}, \tilde{E})$, where the node set is defined as $\tilde{V} = V \cup V_m \cup V_g$ and the edge set as $\tilde{E} = E \cup E_m \cup E_g$. The initial feature associated with a node $v_i \in \tilde{V}$ is given by

$$X_{v_i} \in \mathbb{R}^{d_v},$$

while the feature corresponding to an edge $e_{v_i v_j} \in \tilde{E}$ is denoted by

$$X_{v_i v_j} \in \mathbb{R}^{d_e}.$$

Here, X_{v_i} is a 7-dimensional feature vector assigned to node v_i, and $X_{v_i v_j}$ is a 5-dimensional feature vector characterizing the connection between nodes v_i and v_j.

These feature vectors are then transformed by a MLP prior to their use in the GINConv layer, as expressed in

$$h^0_{v_i} = \sum_{k=1}^{d_v} MLP_k\big(x^k_{v_i}\big),$$

$$h^0_{v_i v_j} = \sum_{k=1}^{d_e} MLP_i\big(x^k_{v_i v_j}\big),$$

where $x^i_v \in \mathbb{R}$ is the k-th scalar component of the node feature vector X_{v_i}, and $x^i_{v_i v_j} \in \mathbb{R}$ is the k-th scalar component of the edge feature vector $X_{v_i v_j}$. Here, $h^0_{v_i}$ and $h^0_{v_i v_j}$ denote the input feature embeddings for node v_i and edge $v_i v_j$, respectively.

Next, $h^0_{v_i}$ and $h^0_{v_i v_j}$ are passed through a GINConv layer, defined by Equation (1):

$$h^{(l)}_{v_i} = MLP^{(l)}\Big(h^{(l-1)}_{v_i} + \sum_{v_j \in \mathbb{N}(v_i)} \big(h^{(l-1)}_{v_j} + h^0_{v_i v_j}\big)\Big), \tag{1}$$

where $\texttt{MLP}^{(l)} = \{\texttt{Linear}(d,\, 2d) \rightarrow \texttt{ReLU} \rightarrow \texttt{Linear}(2d,\, d)\}$, $h^{(l-1)}_{v_i}$ and $h^{(l-1)}_{v_j}$ are the embedding vectors of atoms v_i and v_j at layer $l-1$. $h^{(l)}_{v_i}$ denotes the embedding vector of atom v_i at layer l. $\mathbb{N}(v)$ denote the set of neighboring nodes of v_i in the graph. In general, Equation (1) can be rewritten as:

$$h^{(l)}_{v_i} = \texttt{GINConv}\big(h^{(l-1)}_{v_i},\, h^0_{v_i v_j}\big),$$

The KRG encoding model comprises five GINConv layers (Figure 7), interspersed with BatchNorm and Dropout layers. The feature update process for node v_i from the first to the fourth GINConv layer is given by

$$h^{(l)}_{v_i} = \texttt{Dropout}^{(l)}\Big(\texttt{ELU}\big(\texttt{BatchNorm}^{(l)}\big(\texttt{GINConv}^{(l)}(h^{(l-1)}_{v_i},\, h^0_{v_i v_j})\big)\big)\Big),$$

while, in the final fifth `GINConv` layer, the `ELU` activation function is omitted:

$$h_{v_i}^{(5)} = \texttt{Dropout}^{(5)}\Big(\texttt{BatchNorm}^{(5)}\big(\texttt{GINConv}^{(5)}(h_{v_i}^{(4)},\, h_{v_i v_j}^{0})\big)\Big)$$

Inspired by the success of `Graphormer` [62] and `Himol` [33], a `READOUT` function is not employed to obtain the global graph representations. Instead, we adopt the supernode embedding $h_{v_g}^{(5)}$ (indicated in orange in Figure 7) as the representation of the entire graph. This embedding is subsequently used for both pretext and downstream tasks.

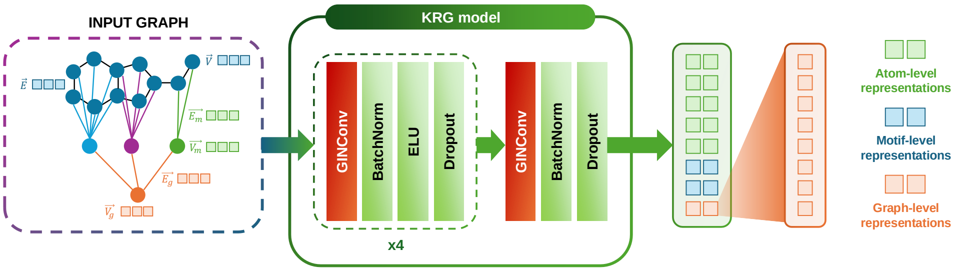


Fig. 7 An overview of the `KRG` encoder, which consists of five `GINConv` layers.

4.4 Pretrained model

The pre-training `KSSP` model is derived from eleven pretext tasks (Figure 8) and the configurations for this stage is detailed in Supporting Section 3.4.1. Eight of these tasks focus on reconstructing two knowledge vectors, one addresses the reconstruction of adjacency matrices, and two target the prediction of graph-level properties. A key aspect of this pre-training scheme lies in reconstructing orbital information embedded in two knowledge vectors (hybridization and bond types), which play a pivotal role in defining chemical semantics yet have been overlooked in prior studies [33–36, 40, 41, 48–54].

Reconstruction of hybridization vector: We utilize *atomic-level* embeddings (h_{v_i}) and employ Cross Entropy Loss (`CELoss`) to reconstruct five elements of $\phi_{\text{comp}}(\text{state, geometry}) = \big(n_s,\, n_p,\, n_d,\, x,\, y\big)$ as follows:

$$\begin{aligned}
h_{v_i} &\rightarrow \phi_{\mathrm{n_s}}\ \{\texttt{Linear}(d,d)\rightarrow\texttt{ReLU}\rightarrow\texttt{Linear}(d,2)\} \rightarrow \hat{y}_s\\
h_{v_i} &\rightarrow \phi_{\mathrm{n_p}}\ \{\texttt{Linear}(d,d)\rightarrow\texttt{ReLU}\rightarrow\texttt{Linear}(d,3)\} \rightarrow \hat{y}_p\\
h_{v_i} &\rightarrow \phi_{\mathrm{n_d}}\ \{\texttt{Linear}(d,d)\rightarrow\texttt{ReLU}\rightarrow\texttt{Linear}(d,4)\} \rightarrow \hat{y}_d\\
h_{v_i} &\rightarrow \phi_{\mathrm{x}}\ \{\texttt{Linear}(d,d)\rightarrow\texttt{ReLU}\rightarrow\texttt{Linear}(d,7)\} \rightarrow \hat{y}_{\mathrm{x}}\\
h_{v_i} &\rightarrow \phi_{\mathrm{y}}\ \{\texttt{Linear}(d,d)\rightarrow\texttt{ReLU}\rightarrow\texttt{Linear}(d,7)\} \rightarrow \hat{y}_{\mathrm{y}}
\end{aligned}$$

The hybridization loss is defined as the aggregate of five individual loss components, as detailed in Equation 2:

$$L_{\text{hybridization}} = L_{n_s}^{\text{CELoss}} + L_{n_p}^{\text{CELoss}} + L_{n_d}^{\text{CELoss}} + L_{\text{x}}^{\text{CELoss}} + L_{\text{y}}^{\text{CELoss}} \tag{2}$$

Reconstruction of bond type vector: Bond formation arises from axial (σ) and lateral (π) orbital overlaps, which determine properties such as bond strength and length. Consequently, predicting these overlaps enables the model to capture essential chemical information. The three bond-type tasks are defined as follows:

$$\text{concat}[h_{v_i},\, h_{v_j}] \;\rightarrow\; \phi_\sigma \;\{\texttt{Linear}(2d, d) \rightarrow \texttt{ReLU} \rightarrow \texttt{Linear}(d, 1)\} \;\rightarrow\; \hat{y}_\sigma$$

$$\text{concat}[h_{v_i},\, h_{v_j}] \;\rightarrow\; \phi_\pi \;\{\texttt{Linear}(2d, d) \rightarrow \texttt{ReLU} \rightarrow \texttt{Linear}(d, 1)\} \;\rightarrow\; \hat{y}_\pi$$

$$\text{concat}[h_{v_i},\, h_{v_j}] \;\rightarrow\; \phi_{\text{conjugation}} \;\{\texttt{Linear}(2d, d) \rightarrow \texttt{ReLU} \rightarrow \texttt{Linear}(d, 1)\} \;\rightarrow\; \hat{y}_{\text{conjugation}}$$

Here, h_{v_i} and h_{v_j} are the atomic-level embeddings of the two atoms v_i and v_j forming a covalent bond. The total bond type loss is the sum of three individual losses:

$$L_{\text{bond\_type}} = L_\sigma^{\text{BCE}} + L_\pi^{\text{SmoothL1}} + L_{\text{conjugation}}^{\text{BCE}}$$

Adjacency Matrix Reconstruction: This task is defined by

$$\text{concat}[h_{v_i},\, h_{v_j}] \;\rightarrow\; \phi_{\text{adj}} \;\{\texttt{Linear}(2d, d) \rightarrow \texttt{ReLU} \rightarrow \texttt{Linear}(d, 1)\} \;\rightarrow\; \hat{y}_{v_i v_j}$$

where $y_{v_i v_j} \in \{0, 1\}$ indicates whether a bond exists between atoms v_i and v_j. The loss for adjacency matrix reconstruction is

$$L_{\text{adj}} = -\sum_{v_i, v_j \in V} \big(y_{v_i v_j}\, \log \hat{y}_{v_i v_j} + (1 - y_{v_i v_j}) \log(1 - \hat{y}_{v_i v_j})\big)$$

Graph-Level Prediction Tasks: We utilize the *graph-level* embedding (h_g) to predict the number of atoms and the number of bonds. These tasks are expressed as:

$$h_g \;\rightarrow\; \phi_{\text{atoms}} \;\{\texttt{Linear}(d,\, \tfrac{d}{4}) \rightarrow \texttt{Softplus} \rightarrow \texttt{Linear}(\tfrac{d}{4},\, 1)\} \;\rightarrow\; \hat{y}_{\text{atoms}}$$

$$h_g \;\rightarrow\; \phi_{\text{bonds}} \;\{\texttt{Linear}(d,\, \tfrac{d}{4}) \rightarrow \texttt{Softplus} \rightarrow \texttt{Linear}(\tfrac{d}{4},\, 1)\} \;\rightarrow\; \hat{y}_{\text{bonds}}$$

Here, $\hat{y}_{\text{atoms}}$ and $\hat{y}_{\text{bonds}}$ are the predicted values for the number of atoms and bonds, respectively. The `SmoothL1Loss` is less sensitive to outliers compared to `MSE` [63] and, according to Girshick [64] can help prevent gradient explosions. Based on `SmoothL1Loss`, the losses are formulated as:

$$L_{\text{atoms}} = \begin{cases} \frac{1}{2} \big\| \, y_{\text{atoms}} - \hat{y}_{\text{atoms}} \big\|_2^2, & \text{if } \big\| \, y_{\text{atoms}} - \hat{y}_{\text{atoms}} \big\|_1 < 1 \\ \big\| \, y_{\text{atoms}} - \hat{y}_{\text{atoms}} \big\|_1 - \frac{1}{2}, & \text{otherwise} \end{cases}$$

$$L_{\text{bonds}} = \begin{cases} \frac{1}{2}\left\| \, y_{\text{bonds}} - \hat{y}_{\text{bonds}} \right\|_2^2, & \text{if } \left\| \, y_{\text{bonds}} - \hat{y}_{\text{bonds}} \right\|_1 < 1 \\ \left\| \, y_{\text{bonds}} - \hat{y}_{\text{bonds}} \right\|_1 - \frac{1}{2}, & \text{otherwise} \end{cases}$$

Objective Loss Function: Finally, the total loss for backpropagation is the sum of all five principal losses, as represented in Equation 3:

$$L = L_{\text{hybridization}} + L_{\text{bond\_type}} + L_{\text{adj}} + L_{\text{atoms}} + L_{\text{bonds}}. \tag{3}$$

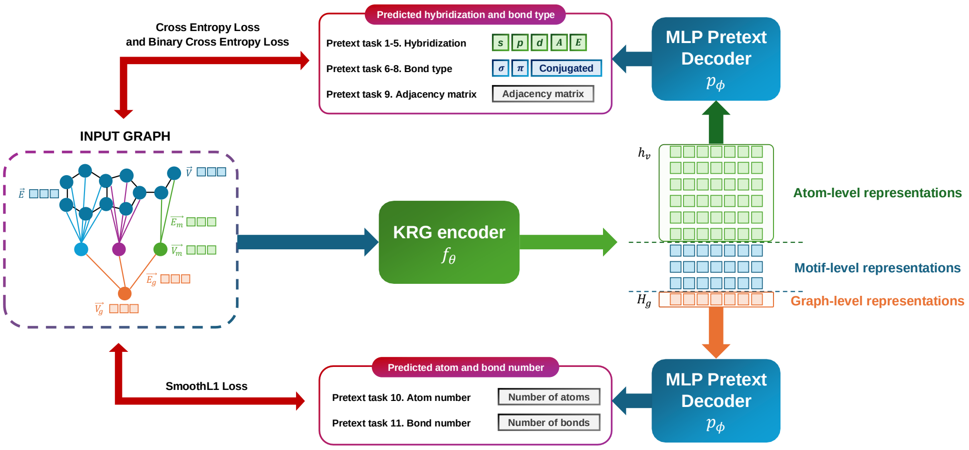

Cross Entropy Loss
and Binary Cross Entropy Loss
Predicted hybridization and bond type
Pretext task 1-5. Hybridization
s
p
d
A
E
Pretext task 6-8. Bond type
σ
π
Conjugated
Pretext task 9. Adjacency matrix
Adjacency matrix
MLP Pretext Decoder
p_ϕ
INPUT GRAPH
$\vec{E}$
$\vec{V}$
$\vec{E_m}$
$\vec{V_m}$
$\vec{E_g}$
$\vec{V_g}$
KRG encoder
f_θ
h_v
Atom-level representations
Motif-level representations
H_g
Graph-level representations
SmoothL1 Loss
Predicted atom and bond number
Pretext task 10. Atom number
Number of atoms
Pretext task 11. Bond number
Number of bonds
MLP Pretext Decoder
p_ϕ

Fig. 8 An architecture of the KSSP decoder.

4.5 Datasets and Baselines

For the pre-training stage, approximately 250,000 molecules were randomly sampled from the ZINC15 dataset [65]. The model was fine-tuned using twelve molecular property datasets from MoleculeNet [47], comprising six classification and six regression tasks. Further details on both the pre-training and fine-tuning datasets are provided in the Supporting Section 2.1, and summarized in Table S6. Following established studies [33–36, 40, 41, 48–54], we adopted a *scaffold* splitting strategy to divide each dataset into training, validation, and test subsets at an 8:1:1 ratio, thereby ensuring structural differences between training and test molecules and assessing model generalizability. Subsequently, we benchmarked the KGG model against a set of state-of-the-art self-supervised learning baselines. Further details regarding these benchmark methods are available in the Supporting Section 2.2.

4.6 Experimental Settings

Training configurations

The research was conducted on a Linux System 22.04 LTS, powered by an Intel® Core™ i7-13700K processor featuring 16 processing units and 24 CPUs operating at 3.10 GHz. The system includes 512 GB of memory and 96 GB of DDR4 RAM, with graphics capabilities provided by a GTX 4070 Ti Super card containing 16 GB of VRAM.

During both pre-training and fine-tuning phases, the Adam optimizer was utilized, along with a batch size of 32 and embedding dimensionality of 512. Fine-tuning was repeated three times per dataset with distinct random seeds. Further training configuration details can be found in Supporting Section 3.4.

Comparisions with traditional fingerprints

To comprehensively assess the molecular representation capabilities of the `KGG` neural graph fingerprints—specifically, *global representations* extracted from the fine-tuned `KRG` layers of the `KGG` model—were generated for two classification datasets (BACE and BBBP) and four regression datasets (ESOL, FreeSolv, Lipophilicity, and QM7). Subsequently, these embeddings were evaluated using a `kNN` classifier with $n\_neighbors = 3$ to quantify classification performance through the ROC-AUC metric, as well as to measure predictive accuracy on regression tasks using RMSE and MAE. For comparative purposes, identical analyses were performed with three established molecular fingerprints: `MACCS` [66], `ECFP4` [67], and `RDK7` [68]. Dataset partitioning was conducted according to the strategy described in Section 4.5. Additionally, the discriminative power of the `KGG` fingerprints was further illustrated through visualization of the embeddings using the `t-SNE` [69] algorithm, clearly demonstrating their capacity to effectively distinguish molecular structures within a two-dimensional embedding space.

Data contamination

The methodology comprises two principal steps: (1) extracting every compound from the test sets of each fine-tuning dataset, and (2) evaluating the extent of overlap between these compounds and those in the pre-training dataset on the basis of normalized canonical SMILES structures. This normalization and comparison are conducted using the `RDKit` library to ensure consistency and accuracy in detecting potential data contamination.

Data Availability

All code and the pre-trained `KGG` model are publicly available at https://github.com/ThinhUMP/KGGraph. The ZINC dataset employed for pre-training is downloadable from https://github.com/ZangXuan/HiMol, as outlined in `Himol` [33]. Furthermore, the downstream datasets used for fine-tuning can be obtained through the MoleculeNet repository at https://github.com/deepchem/deepchem/tree/master/deepchem/molnet/load_function.

Supporting Information

Supporting Information file can be found at add link.

Author Contribution

V.T.T. developed and structured the experimental source code and composed the manuscript. P.C.V.N. and G.B.T. handled data acquisition and conducted visualizations. T.M.P. packaged the models and deployed them on GitHub. T.L.P., R.F., P.F.S., and T.N.T. introduced key concepts, contributed to manuscript revisions, and guided the overall direction of the project. All authors reviewed and approved the final version of the manuscript.

Acknowledgements

This work has received support from the Korea International Cooperation Agency (KOICA) under the project entitled “Education and Research Capacity Building Project at University of Medicine and Pharmacy at Ho Chi Minh City”, conducted from 2024 to 2025 (Project No. 2021-00020-3).

Declarations

The authors declare no competing financial interest.

References

[1] McNutt, A.T., Francoeur, P., Aggarwal, R., Masuda, T., Meli, R., Ragoza, M., Sunseri, J., Koes, D.R.: Gnina 1.0: molecular docking with deep learning. Journal of Cheminformatics **13**(1), 43 (2021) https://doi.org/10.1186/s13321-021-00522-2

[2] Corso, G., Deng, A., Polizzi, N., Barzilay, R., Jaakkola, T.S.: Deep confident steps to new pockets: Strategies for docking generalization. In: The Twelfth International Conference on Learning Representations (2024). https://doi.org/10.48550/arXiv.2402.18396

[3] Jin, W., Barzilay, R., Jaakkola, T.: Junction tree variational autoencoder for molecular graph generation. In: International Conference on Machine Learning, pp. 2323–2332 (2018). PMLR

[4] Jin, W., Barzilay, D.R., Jaakkola, T.: Hierarchical generation of molecular graphs using structural motifs. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 4839–4848 (2020). PMLR. https://proceedings.mlr.press/v119/jin20a.html

[5] Lim, J., Ryu, S., Kim, J.W., Kim, W.Y.: Molecular generative model based on conditional variational autoencoder for de novo molecular design. Journal of Cheminformatics **10**(1), 31 (2018) https://doi.org/10.1186/s13321-018-0286-7

[6] Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A.J., Bambrick, J., Bodenstein, S.W., Evans, D.A., Hung, C.-C., O'Neill, M., Reiman, D., Tunyasuvunakool, K., Wu, Z., Žemgulytė, A., Arvaniti, E., Beattie, C., Bertolli, O., Bridgland, A., Cherepanov, A., Congreve, M., Cowen-Rivers, A.I., Cowie, A., Figurnov, M., Fuchs, F.B., Gladman, H., Jain, R., Khan, Y.A., Low, C.M.R., Perlin, K., Potapenko, A., Savy, P., Singh, S., Stecula, A., Thillaisundaram, A., Tong, C., Yakneen, S., Zhong, E.D., Zielinski, M., Žídek, A., Bapst, V., Kohli, P., Jaderberg, M., Hassabis, D., Jumper, J.M.: Accurate structure prediction of biomolecular interactions with alphafold 3. Nature **630**(8016), 493–500 (2024) https://doi.org/10.1038/s41586-024-07487-w

[7] Hessler, G., Baringhaus, K.-H.: Artificial intelligence in drug design. Molecules **23**(10) (2018) https://doi.org/10.3390/molecules23102520

[8] Walters, W.P., Barzilay, R.: Applications of deep learning in molecule generation and molecular property prediction. Accounts of Chemical Research **54**(2), 263–270 (2021) https://doi.org/10.1021/acs.accounts.0c00699 . PMID: 33370107

[9] Wieder, O., Kohlbacher, S., Kuenemann, M., Garon, A., Ducrot, P., Seidel, T., Langer, T.: A compact review of molecular property prediction with graph neural networks. Drug Discovery Today: Technologies **37**, 1–12 (2020) https://doi.org/10.1016/j.ddtec.2020.11.009

[10] Wu, Z., Ramsundar, B., Feinberg, E., Gomes, J., Geniesse, C., Pappu, A.S., Leswing, K., Pande, V.: Moleculenet: a benchmark for molecular machine learning. Chem. Sci. **9**, 513–530 (2018) https://doi.org/10.1039/C7SC02664A

[11] Tkatchenko, A.: Machine learning for chemical discovery. Nature Communications **11**(1), 4125 (2020) https://doi.org/10.1038/s41467-020-17844-8

[12] David, L., Thakkar, A., Mercado, R., Engkvist, O.: Molecular representations in ai-driven drug discovery: a review and practical guide. Journal of Cheminformatics **12**(1), 56 (2020) https://doi.org/10.1186/s13321-020-00460-5

[13] Grisoni, F., Ballabio, D., Todeschini, R., Consonni, V.: In: Nicolotti, O. (ed.) Molecular Descriptors for Structure–Activity Applications: A Hands-On Approach, pp. 3–53. Springer, New York, NY (2018). https://doi.org/10.1007/978-1-4939-7899-1_1

[14] Grisoni, F., Consonni, V., Todeschini, R.: In: Brown, J.B. (ed.) Impact of Molecular Descriptors on Computational Models, pp. 171–209. Springer, New York, NY (2018). https://doi.org/10.1007/978-1-4939-8639-2_5

[15] Moriwaki, H., Tian, Y.-S., Kawashita, N., Takagi, T.: Mordred: a molecular descriptor calculator. Journal of Cheminformatics **10**(1), 4 (2018) https://doi.org/10.1186/s13321-018-0258-y

[16] Rogers, D., Hahn, M.: Extended-connectivity fingerprints. Journal of Chemical Information and Modeling **50**(5), 742–754 (2010) https://doi.org/10.1021/ci100050t https://doi.org/10.1021/ci100050t. PMID: 20426451

[17] Capecchi, A., Probst, D., Reymond, J.-L.: One molecular fingerprint to rule them all: drugs, biomolecules, and the metabolome. Journal of Cheminformatics **12**(1), 43 (2020) https://doi.org/10.1186/s13321-020-00445-4

[18] Zagidullin, B., Wang, Z., Guan, Y., Pitkänen, E., Tang, J.: Comparative analysis of molecular fingerprints in prediction of drug combination effects. Briefings in Bioinformatics **22**(6), 291 (2021) https://doi.org/10.1093/bib/bbab291 https://academic.oup.com/bib/article-pdf/22/6/bbab291/41974966/bbab291.pdf

[19] Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: International Conference on Learning Representations (2019). https://doi.org/10.48550/arXiv.1810.00826

[20] Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2017). https://doi.org/10.48550/arXiv.1609.02907 . https://openreview.net/forum?id=SJU4ayYgl

[21] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018). https://openreview.net/forum?id=rJXMpikCZ

[22] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems **32**(1), 4–24 (2021) https://doi.org/10.1109/TNNLS.2020.2978386

[23] Li, Z., Shen, X., Jiao, Y., Pan, X., Zou, P., Meng, X., Yao, C., Bu, J.: Hierarchical bipartite graph neural networks: Towards large-scale e-commerce applications. In: 2020 IEEE 36th International Conference on Data Engineering (ICDE), pp. 1677–1688 (2020). https://doi.org/10.1109/ICDE48307.2020.00149

[24] Wu, Z., Pan, S., Long, G., Jiang, J., Zhang, C.: Graph wavenet for deep spatial-temporal graph modeling. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, pp. 1907–1913. International Joint Conferences on Artificial Intelligence Organization, ??? (2019). https://doi.org/10.24963/ijcai.2019/264 . https://doi.org/10.24963/ijcai.2019/264

[25] Liu, Q., Allamanis, M., Brockschmidt, M., Gaunt, A.: Constrained graph variational autoencoders for molecule design. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 31. Curran Associates, Inc., ??? (2018)

[26] Jiang, D., Wu, Z., Hsieh, C.-Y., Chen, G., Liao, B., Wang, Z., Shen, C., Cao, D., Wu, J., Hou, T.: Could graph neural networks learn better molecular representation for drug discovery? a comparison study of descriptor-based and graph-based models. Journal of cheminformatics **13**, 1–23 (2021)

[27] Reiser, P., Neubert, M., Eberhard, A., Torresi, L., Zhou, C., Shao, C., Metni, H., Hoesel, C., Schopmans, H., Sommer, T., *et al.*: Graph neural networks for materials science and chemistry. Communications Materials **3**(1), 93 (2022)

[28] Rong, Y., Bian, Y., Xu, T., Xie, W., WEI, Y., Huang, W., Huang, J.: Self-supervised graph transformer on large-scale molecular data. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems, vol. 33, pp. 12559–12571. Curran Associates, Inc., Newry, UK (2020)

[29] Hu, Z., Dong, Y., Wang, K., Chang, K.-W., Sun, Y.: Gpt-gnn: Generative pre-training of graph neural networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD '20, pp. 1857–1867. Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3394486.3403237 . https://doi.org/10.1145/3394486.3403237

[30] Rong, Y., Huang, W., Xu, T., Huang, J.: DropEdge: Towards deep graph convolutional networks on node classification. In: International Conference on Learning Representations (2020). https://openreview.net/forum?id=Hkx1qkrKPr

[31] Zhang, M., Hu, L., Shi, C., Wang, X.: Adversarial label-flipping attack and defense for graph neural networks. In: 2020 IEEE International Conference on Data Mining (ICDM), pp. 791–800 (2020). https://doi.org/10.1109/ICDM50108.2020.00088

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

[33] Zang, X., Zhao, X., Tang, B.: Hierarchical molecular graph self-supervised learning for property prediction. Communications Chemistry **6**(1), 34 (2023) https://doi.org/10.1038/s42004-023-00825-5

[34] Rong, Y., Bian, Y., Xu, T., Xie, W., Wei, Y., Huang, W., Huang, J.: Self-supervised graph transformer on large-scale molecular data. Advances in neural information processing systems **33**, 12559–12571 (2020)

[35] Sun, M., Xing, J., Wang, H., Chen, B., Zhou, J.: MoCL: data-driven molecular fingerprint via knowledge-aware contrastive learning from molecular graph. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 3585–3594 (2021)

[36] Zhang, Z., Liu, Q., Wang, H., Lu, C., Lee, C.-K.: Motif-based graph self-supervised learning for molecular property prediction. Advances in Neural Information Processing Systems **34**, 15870–15882 (2021)

[37] Wang, H., Li, W., Jin, X., Cho, K., Ji, H., Han, J., Burke, M.D.: Chemical-reaction-aware molecule representation learning. In: International Conference on Learning Representations (2022)

[38] Zhang, S., Hu, Z., Subramonian, A., Sun, Y.: Motif-driven contrastive learning of graph representations. IEEE Transactions on Knowledge and Data Engineering **36**(8), 4063–4075 (2024) https://doi.org/10.1109/TKDE.2024.3364059

[39] Wang, Y., Magar, R., Liang, C., Barati Farimani, A.: Improving molecular contrastive learning via faulty negative mitigation and decomposed fragment contrast. Journal of Chemical Information and Modeling **62**(11), 2713–2725 (2022) https://doi.org/10.1021/acs.jcim.2c00495 https://doi.org/10.1021/acs.jcim.2c00495. PMID: 35638560

[40] Wang, Y., Wang, J., Cao, Z., Barati Farimani, A.: Molecular contrastive learning of representations via graph neural networks. Nature Machine Intelligence **4**(3), 279–287 (2022)

[41] Li, H., Zhang, R., Min, Y., Ma, D., Zhao, D., Zeng, J.: A knowledge-guided pre-training framework for improving molecular representation learning. Nature Communications **14**(1), 7568 (2023)

[42] Kosaraju, N., Sankepally, S.R., Mallikharjuna Rao, K.: Categorical data: Need, encoding, selection of encoding method and its emergence in machine learning models—a practical review study on heart disease prediction dataset using pearson correlation. In: Saraswat, M., Chowdhury, C., Kumar Mandal, C., Gandomi, A.H. (eds.) Proceedings of International Conference on Data Science and Applications, pp. 369–382. Springer, Singapore (2023)

[43] Pattanaik, L., Coley, C.W.: Molecular representation: going long on fingerprints. Chem **6**(6), 1204–1207 (2020)

[44] Jiang, M., Liu, K., Zhong, M., Schaeffer, R., Ouyang, S., Han, J., Koyejo, S.: Does data contamination make a difference? insights from intentionally contaminating pre-training data for language models. In: ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models (2024). https://openreview.net/forum?id=wSpwj7xab9

[45] Benkö, G., Flamm, C., Stadler, P.F.: A graph-based toy model of chemistry. J. Chem. Inf. Comput. Sci. **43**, 1085–1093 (2003) https://doi.org/10.1021/ci0200570

[46] Gillespie, R.: The valence-shell electron-pair repulsion (vsepr) theory of directed valency. Journal of Chemical Education **40**(6), 295 (1963)

[47] Wu, Z., Ramsundar, B., Feinberg, E.N., Gomes, J., Geniesse, C., Pappu, A.S., Leswing, K., Pande, V.: MoleculeNet: a benchmark for molecular machine learning. Chemical science **9**(2), 513–530 (2018)

[48] Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Advances in neural information processing systems **30** (2017)

[49] Hu, Z., Dong, Y., Wang, K., Chang, K.-W., Sun, Y.: GPT-GNN: Generative pre-training of graph neural networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1857–1867 (2020)

[50] Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., Leskovec, J.: Strategies for pre-training graph neural networks. arXiv preprint arXiv:1905.12265 (2019)

[51] Sun, F.-Y., Hoffmann, J., Verma, V., Tang, J.: Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. arXiv preprint arXiv:1908.01000 (2019)

[52] Xu, M., Wang, H., Ni, B., Guo, H., Tang, J.: Self-supervised graph-level representation learning with local and global structure. In: International Conference on Machine Learning, pp. 11548–11558 (2021). PMLR

[53] You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., Shen, Y.: Graph contrastive learning with augmentations. Advances in neural information processing systems **33**, 5812–5823 (2020)

[54] You, Y., Chen, T., Shen, Y., Wang, Z.: Graph contrastive learning automated. In: International Conference on Machine Learning, pp. 12121–12132 (2021). PMLR

[55] Vishnusai, Y., Kulakarni, T.R., Sowmya Nag, K.: Ablation of artificial neural networks. In: Raj, J.S., Bashar, A., Ramson, S.R.J. (eds.) Innovative Data Communication Technologies and Application, pp. 453–460. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-38040-3_52

[56] Chicco, D., Jurman, G.: The matthews correlation coefficient (mcc) should replace the roc auc as the standard metric for assessing binary classification. BioData Mining **16**(1), 4 (2023)

[57] Chicco, D., Tötsch, N., Jurman, G.: The matthews correlation coefficient (mcc) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. BioData mining **14**, 1–22 (2021)

[58] Chicco, D., Jurman, G.: The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. BMC genomics **21**, 1–13 (2020)

[59] Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 233–240 (2006)

[60] Landrum, G., *et al.*: Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. Greg Landrum **8**(31.10), 5281 (2013)

[61] Degen, J., Wegscheid-Gerlach, C., Zaliani, A., Rarey, M.: On the art of compiling and using'drug-like'chemical fragment spaces. ChemMedChem **3**(10), 1503 (2008)

[62] Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., Liu, T.-Y.: Do transformers really perform badly for graph representation? Advances in neural information processing systems **34**, 28877–28888 (2021)

[63] Bickel, P.J., Doksum, K.A.: Mathematical Statistics: Basic Ideas and Selected Topics, Volumes I-II Package. Chapman and Hall/CRC, ??? (2015)

[64] Girshick, R.: Fast r-cnn. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1440–1448 (2015). https://doi.org/10.1109/ICCV.2015.169

[65] Sterling, T., Irwin, J.J.: ZINC 15–ligand discovery for everyone. Journal of chemical information and modeling **55**(11), 2324–2337 (2015)

[66] Cereto-Massagué, A., Ojeda, M.J., Valls, C., Mulero, M., Garcia-Vallvé, S., Pujadas, G.: Molecular fingerprint similarity search in virtual screening. Methods **71**, 58–63 (2015)

[67] Rogers, D., Hahn, M.: Extended-connectivity fingerprints. Journal of chemical information and modeling **50**(5), 742–754 (2010)

[68] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., *et al.*: Scikit-learn: Machine learning in python. the Journal of machine Learning research **12**, 2825–2830 (2011)

[69] Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)