nature machine intelligence

Article

https://doi.org/10.1038/s42256-023-00654-0

Knowledge graph-enhanced molecular contrastive learning with functional prompt

Received: 12 September 2022

Accepted: 6 April 2023

Published online: 4 May 2023

Check for updates

Yin Fang[1,2,3], Qiang Zhang[1,2] ✉, Ningyu Zhang[1,4,5], Zhuo Chen[1,2], Xiang Zhuang[1,2], Xin Shao[3], Xiaohui Fan[3,6,7] ✉ & Huajun Chen[1,2,5,8] ✉

Deep learning models can accurately predict molecular properties and help making the search for potential drug candidates faster and more efficient. Many existing methods are purely data driven, focusing on exploiting the intrinsic topology and construction rules of molecules without any chemical prior information. The high data dependency makes them difficult to generalize to a wider chemical space and leads to a lack of interpretability of predictions. Here, to address this issue, we introduce a chemical element-oriented knowledge graph to summarize the basic knowledge of elements and their closely related functional groups. We further propose a method for knowledge graph-enhanced molecular contrastive learning with functional prompt (KANO), exploiting external fundamental domain knowledge in both pre-training and fine-tuning. Specifically, with element-oriented knowledge graph as a prior, we first design an element-guided graph augmentation in contrastive-based pre-training to explore microscopic atomic associations without violating molecular semantics. Then, we learn functional prompts in fine-tuning to evoke the downstream task-related knowledge acquired by the pre-trained model. Extensive experiments show that KANO outperforms state-of-the-art baselines on 14 molecular property prediction datasets and provides chemically sound explanations for its predictions. This work contributes to more efficient drug design by offering a high-quality knowledge prior, interpretable molecular representation and superior prediction performance.

Molecular property prediction is widely considered one of the most important tasks in drug discovery. Traditional wet-lab experiments are time consuming and require a huge and incessant investment[1,2]. With artificial intelligence, researchers have studied molecular property prediction models to assess the clinical trial success rate and therapeutic potential of drug candidates, or even directly predict whether a compound will receive US Food and Drug Administration approval, substantially speeding up drug development and avoiding costly late-stage failures.

With the increasing availability of chemical experimental data, researchers have adopted pre-training models on extensive collections of unlabelled molecules, followed by fine-tuning on a limited number of labelled molecules for a specific task[3–6]. Most of these self-supervised learning (SSL) methods on molecules are purely data driven, focusing

[1]College of Computer Science and Technology, Zhejiang University, Hangzhou, China. [2]ZJU-Hangzhou Global Scientific and Technological Innovation Center, Hangzhou, China. [3]College of Pharmaceutical Sciences, Zhejiang University, Hangzhou, China. [4]School of Software Technology, Zhejiang University, Ningbo, China. [5]Alibaba-ZJU Frontier Technology Research Center, Hangzhou, China. [6]Future Health Laboratory, Innovation Center of Yangtze River Delta, Zhejiang University, Jiaxing, China. [7]National Key Laboratory of Modern Chinese Medicine Innovation and Manufacturing, Hangzhou, China. [8]Donghai Laboratory, Zhoushan, China. ✉e-mail: qiang.zhang.cs@zju.edu.cn; fanxh@zju.edu.cn; huajunsir@zju.edu.cn

on exploiting the intrinsic information of molecular graphs without any prior chemical knowledge[7–10]. Moreover, with the enormous chemical space, these models rely heavily on pre-training datasets and may not generalize well to different downstream prediction tasks. Additionally, models that capture only the topology of molecular graphs and simple construction rules generally yield low interpretability. Therefore, it is important to leverage the fundamental chemical knowledge as a prior to guide the model to explore the chemical semantics of molecules at the microscopic level and discover meaningful patterns in both pre-training and fine-tuning.

As a typical SSL method, contrastive learning has attracted more research interest. To construct similar pairs and maximize agreement between them, existing methods rely on universal graph augmentation techniques that include node deletion, edge perturbation and subgraph extraction[11]. However, these techniques can be unsuitable for molecular graphs due to the considerable impact of adding or removing chemical bonds or atoms, which can alter the molecule's properties and identity[12]. Moreover, most existing methods consider only the connections between atoms established by chemical bonds, and thus do not fully explore the underlying relations of atoms in a molecular graph, which also highlights the key to incorporating external domain knowledge.

Another neglected issue is that the pre-training tasks differ greatly from the downstream tasks. Directly applying pre-trained representations to downstream tasks may result in suboptimal performance. In this Article, to address this, we propose providing a chemical prompt during fine-tuning based on fundamental chemical knowledge to bridge this gap. Inspired by prompt-tuning[13], an emerging paradigm that has demonstrated remarkable performance on a wide range of natural language processing tasks[14–17], it is crucial to devise appropriate prompts for molecular graphs based on fundamental chemical knowledge to enable more reliable predictions.

To this end, we propose a chemical element-oriented knowledge graph (ElementKG), which integrates basic knowledge of elements and functional groups in an organized and standardized manner. Then we exploit the contained fundamental chemical knowledge as a prior in both pre-training and fine-tuning, and propose a novel knowledge graph-enhanced molecular contrastive learning with functional prompt (KANO).

Firstly, we construct a chemical ElementKG based on the Periodic Table (https://ptable.com) and Wikipedia pages (https://en.wikipedia.org/wiki/Functional_group). ElementKG offers a comprehensive and standardized view from a chemical element perspective, which forms the foundation of our work. ElementKG covers the class hierarchy of elements, the chemical attributes of elements, the relationships between elements, the corresponding functional groups, and the connections between functional groups and their constituent elements.

Second, we introduce an element-guided graph augmentation in contrastive pre-training. Specifically, we augment the original molecular graph under the guidance of element knowledge in ElementKG, extracting rich relations between elements and associations between atoms that share the same element type but are not directly connected by chemical bonds. The resulting augmented graph respects the chemical semantics within molecules and establishes essential connections between atoms that go beyond the structural information. On top of this, a contrastive learning framework is developed to avoid indiscriminate implantation of external knowledge and to mitigate injection noise by allowing the two graph views to complement each other.

Third, we propose functional prompts to bridge the gap between pre-training contrastive tasks and downstream molecular property prediction tasks. As sets of atoms bonded together in a specific pattern, functional groups play a crucial role in determining the properties of the parent molecule[18] and are therefore closely related to downstream tasks. Therefore, in fine-tuning, we utilize the functional group knowledge in ElementKG to generate functional prompts, prompting the pre-trained model to recall task-related knowledge.

Finally, we thoroughly evaluate KANO on 14 various molecular property prediction tasks, demonstrating its superiority over competitive baselines. We also conduct extensive experiments to verify the necessity of each component of KANO, and to investigate its robustness and interpretability.

Results

Overview of KANO

In this paper, we propose KANO, a new KG-enhanced molecular contrastive learning with functional prompt method, which consists of three main components: (1) ElementKG construction and embedding, (2) contrastive-based pre-training and (3) prompt-enhanced fine-tuning. An overview of KANO is shown in Fig. 1.

ElementKG construction and embedding. Chemical domain knowledge is critical for molecular analysis, and integrating it into structured data can make it more standardized and easier to use. Some researchers have built KGs from public chemical databases and scientific literature to extract associations between chemicals and diseases or drug pairs[19,20]. However, in contrast to these approaches, we focus on the most fundamental chemical knowledge—the chemical elements. Over more than a century, the Periodic Table has evolved into an interrelated and complete system of elements, revealing the inherent laws of the complex real world and enabling chemical research to achieve a fundamental leap from phenomenon to essence. While a recent study[11] developed a KG that incorporates elements and their corresponding chemical attributes, its basic relations are inadequate for accommodating thorough and well-organized fundamental chemical knowledge. To provide a holistic view of the Periodic Table, we construct an element-oriented KG that combs the class hierarchy, data properties and object properties of elements. Additionally, we recognize the importance of functional groups and their close relationship to chemical elements, and thus, we collect relevant knowledge about functional groups from Wikipedia pages to make ElementKG more informative.

Figure 2a shows a snapshot of ElementKG, which consists of two levels: instance level and class level, coloured as red and blue, respectively. At the instance level, chemical elements and functional groups are represented as entities in ElementKG, denoted by red blocks. To record various chemical attributes of each element (for example, electron affinity and boiling point) and the composition of each functional group (for example, bond type), we apply data properties that attach literal data type values to an entity. The dotted block represents the data properties of the entity in the red block above it. Furthermore, as indicated by the red arrows, we establish associations between entities through object properties, such as chemical attribute relations between elements and the inclusion relations between elements and functional groups. We then classify all entities on the basis of their commonalities, resulting in the class level of ElementKG. Entities are assigned to the corresponding classes via rdf:type, denoted by dashed black arrows. The blue blocks represent different classes, while the blue arrows reflect the inclusion (rdfs:subClassOf) or disjointness (owl:disjointWith) between them. In particular, the subClassOf relations between classes form the class hierarchy, which serves as the backbone of ElementKG. The construction details can be found in Methods, and the statistics of ElementKG are displayed in Supplementary Information.

To comprehensively explore the structural and semantic information and obtain meaningful representations of all entities, relations and other components in ElementKG, we adopt a KG embedding approach based on OWL2Vec* (ref. 21). For further elaboration, please see Methods.

Contrastive-based pre-training. After obtaining ElementKG and its embeddings, we aim to incorporate it into pre-training to enhance the model's understanding of fundamental domain knowledge. We employ

https://doi.org/10.1038/s42256-023-00654-0

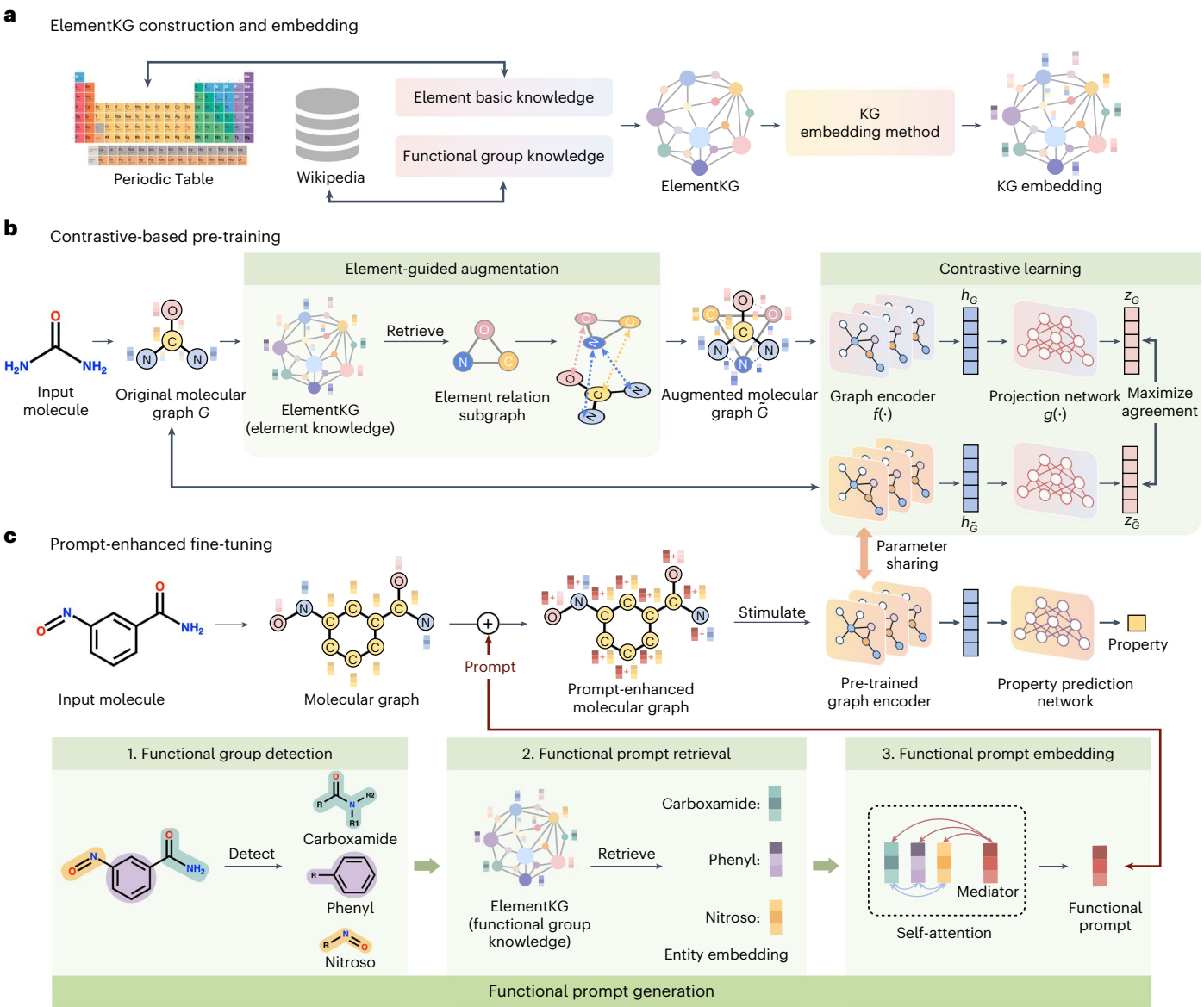


Fig. 1 | Overview of KANO. a, ElementKG construction and embedding. We collect basic element knowledge from the Periodic Table and functional group knowledge from Wikipedia pages to build ElementKG. Then we apply the KG embedding method to obtain the embeddings of all entities and relations in ElementKG. **b**, Contrastive-based pre-training. We use an element-guided graph augmentation strategy based on element knowledge of ElementKG to convert the original molecular graph G into the augmented molecular graph $\tilde{G}$, establishing essential connections between atoms beyond the inherent structure. The graph encoders are then trained to maximize the agreement between these two graph views to avoid excessive knowledge injection in $\tilde{G}$. **c**, Prompt-enhanced fine-tuning. We leverage functional group knowledge of ElementKG to generate a corresponding functional prompt for each molecule, stimulating the pre-trained graph encoder to recall the learned molecular property-related knowledge and bridging the gap between the pre-training contrastive tasks and the downstream tasks. The resulting prompt-enhanced molecular graph is then fed into the pre-trained graph encoder for molecular property prediction.

a contrastive learning method to pre-train a graph encoder on a large set of unlabelled molecules, using the basic element knowledge in ElementKG. Traditional graph augmentation techniques for creating positive pairs of contrastive learning often involve dropping nodes or perturbing edges, which can violate chemical semantics within molecules. To address this issue and establish more meaningful connections between atoms, we propose an element-guided graph augmentation approach for constructing positive pairs in contrastive learning.

As shown in Fig. 1b, we begin by identifying the element types present in a given molecule (for example, C, N and O) and retrieving their corresponding entities and relations from ElementKG (for example, (N, hasStateGas, O), (O, inPeriod2, C)). This forms an element relation subgraph that describes the relationships between elements using their associated entities and relations. We link the element entity nodes in this subgraph to their corresponding atom nodes in the original molecular graph to create an augmented molecular graph that integrates fundamental domain knowledge and captures the essential associations between atoms that share the same element type, even if they are not directly connected by chemical bonds. Our approach preserves the topology structure while incorporating important chemical semantics. Additional details about the input features and the triple definition can be found in Supplementary Information.

On top of this, we employ a contrastive learning framework to train the graph encoder by maximizing the consistency between the original molecular graph and the augmented molecular graph, without indiscriminately embedding element knowledge in the augmented graph.

https://doi.org/10.1038/s42256-023-00654-0

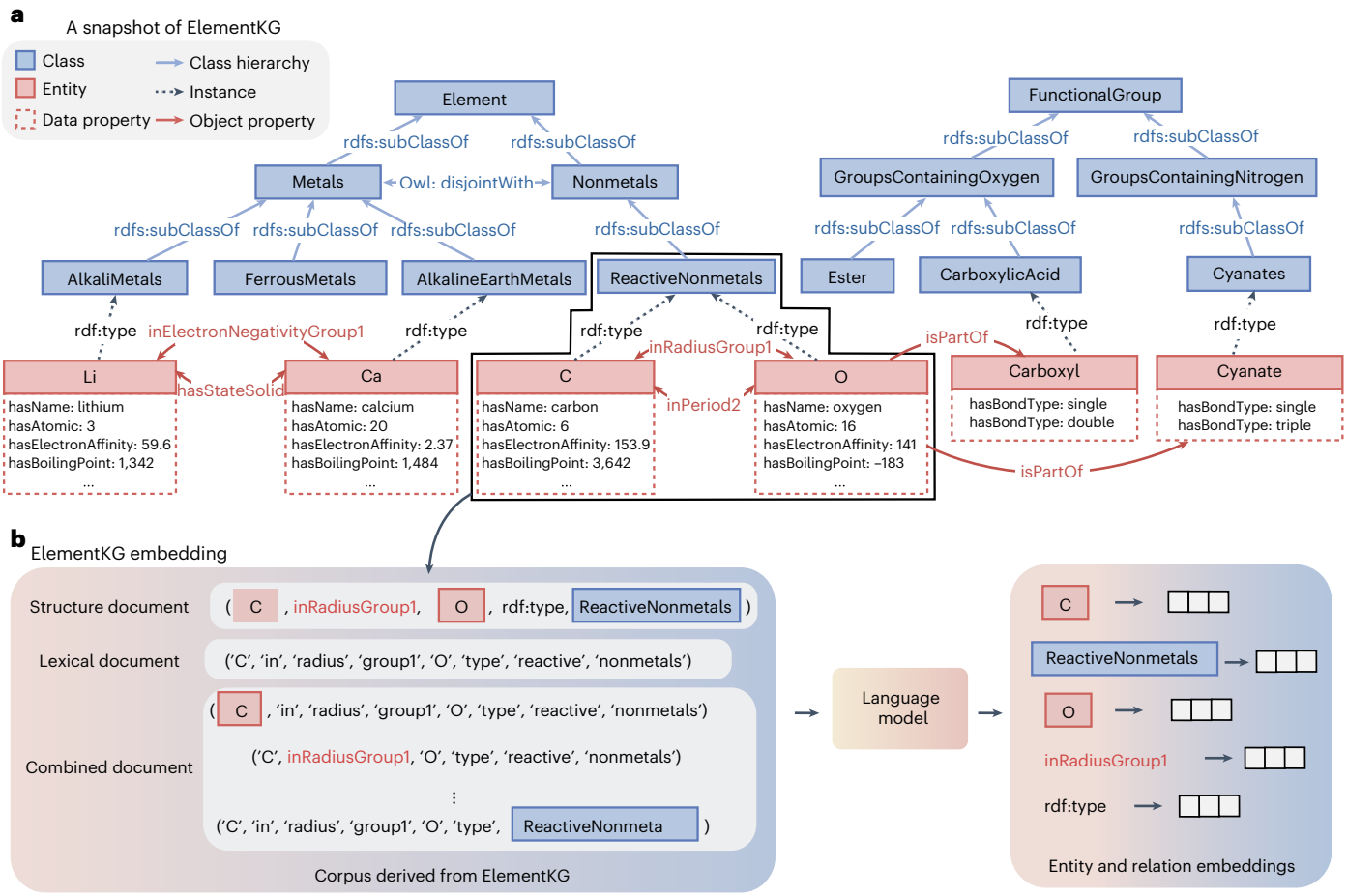


Fig. 2 | Illustration of ElementKG and its embedding process. a, A snapshot of ElementKG. ElementKG contains the class hierarchy, data properties, object properties and entities of both elements and functional groups. **b**, The process of ElementKG embedding. We derive a corpus of three documents (structure document, lexical document and combined document) from ElementKG, considering the structural topology, literal semantics and correspondence between entity IDs and literal words in ElementKG, respectively. We then train a language model to learn entity and relation embeddings from this corpus. This process enables the integration of element and functional group knowledge into a unified representation, which facilitates downstream molecular property prediction.

Given a minibatch of N randomly sampled molecules, we create a set of $2N$ graphs by transforming their molecular graphs $\{G_i\}_{i=1}^{N}$ into augmented graphs $\{\tilde{G}_i\}_{i=1}^{N}$ using element-guided graph augmentation. Following refs. 12,22, we treat the $2(N-1)$ graphs other than the positive pair within the same minibatch as negatives, where a positive pair consists of the original molecular graph G_i and its augmented molecular graph $\tilde{G}_i$. We apply a graph encoder $f(\cdot)$ to extract graph embeddings $\{\boldsymbol{h}_{G_i}\}_{i=1}^{N}$ and $\{\boldsymbol{h}_{\tilde{G}_i}\}_{i=1}^{N}$ from the two graph views, and a non-linear projection network $g(\cdot)$ to map these embeddings into a space where the contrastive loss is applied, resulting in two new representations $\{\boldsymbol{z}_{G_i}\}_{i=1}^{N}$ and $\{\boldsymbol{z}_{\tilde{G}_i}\}_{i=1}^{N}$. Finally, a contrastive loss is used to maximize the consistency between positive pairs while minimizing the agreement between negative pairs. For further details, refer to Methods.

Prompt-enhanced fine-tuning. After pre-training, the molecular graph encoder needs to be fine-tuned for downstream property prediction. Specifically, the input molecular graph G is fed into the pre-trained graph encoder $f(\cdot)$ to extract the graph embedding $\boldsymbol{h}_G$, which is then fed into the predictor to output the property value. To bridge the gap between the pre-training contrastive tasks and downstream tasks, we propose to use functional group knowledge as prompts to stimulate the pre-trained graph encoder.

As shown in Fig. 1c, we generate the functional prompt from the functional group knowledge of ElementKG. First, we detect all functional groups in the input molecule, retrieve their corresponding entity embeddings in ElementKG and construct a mediator with a learnable embedding to capture the importance of each functional group. We then apply a self-attention mechanism to the embedding of the mediator (coloured in red) and the embeddings of the functional group entities to comprehensively aggregate their semantics and obtain the functional prompt. Finally, the functional prompt is added to the original representation of each atom node in the input molecular graph with a learnable scale parameter to produce the prompt-enhanced molecular graph, which is then fed into the pre-trained graph encoder and a predictor for molecular property prediction. The technical details of functional prompts are provided in Methods.

KANO boosts the performance of property prediction

Molecular properties of interest can vary widely in scale, ranging from macroscopic influences on the human body to microscopic electronic properties, such as drug side-effects[23], the ability to inhibit human immunodeficiency virus (HIV) replication[24] and hydration free energy[3]. To assess the effectiveness of KANO, we evaluated its performance on datasets in four categories: physiology, biophysics, physical chemistry and quantum mechanics. For more information on the datasets and baselines, please refer to Supplementary Information.

Tables 1 and 2 present the results of various supervised and SSL methods. #Molecules represents the number of molecules in each dataset, and #Tasks indicates the number of binary prediction tasks in each dataset.

Table 1 | Test performance of different models on eight classification benchmarks of physiology and biophysics. The first five models are supervised learning methods, while the last eight are self-supervised methods. The mean and standard deviation of test ROC-AUC (%) on three independent runs are reported

Category	Physiology					Biophysics		
Dataset	**BBBP**	**Tox21**	**ToxCast**	**SIDER**	**ClinTox**	**BACE**	**MUV**	**HIV**
Number of molecules	**2,039**	**7,831**	**8,575**	**1,427**	**1,478**	**1,513**	**93,807**	**41,127**
Number of tasks	**1**	**12**	**617**	**27**	**2**	**1**	**17**	**1**
GCN[47]	71.8±0.9	70.9±0.3	65.0±6.1	53.6±0.3	62.5±2.8	71.6±2.0	71.6±4.0	74.0±3.0
GIN[48]	65.8±4.5	74.0±0.8	66.7±1.5	57.3±1.6	58.0±4.4	70.1±5.4	71.8±2.5	75.3±1.9
MPNN[49]	91.3±4.1	80.8±2.4	69.1±3.0	59.5±3.0	87.9±5.4	81.5±1.0	75.7±1.3	77.0±1.4
DMPNN[50]	91.9±3.0	75.9±0.7	63.7±0.2	57.0±0.7	90.6±0.6	85.2±0.6	78.6±1.4	77.1±0.5
CMPNN[26]	92.7±1.7	80.1±1.6	70.8±1.3	61.6±0.3	89.8±0.8	86.7±0.2	79.0±2.0	78.2±2.2
N-GRAM[46]	91.2±0.3	76.9±2.7	-	63.2±0.5	87.5±2.7	79.1±1.3	76.9±0.7	78.7±0.4
Hu et.al[7]	70.8±1.5	78.7±0.4	65.7±0.6	62.7±0.8	72.6±1.5	84.5±0.7	81.3±2.1	79.9±0.7
MGSSL[10]	70.5±1.1	76.4±0.4	64.1±0.7	61.8±0.8	80.7±2.1	79.7±0.8	78.7±1.5	79.5±1.1
GEM[9]	88.8±0.4	78.1±0.4	68.6±0.2	63.2±1.5	90.3±0.7	87.9±1.1	75.3±1.5	81.3±0.3
GROVER[8]	86.8±2.2	80.3±2.0	56.8±3.4	61.2±2.5	70.3±13.7	82.4±3.6	67.3±1.8	68.2±1.1
GraphMVP[51]	72.4±1.6	75.9±0.5	63.1±0.4	63.9±1.2	79.1±2.8	81.2±0.9	77.7±0.6	77.0±1.2
MolCLR[11]	73.3±1.0	74.1±5.3	65.9±2.1	61.2±3.6	89.8±2.7	82.8±0.7	78.9±2.3	77.4±0.6
$MolCLR_{CMPNN}$	72.4±0.7	78.4±2.6	69.1±1.2	59.7±3.4	88.0±4.0	85.0±2.4	74.5±2.1	77.8±5.5
KANO	**96.0±1.6**	**83.7±1.3**	**73.2±1.6**	**65.2±0.8**	**94.4±0.3**	**93.1±2.1**	**83.7±2.3**	**85.1±2.2**

*Note that the N-GRAM model on ToxCast is too time consuming to finish in time, and its results are not presented. The best-performing results are marked in bold.

Table 2 | Test performance of different models on six regression benchmarks of physical chemistry and quantum mechanics. The first five models are supervised learning methods, and the last six are self-supervised methods. The mean and standard deviation of test root mean square error (for ESOL, FreeSolv and Lipophilicity) or mean absolute error (for QM7, QM8 and QM9) on three independent runs are reported

Category	Physical chemistry			Quantum mechanics		
Dataset	**ESOL**	**FreeSolv**	**Lipophilicity**	**QM7**	**QM8**	**QM9**
Number of molecules	**1,128**	**642**	**4,200**	**7,160**	**21,786**	**133,885**
Number of tasks	**1**	**1**	**1**	**1**	**12**	**3**
GCN[47]	1.431±0.050	2.870±0.135	0.712±0.049	122.9±2.2	0.0366±0.000	0.00835±0.00001
GIN[48]	1.452±0.020	2.765±0.180	0.850±0.071	124.8±0.7	0.0371±0.001	0.00824±0.00004
MPNN[49]	1.167±0.430	1.621±0.952	0.672±0.051	111.4±0.9	0.0148±0.001	0.00522±0.00003
DMPNN[50]	1.050±0.008	1.673±0.082	0.683±0.016	103.5±8.6	0.0156±0.001	0.00514±0.00001
CMPNN[26]	0.798±0.112	1.570±0.442	0.614±0.029	75.1±3.1	0.0153±0.002	0.00405±0.00002
N-GRAM[46]	1.100±0.030	2.510±0.191	0.880±0.121	125.6±1.5	0.0320±0.003	0.00964±0.00031
Hu et.al[7]	1.100±0.006	2.764±0.002	0.739±0.003	113.2±0.6	0.0215±0.001	0.00922±0.00004
GEM[9]	0.813±0.028	1.748±0.114	0.674±0.022	60.0±2.7	0.0163±0.001	0.00562±0.00007
GROVER[8]	1.423±0.288	2.947±0.615	0.823±0.010	91.3±1.9	0.0182±0.001	0.00719±0.00208
MolCLR[11]	1.113±0.023	2.301±0.247	0.789±0.009	90.9±1.7	0.0185±0.013	0.00480±0.00003
$MolCLR_{CMPNN}$	0.911±0.082	2.021±0.133	0.875±0.003	89.8±6.3	0.0179±0.001	0.00475±0.00001
KANO	**0.670±0.019**	**1.142±0.258**	**0.566±0.007**	**56.4±2.8**	**0.0123±0.000**	**0.00320±0.00001**

*The best-performing results are marked in bold.

Table 1 reports the test receiver operating characteristic-area under curve (ROC-AUC,%) on classification tasks in physiology and biophysics. Key observations include: (1) KANO consistently outperforms other methods on all eight datasets, with a significant improvement of 3.79%, showcasing its effectiveness. (2) KANO performs well on multiple-task learning datasets such as Tox21, ToxCast, SIDER and MUV. In particular, KANO achieves a 3.39% improvement on the ToxCast dataset with 617 binary classification tasks. The robust performance indicates that its representations cover diverse molecular semantics.

Table 2 presents the test performance of regression tasks in physical chemistry and quantum mechanics. The key observations

https://doi.org/10.1038/s42256-023-00654-0

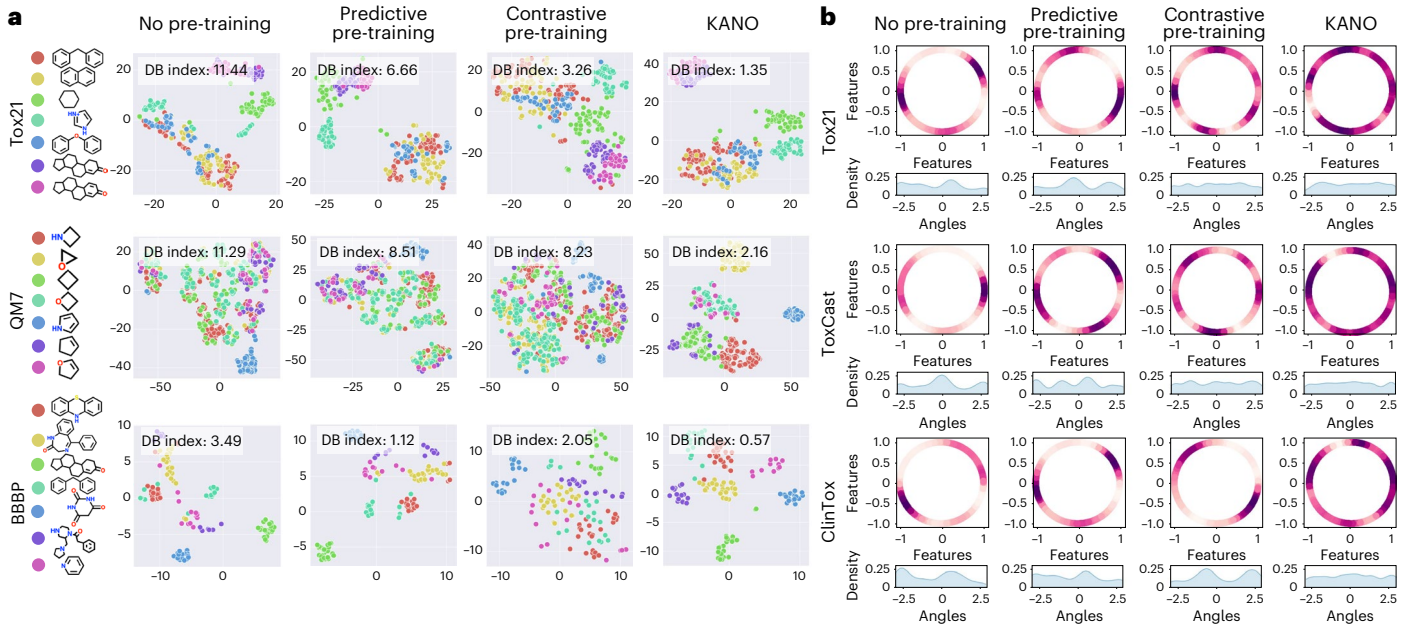


Fig. 3 | Alignment and uniformity analysis. a, Alignment analysis. We show t-SNE visualization of molecular representations to investigate the similarity of molecules with the same scaffold. Different colours represent different scaffolds, with a lower DB index indicating better clustering separation. **b**, Uniformity analysis. Molecular feature distributions are plotted with Gaussian KDE in $\mathbb{R}^2$ (darker colours indicate more points fall in the region), along with KDE on angles (that is, arctan2(y, x) for each point $(x, y) \in S^1$) for a clearer presentation.

are as follows: (1) KANO receives top scores among supervised and self-supervised models, surpassing previous records by a relative improvement of 15.8% on all six regression tasks. (2) KANO's fine-grained chemical understanding helps it achieve remarkable accuracy on quantum mechanical datasets, even surpassing models that incorporate additional 3D information[9]. (3) KANO greatly helps tasks with limited label information, as evidenced by the average improvement of 21.7% on small datasets ESOL and FreeSolv with only 1,128 and 642 labelled molecules, respectively.

In summary, KANO outperforms other models in all benchmarks, demonstrating the effectiveness of integrating ElementKG into the pre-training and fine-tuning stages. KANO not only outperforms other SSL methods but also demonstrates its superiority over supervised methods, providing a competitive advantage for generalization to a broader chemical space.

Richer knowledge in KG leads to more robust representations

ElementKG is essential in the KANO framework as it guides molecular augmentation and functional prompt generation. To determine the contributions of its various components, we evaluate KANO's performance using different KG components, such as class hierarchy, data property and functional group knowledge. We only prune ElementKG's components during pre-training and keep the experimental settings for fine-tuning consistent with the original KANO approach.

Extended Data Fig. 1a reveals that: (1) KANO with the complete ElementKG architecture ('complete ElementKG') outperforms the other versions across all datasets, highlighting the indispensability of each component. (2) Removing class hierarchy ('w/o class hierarchy') results in performance degradation, accentuating the significance of class division and transitive relations between subclasses in refining and transferring fundamental domain knowledge. (3) Excluding functional groups from ElementKG ('w/o functional group') causes a noticeable drop in performance, underscoring the critical role of functional groups. (4) Excluding data properties of entities ('w/o data properties') almost always perform the worst, emphasizing the importance of chemical attributes.

To further investigate the impact of data properties, which each element contains more than 15 of, we mask a certain proportion of them and report the test performance on four categories of tasks. Extended Data Fig. 1b shows the test results for varying keeping rates of data properties. Notably, the model's performance consistently improves as the proportion of retained properties increases, verifying that richer data properties provide more comprehensive fundamental knowledge and consequently enable the learning of more robust molecular representations.

Contrastive learning produces a high-quality feature space

The quality of a representation space can be evaluated by two key properties: alignment and uniformity[25]. The former indicates that similar samples should be mapped to nearby embeddings, while the latter suggests that feature vectors should be uniformly distributed on the unit hypersphere, preserving as much data information as possible. In Fig. 3, we compare the molecular representations produced by our method with those obtained by other methods, including a supervised model (CMPNN[26]), a representative predictive method (GROVER[8]) and a contrastive method with universal augmentation strategy (MolCLR$_{\text{CMPNN}}$[11]).

Alignment analysis. We visualize representations of the molecules with different scaffolds by t-distributed stochastic neighbour embedding (t-SNE)[27] to test whether molecules with the same scaffold would have similar representations. The scaffold, which represents the core structure of a molecule, is a fundamental concept in chemistry and provides a basis for systematic investigations of molecular cores and building blocks[28]. Molecules with different scaffolds typically have very different chemical properties. We choose the seven most common scaffolds from each dataset (Tox21, QM7 and BBBP) and distinguish the scaffolds with different colours. As shown in Fig. 3a, the model without pre-training cannot distinguish molecules with these scaffolds, and the predictive and contrastive methods show only slight improvement. In contrast, KANO produces more distinctive clusters with the lowest Davies–Bouldin (DB) index.

https://doi.org/10.1038/s42256-023-00654-0

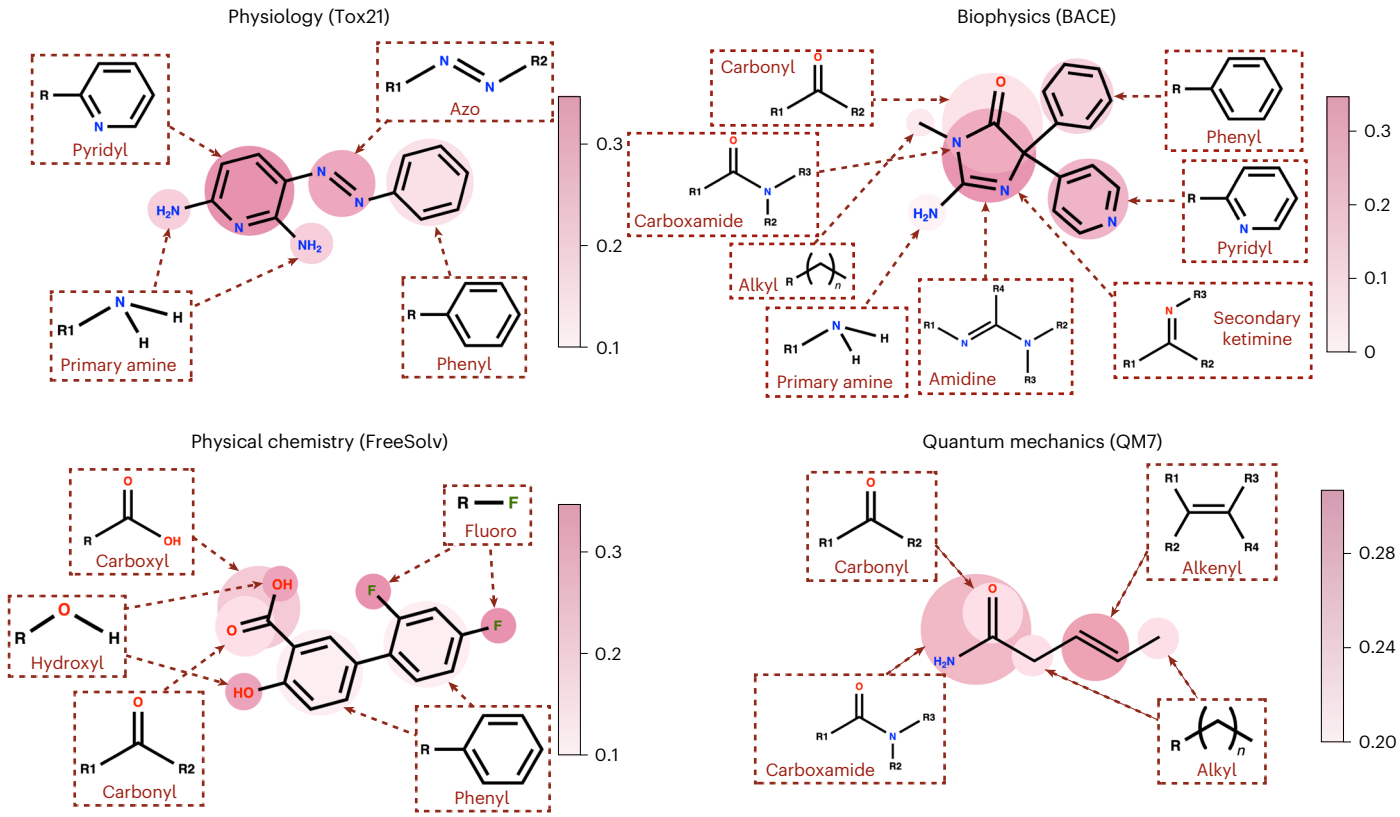


Fig. 4 | Investigation of interpretability of functional prompts. Attention visualization examples of different functional groups in four data categories. The attention weights reflect the functional groups' significance to the global characteristics of the molecule, extracted from the final self-attention layer and normalized. Darker colours indicate higher attention weights.

Uniformity analysis. To examine the uniformity of the learned molecular representations, we first map them onto the unit hypersphere $\mathcal{S}^1$ using t-SNE[27], and then visualize the density distributions of the representations on $\mathcal{S}^1$ using non-parametric Gaussian kernel density estimation (KDE)[29] in $\mathbb{R}^2$. We also show the density estimations of angles for each point on $\mathcal{S}^1$ to present the results more clearly. Figure 3b illustrates the feature and density distributions of the molecular representations learned by our model and the three baselines on the Tox21, ToxCast and ClinTox datasets. In the first three columns, the distributions of the representations are relatively highly clustered with sharp density distributions. In the last column, the distribution becomes more uniform, and the density estimation curves are markedly less sharp.

From Fig. 3, we observe that our model can map molecules with the same scaffold to similar representations, and the pre-trained representations have a more uniform distribution than the baselines. Our ElementKG and KG-guided contrastive learning framework enable KANO to capture globally intrinsic molecular characteristics by normalizing the filtering of knowledge and perceiving global structural insights. Supplementary Information provides additional visualizations of KANO pre-trained representations.

Functional prompts enable explainable predictions

In Extended Data Fig. 2, we compared KANO's performance with functional prompts with that without prompts and evaluated two alternative architectures that integrated functional group knowledge through adding and concatenating to each atom. Results show that the model with functional prompts performs better than the one without, with an 8.41% relative improvement. Furthermore, adding and concatenating functional group features were proven to be suboptimal choices, emphasizing the effectiveness of functional prompts.

Since functional prompts act as a bridge between pre-training contrastive tasks and downstream molecular property prediction tasks, we are interested in their potential to provide domain-specific interpretability. We visualize the attention weights of functional groups in molecular graphs from four property categories in Fig. 4. (1) The first example is from the Tox21 (ref. 30) public database, which measures the toxicity of compounds. We observe higher attention weights for pyridyl and azo functional groups, followed closely by primary amine. Interestingly, pyridyl and primary amine groups can combine to form 2,6-diaminopyridine, a major component of secondary hepatotoxins and skin sensitizers[31]. Azo-containing compounds, such as azo dyes, exhibit carcinogenic and mutagenic properties, making them highly significant[32]. (2) The second example is a human β-secretase 1 (BACE-1) inhibitor from the BACE dataset[33]. The molecule assigns more attention to amidine, carboxamide and secondary ketimine, which form the imidazole component. In addition, pyridyl and phenyl also receive more attention. These findings align with previous research[34,35], suggesting that the aromatic heterocycle family inhibits BACE-1. (3) The third sample is from FreeSolv[36], which focuses on the hydration free energy of small molecules in water. Fluoro and hydroxyl groups receive higher attention due to fluoro's strong electron-acquiring ability and hydroxyl's hydrophilicity, affecting the molecule's interaction force with water. Additionally, carboxyl groups with strong polarity receive more attention weights. (4) The final molecule is from QM7 (ref. 37), recording the atomization energies of molecules. Alkenyl and carboxamide groups receive more attention due to the higher bond energy of the carbon–carbon double bond and the stability of the amide bond, requiring more energy to break them apart into separate atoms. The interpretability exploration illustrates how functional prompts bridge the gap between pre-training tasks and downstream tasks by invoking

relevant functional group knowledge from the molecular property prediction task perspective.

Conclusion

In this study, we presented KANO, a novel approach that enhances molecular property prediction tasks by incorporating chemical domain knowledge. KANO achieved superior performance on 14 molecular benchmarks by leveraging ElementKG, a KG that organizes the knowledge of elements and functional groups. KG-guided pre-training allowed KANO to obtain a high-quality molecular representation space, while functional prompts captured meaningful chemical substructures relevant to downstream tasks.

While KANO has shown promising performance, it may still have some limitations. For instance, ElementKG may not fully capture molecular system complexity, and the current functional prompts may not be able to capture long-range interactions between substructures. To address these limitations, we suggest several interesting future directions. Firstly, extending ElementKG to cover other areas of chemistry and integrating it with other existing KGs could provide a more comprehensive understanding of molecular systems. Secondly, studying the interpretability of KANO's learned representations and the chemical knowledge captured by the functional prompts could provide insights for molecular design and optimization. Finally, exploring the possibility of combining KANO with other techniques to improve its performance on small datasets and accelerate drug discovery could be a promising direction to pursue.

Methods

ElementKG construction and representation

We constructed ElementKG by integrating knowledge from the Periodic Table and Wikipedia pages, providing a holistic view of the element class hierarchy, the chemical attributes of elements and functional groups, and the relations between them. The detailed construction process is shown in Fig. 2 and described below.

First, we extracted the class hierarchy from the collected knowledge of elements and functional groups, which serves as the backbone of ElementKG. As shown in the upper part of Fig. 2, blue blocks represent different classes and blue arrows reflect the containment or disjoint relations between them. For example, the rdfs:subClassOf construct between the class ReactiveNonmetals and the class Nonmetals means that the set of entities in ReactiveNonmetals is a subset of entities in Nonmetals. Also, every entity in the Ester class is a member of its parent class, GroupContainingOxygen. It is important to note that the subclass relations are transitive, implying that the ReactiveNonmetals class is also a subclass of the Element class. However, since literal names can be insufficient to differentiate between different classes, we defined disjointness for the classes and added disjointness axioms using owl:disjointWith. For example, the disjointness between the Metals and Nonmetals classes indicates that an element entity in the Metals class cannot be a member of the Nonmetals class at the same time. Using the class hierarchy, we assigned corresponding entities to each class via rdf:type, with both C and O elements in red blocks being members of the ReactiveNonmetals class.

Second, we compile a list of chemical attributes sourced from the Periodic Table and assign them as data properties to each entity in ElementKG (the dotted block). Over 15 data properties, including hasName, hasAtomic, hasDensity and hasIonization, are associated with each element. On the other hand, for functional groups, we record the type of bonds they contain. For instance, CarboxylhasBondType contains single and double bonds, while Phenyl contains both single and aromatic bonds.

Third, we use object properties (red directional arrows) to model the relationships between entities in ElementKG. To achieve this, we discretize the continuous chemical attribute values of elements and use them as object properties (for example, inRadiusGroup1 and inWeightGroup2) to connect element entities to each other. For instance, the triple (C, inRadiusGroup1, O) indicates that the entities C and O are both in Radius Group 1, while (C, hasStateGas, O) means that they are both in the gaseous state. We add symmetric characteristics to these object properties, which means that (O, hasStateGas, C) also holds when given (C, hasStateGas, O). Since ElementKG is primarily element oriented, we do not directly add object properties to functional groups. Instead, we establish the connection between element and functional group entities through the isPartOf object property, which indicates that the element is involved in the formation of the functional group.

To fully explore the structural and semantic information and obtain meaningful representations of all entities, relations and other components in ElementKG, we employ a KG embedding approach based on OWL2Vec* (ref. 21). As illustrated in Fig. 2b, this approach involves two steps: (1) extracting a corpus from ElementKG, including a structure document, a lexical document and a combined document, and (2) training a language model on the corpus to obtain high-quality KG embeddings[38]. The structure document captures the graph structure and the logical constructors by computing random walks for each target entity and combining the traversed relations and entities into sentences. For example, a random walk of depth 3 starting from the element C would result in the sentence (C, inRadiusGroup1, O, rdf:type, ReactiveNonmetals). The lexical document includes sentences parsed from the structure document. For example, the sentence above can be parsed as ('C', 'in', 'radius', 'group1', 'O', 'type', 'reactive', 'nonmetals'). To establish the correspondence between entities and their literal names, we replace each word in the lexical document with the corresponding entity in the structure document, resulting in a combined document. That is, the example above can be converted to a set of sentences: (C, 'in', 'radius', 'group1', 'O', 'type', 'reactive', 'nonmetals'), ('C', inRadiusGroup1, 'O', 'type', 'reactive', 'nonmetals') and so on. These three documents are merged into a single document, which is then used to train a word2vec[39] model with the skip-gram architecture. Finally, we obtain embeddings for each entity and relation in ElementKG, which we use for input feature initialization of the augmented molecular graph and functional prompt generation.

Contrastive learning framework

We employ a contrastive learning framework to learn the representations of molecular graphs. Given a minibatch of size N, we generate $2N$ graphs by transforming the N original molecular graphs into N augmented molecular graphs. The original molecular graph G_i and its augmented version $\tilde{G}_i$ constitute a positive pair $(G_i, \tilde{G}_i)$, while $(G_i, G_j)_{j\neq 1}$ and $(G_i, \tilde{G}_j)_{j\neq 1}$ form negative pairs.

After capturing the graph representations using the graph encoders $f(\cdot)$, a non-linear transformation $g(\cdot)$ called the projection network maps both the original and augmented graph representations to a latent space where the contrastive loss is calculated, as proposed in simCLR[40]. We adopt a two-layer perceptron (MLP) to perform the projection. Then, we use the normalized temperature-scaled cross-entropy (NT-Xent) loss function[40] to train the graph encoders to maximize the agreement between positive pairs and the discrepancy between negative pairs.

Let $\mathrm{sim}(z_1, z_2) = \frac{z_1^{\top} z_2}{\|z_1\| \cdot \|z_2\|}$ denote the cosine similarity between ℓ_2 normalized z_1 and z_2. The loss function for a positive pair $(G_i, \tilde{G}_i)$ is defined as

$$\ell_i = -\log \frac{e^{\mathrm{sim}\left(z_{G_i}, z_{\tilde{G}_i}\right)/\tau}}{\sum_{k=1}^{N} \mathbb{1}_{[k\neq i]} \left(e^{\mathrm{sim}\left(z_{G_i}, z_{G_k}\right)/\tau} + e^{\mathrm{sim}\left(z_{G_i}, z_{\tilde{G}_k}\right)/\tau}\right) + \left(e^{\mathrm{sim}\left(z_{G_i}, z_{\tilde{G}_k}\right)/\tau} + e^{\mathrm{sim}\left(z_{G_i}, z_{G_k}\right)/\tau}\right)}, \quad (1)$$

where $\mathbb{1}_{[k\neq i]}$ is an indicator function that evaluates to 1 if $k \neq i$, τ is a temperature parameter and z represents the latent representation.

The numerator of the contrastive loss measures the agreement between the positive pair, while the denominator calculates the sum of the agreement between each graph and the other $2N-1$ graphs. This means that the latent representation $\boldsymbol{z}_{G_i}$ of the original graph should consider the similarity with not only other original graph latent vectors $\{\boldsymbol{z}_{G_k}\}_{k\neq i}$ but also all augmented graphs $\{\boldsymbol{z}_{\tilde{G}_k}\}_{k=1}^{N}$. The latent representation of the augmented graph $\boldsymbol{z}_{\tilde{G}_i}$ also follows the same calculation process. Finally, the loss is computed across all positive pairs in the minibatch.

Prompt generator

To stimulate the pre-trained model to recall the relevant knowledge learned before, we design a prompt generator f_{prompt} to produce a prompt x_{prompt} based on ElementKG and the input molecular graph G, that is, $x_{\text{prompt}} = f_{\text{prompt}}(G, \text{ElementKG})$. We detect all functional groups contained in G using the open-source package RDKit[41] and retrieve the corresponding functional group entities in ElementKG on the basis of their names. Then we obtain the embeddings of functional group entities $\{x_1, \ldots, x_m\}$ using the KG embedding method, where m is the number of detected functional groups. To capture the importance of functional groups, we construct a learnable vector as the mediator (denoted as x_0) and then apply the self-attention mechanism[42] on both the embeddings of the mediator and functional groups. Specifically, the input $X = \{x_0, x_1, \ldots, x_m\}$ is first projected into the query/key/value vector:

$$Q = XW^Q, K = XW^K, V = XW^V, \tag{2}$$

where W^Q, W^K, $W^V \in \mathbb{R}^{d\times d}$ and d is the hidden dimension. The self-attention mechanism calculates the attention weight between queries and keys, and then multiplies by the value. The output embedding is formulated as

$$X' = \text{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d}}\right)V. \tag{3}$$

We implement two self-attention layers and obtain the embedding of the mediator $x'_0 = X'[:, 0]$, which reflects the combined contributions of functional groups with varying importance. We then feed it into a fully connected layer followed by layer normalization[43] to obtain the functional prompt

$$x_{\text{prompt}} = \text{LayerNorm}(W \cdot x'_0). \tag{4}$$

Finally, we add the prompt x_{prompt} to the original representation of each atom node in G with a learnable scale parameter α, resulting in the new input feature of a node v in G expressed as $x_v^{\text{new}} = x_v + \alpha \cdot x_{\text{prompt}}$. We then feed this prompt-enhanced molecular graph into the pre-trained graph encoder, followed by a prediction network for downstream molecular properties.

Graph encoder architecture

A molecular graph can be represented as $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes a set of nodes and $\mathcal{E}$ denotes a set of edges. Each edge is bidirectional. Let x_v denote the initial features of node v, and $x_{e_{(u,v)}}$ as the initial features of edge $e_{(u,v)}$. In particular, for atoms and bonds in the original molecular graph, we extract different initial features for them following specific chemical rules, as detailed in Supplementary Information.

Taking Fig. 1b as an example, for the augmented graph, we take the element entity embeddings obtained above as the initial features of element nodes. The initial feature of an edge between every two element nodes is obtained by mean pooling of the embeddings of multiple relations between the corresponding element entities in ElementKG. Following the same feature extraction method in the original molecular graph, we obtain the initial features of atoms and bonds. The edges between elements and their corresponding atoms are distinguished by different random initialization features, that is, the dashed edges with the same colour represent the same initial features while different colours indicate different representations.

Given the graph structure, node features and edge features, our goal is to learn a graph encoder $f(\cdot)$ that maps the input graph to a vector representation. In our case, we implement CMPNN[26] as the graph encoder, which improves graph embeddings by strengthening the message interactions between edges and nodes.

Firstly, to update the node hidden states, each node $v \in \mathcal{V}$ aggregates representations of their incoming edges instead of its neighbouring nodes in G. The intermediate message vector is obtained as

$$\begin{aligned} m^k(v) &= \text{AGGREGATE}\left(\{h^{k-1}(e_{(u,v)}), \forall u \in \mathcal{N}_v\}\right) \\ &= \sum_{u\in N(v)} h^{k-1}(e_{u,v}) \odot \text{pooling}\left(\sum_{u\in N(v)} h^{k-1}(e_{u,v})\right), \end{aligned} \tag{5}$$

where k denotes the current depth of the message passing, the pooling operator is a max pooling function and $\odot$ is an element-wise multiplication operator. Here we apply max pooling to highlight the edges with the highest information intensity, as the hidden state of a node is mainly based on the strongest message from incoming edges. Then, the node's current hidden state $h^{k-1}(v)$ is concatenated with the message vector $m^k(v)$ and fed through a communicative function to update the node's hidden state $h^k(v)$:

$$\begin{aligned} h^k(v) &= \text{COMMUNICATE}\left(m^k(v), h^{k-1}(v)\right) \\ &= \sigma\left(W^k \cdot \text{CONCAT}\left(h^{k-1}(v), m^k(v)\right)\right), \end{aligned} \tag{6}$$

where the hidden state $h^k(v)$ acts as a message transfer station that receives incoming messages, integrates them and sends them to the next station. The specific communication function is implemented by feeding both the node and edge features into an MLP followed by a rectified linear unit (ReLU) activation.

Secondly, we extract message of the edge $e_{(v,w)}$ by subtracting its inverse edge information from the $h^k(v)$:

$$m^k\left(e_{(v,w)}\right) = h^k(v) - h^{k-1}\left(e_{(w,v)}\right), \tag{7}$$

where $e_{(w,v)}$ is the inverse edge of $e_{(v,w)}$. To update the edge hidden states, we first feed the edge intermediate message $m^k\left(e_{(v,w)}\right)$ into a fully connected layer and add it with the initial edge feature $x_{e_{(u,v)}}$. We apply a ReLU activation function to the output and use it as the intermediate message vector for the next iteration. This procedure can be mathematically expressed as

$$h^k\left(e_{(v,w)}\right) = \sigma\left(x_{e_{(u,v)}} + W \cdot m^k\left(e_{(v,w)}\right)\right). \tag{8}$$

Thirdly, after K iterations, one more round of interaction is applied:

$$m(v) = \text{AGGREGATE}\left(\{h^K\left(e_{(u,v)}\right), \forall u \in N(v)\}\right), \tag{9}$$

then the final node representation $h(v)$ of the graph is obtained by gathering the message from incoming edges, the current node representation and the initial node feature:

$$h(v) = \text{COMMUNICATE}\left(m(v), h^K(v), x_v\right). \tag{10}$$

Finally, a readout operator is applied to get the whole graph representation:

$$h_G = \sum_{v\in\mathcal{V}} \text{GRU}(h(v)), \tag{11}$$

where GRU is the gated recurrent unit introduced in ref. 44.

Experimental setup

Pre-training and downstream dataset. In the pre-training phase, we pre-train KANO using 250,000 unlabelled molecules sampled from ZINC15 (ref. 4), a public access database containing purchasable drug-like compounds. In the fine-tuning phase, we use 14 benchmark datasets from MoleculeNet[5], comprising 678 binary classification tasks and 19 regression tasks. The datasets cover molecular data from a wide range of domains, such as drugs, biology, physics and chemistry. We perform three independent runs on three random-seeded scaffold splitting for all datasets, except QM9, with a train/validation/test ratio of 8:1:1. Scaffold splitting[45] is a more challenging splitting method that splits molecules according to their scaffolds (molecular substructures) and can better evaluate the generalization ability of the models on out-of-distribution data samples. For the QM9 dataset, we follow the random splitting setting of most related works[11,46] for comparison. Supplementary Information contains more details about the datasets.

Implementation details. Since the raw data are in the form of molecular SMILES, which is a line notation for describing the structure of chemical species using short ASCII strings, we utilize the open-source chemical analysis tool RDKit to convert them into 2D molecular graphs and extract the atom and bond features. The initial features of atoms are determined by their associated eight attributes (for example, chirality, hybridization and atomic mass), and the bonds are embedded by their four related attributes (for example, bond type and conjugated), as detailed in Supplementary Information.

In contrastive pre-training, we utilize the Adam optimizer with a learning rate of 3×10^{-5} to optimize the NT-Xent loss and set the temperature parameter τ to 0.1. We apply an MLP with a ReLU activation function as the projection network. The model is trained with a batch size of 1,024 and 50 epochs.

In prompt-enhanced fine-tuning, we use RDKit to detect the functional groups in each molecule. We apply two self-attention layers on all functional groups and the mediator. The output is fed into a fully connected layer, which is then layer normalized. We adopt a two-layer MLP as the property prediction network. For classification tasks, we utilize the binary cross-entropy (BCE) loss combined with the sigmoid layer (BCEWithLogits loss) when training the graph encoder and the property prediction network, while for regression tasks, we apply the mean squared error loss. The Adam optimizer is applied to the graph encoder with a learning rate ranging from 1×10^{-4} to 1×10^{-3} for all datasets, and the learning rate of the prompt generator is five times that of the graph encoder. We train the model on the training set and search hyper-parameters on the validation set for the best results. The training is set to 100 epochs. We implement fine-tuning of the pre-trained model three times with a batch size of 256 to report the average and standard deviation of performance on the testing set, using ROC-AUC for classification tasks and mean absolute error/root mean square error for regression tasks. KANO is implemented using Pytorch and runs on a Ubuntu Server with NVIDIA GeForce RTX 3090Ti graphics processing units.

Data availability

The ElementKG, pre-training data and molecular property prediction benchmarks used in this work are available in the Code Ocean capsule at https://doi.org/10.24433/CO.5629517.v1 and the GitHub repository at https://github.com/HICAI-ZJU/KANO. Source data are provided with this paper.

Code availability

The source code of this work is freely available in the Code Ocean capsule at https://doi.org/10.24433/CO.5629517.v1 and the GitHub repository https://github.com/HICAI-ZJU/KANO.

References

1. Hay, M., Thomas, D. W., Craighead, J. L., Economides, C. & Rosenthal, J. Clinical development success rates for investigational drugs. *Nat. Biotechnol.* **32**, 40–51 (2014).
2. Dowden, H. & Munro, J. Trends in clinical success rates and therapeutic focus. *Nat. Rev. Drug Discov.* **18**, 495–496 (2019).
3. Gaulton, A. et al. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.* **40**, 1100–1107 (2012).
4. Sterling, T. & Irwin, J. J. ZINC 15—ligand discovery for everyone. *J. Chem. Inf. Model.* **55**, 2324–2337 (2015).
5. Wu, Z. et al. Moleculenet: a benchmark for molecular machine learning. *Chem. Sci.* **9**, 513–530 (2018).
6. Kim, S. et al. Pubchem 2019 update: improved access to chemical data. *Nucleic Acids Res.* **47**, D1102–D1109 (2019).
7. Hu, W. et al. Strategies for pre-training graph neural networks. In *Proc. 8th International Conference on Learning Representations* (OpenReview.net, 2020).
8. Rong, Y. et al. in *Advances in Neural Information Processing Systems 33* (eds Larochelle, H. et al.) 12559–12571 (Curran Associates, 2020).
9. Fang, X. et al. Geometry-enhanced molecular representation learning for property prediction. *Nat. Mach. Intell.* **4**, 127–134 (2022).
10. Zhang, Z., Liu, Q., Wang, H., Lu, C. & Lee, C. in *Advances in Neural Information Processing Systems 34* (eds Ranzato, M. et al.) 15870–15882 (Curran Associates, 2021).
11. Wang, Y., Wang, J., Cao, Z. & Farimani, A. B. Molecular contrastive learning of representations via graph neural networks. *Nat. Mach. Intell.* **4**, 279–287 (2022).
12. You, Y. et al. in *Advances in Neural Information Processing Systems 33* (eds Larochelle, H. et al.) 5812–5823 (Curran Associates, 2020).
13. Liu, P. et al. Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.* **55**, 1–35 (2023).
14. Brown, T. et al. in *Advances in Neural Information Processing Systems 33* (eds Larochelle, H. et al.) 1877–1901 (Curran Associates, 2020).
15. Sainz, O., de Lacalle, O. L., Labaka, G., Barrena, A. & Agirre, E. Label verbalization and entailment for effective zero and few-shot relation extraction. In *Proc. 2021 Conference on Empirical Methods in Natural Language Processing* (eds Moens, M.-F. et al.) 1199–1212 (Association for Computational Linguistics, 2021).
16. Ye, H. et al. Learning to ask for ata-efficient event argument extraction (student abstract). *Proc. AAAI Conference on Artificial Intelligence* **36**, 13099–13100 (2022).
17. Tsimpoukelli, M. et al. in *Advances in Neural Information Processing Systems 34* (eds Ranzato, M. et al.) 200–212 (Curran Associates, 2021).
18. Ertl, P., Altmann, E. & McKenna, J. M. The most common functional groups in bioactive molecules and how their popularity has evolved over time. *J. Med. Chem.* **63**, 8408–8418 (2020).
19. Delmas, M. et al. Building a knowledge graph from public databases and scientific literature to extract associations between chemicals and diseases. *Bioinformatics* **37**, 3896–3904 (2021).
20. Lin, X., Quan, Z., Wang, Z., Ma, T. & Zeng, X. KGNN: knowledge graph neural network for drug–drug interaction prediction. In *Proc. Twenty-Ninth International Joint Conference on Artificial Intelligence* (ed. Bessiere, C) 2739–2745 (International Joint Conferences on Artificial Intelligence Organization, 2020).
21. Chen, J. et al. Owl2vec*: embedding of OWL ontologies. *Mach. Learn.* **110**, 1813–1845 (2021).

22. Sun, M., Xing, J., Wang, H., Chen, B. & Zhou, J. MoCL: data-driven molecular fingerprint via knowledge-aware contrastive learning from molecular graph. In *Proc. 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (eds Feida, Z. et al.) 3585–3594 (ACM, 2021).
23. Kuhn, M., Letunic, I., Jensen, L. J. & Bork, P. The SIDER database of drugs and side effects. *Nucleic Acids Res.* **44**, 1075–1079 (2016).
24. Riesen, K. & Bunke, H. IAM graph database repository for graph based pattern recognition and machine learning. In *Structural, Syntactic, and Statistical Pattern Recognition. SSPR /SPR 2008.* Lecture Notes in Computer Science, Vol. 5342 (eds Lobo, N. V. et al.) 287–297 (Springer, 2008).
25. Wang, T. & Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proc. 37th International Conference on Machine Learning* (eds Daumé, H. III & Sing, A.) 9929–9939 (PMLR, 2020).
26. Song, Y. et al. Communicative representation learning on attributed molecular graphs. In *Proc. Twenty-Ninth International Joint Conference on Artificial Intelligence* (ed Bessiere, C.) 2831–2838 (International Joint Conferences on Artificial Intelligence Organization, 2020).
27. van der Maaten, L. Accelerating t-SNE using tree-based algorithms. *J. Mach. Learn. Res.* **15**, 3221–3245 (2014).
28. Bemis, G. W. & Murcko, M. A. The properties of known drugs. 1. Molecular frameworks. *J. Med. Chem.* **39**, 2887–2893 (1996).
29. Botev, Z. I., Grotowski, J. F. & Kroese, D. P. Kernel density estimation via diffusion. *Ann. Stat.* **38**, 2916–2957 (2010).
30. Hartung, T. Toxicology for the twenty-first century. *Nature* **460**, 208–212 (2009).
31. Fitzpatrick, R. B. Haz-map: information on hazardous chemicals and occupational diseases. *Med. Ref. Serv. Q.* **23**, 49–56 (2004).
32. Puvaneswari, N., Muthukrishnan, J. & Gunasekaran, P. Toxicity assessment and microbial degradation of az dyes. *Indian J. Exp. Biol.* **44**, 618–626 (2006).
33. Subramanian, G., Ramsundar, B., Pande, V. & Denny, R. A. Computational modeling of β-secretase 1 (BACE-1) inhibitors using ligand based approaches. *J. Chem. Inf. Model.* **56**, 1936–1949 (2016).
34. Mureddu, L. G. & Vuister, G. W. Fragment-based drug discovery by NMR. Where are the successes and where can it be improved? *Front. Mol. Biosci.* **9**, 110 (2022).
35. García Marín, I. D. et al. New compounds from heterocyclic amines scaffold with multitarget inhibitory activity on aβ aggregation, ache, and bace1 in the alzheimer disease. *PLoS ONE* **17**, e0269129 (2022).
36. Mobley, D. L. & Guthrie, J. P. Freesolv: a database of experimental and calculated hydration free energies, with input files. *J. Comput. Aided Mol. Design* **28**, 711–720 (2014).
37. Blum, L. C. & Reymond, J.-L. 970 million druglike small molecules for virtual screening in the chemical universe database gdb-13. *J. Am. Chem. Soc.* **131**, 8732–8733 (2009).
38. Li, Y. & Yang, T. in *Guide to Big Data Applications* (ed. Srinivasan, S.) 83–104 (Springer, 2018).
39. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. in *Advances in Neural Information Processing Systems 26* (eds Burges, C. J. et al.) 3111–3119 (Curran Associates, 2013).
40. Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. E. A simple framework for contrastive learning of visual representations. In *Proc. 37th International Conference on Machine Learning* (eds Daumé, H. III & Singh, A.) 1597–1607 (PMLR, 2020).
41. Landrum, G. Rdkit documentation. *Release* **1**, 4 (2013).
42. Shang, C. et al. Edge attention-based multi-relational graph convolutional networks. Preprint at https://arxiv.org/abs/1802.04944 (2018).
43. Ba, L. J., Kiros, J. R. & Hinton, G. E. Layer normalization. Preprint at http://arxiv.org/abs/1607.06450 (2016).
44. Cho, K., van Merrienboer, B., Bahdanau, D. & Bengio, Y. On the properties of neural machine translation: encoder-decoder approaches. In *Proc. SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation* (eds Wu, D. et al.) 103–111 (Association for Computational Linguistics, 2014).
45. Ramsundar, B., Eastman, P., Walters, P. & Pande, V. *Deep Learning for the Life Sciences: Applying Deep Learning to Genomics, Microscopy, Drug Discovery, and More* (O'Reilly Media, 2019).
46. Liu, S., Demirel, M. F. & Liang, Y. in *Advances in Neural Information Processing Systems 32* (eds Wallach, H. et al.) (Curran Associates, 2019).
47. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. In *Proc. 5th International Conference on Learning Representations* (OpenReview.net, 2017).
48. Xu, K., Hu, W., Leskovec, J. & Jegelka, S. How powerful are graph neural networks? In *Proc. 7th International Conference on Learning Representations* (OpenReview.net, 2019).
49. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural message passing for quantum chemistry. In *Proc. 34th International Conference on Machine Learning* 70 (eds Doina, P. & Teh, Y. W.) 1263–1272 (PMLR, 2017).
50. Yang, K. et al. Are learned molecular representations ready for prime time? Preprint at http://arxiv.org/abs/1904.01561 (2019).
51. Liu, S. et al. Pre-training molecular graph representation with 3d geometry. In *Proc. Tenth International Conference on Learning Representations* (OpenReview.net, 2022).

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFCU19B2027 and NSFC91846204, H.C.), Donghai Lab's joint project (DH-2022ZY0012, H.C.), the Innovation Team and Talents Cultivation Program of the National Administration of Traditional Chinese Medicine (No. ZYYCXTD-D-202002, X.F.), the Fundamental Research Funds for the Central Universities (no. 226-2022-00226, X.F.) and the CAAI-Huawei MindSpore Open Fund (CAAIXSJLJJ-2022-052A, Q.Z.). We want to express gratitude to the Hangzhou AI Computing Center for their technical support.

Author contributions

Y.F. and H.C. conceived the study. Y.F. developed the method, wrote the code and performed the analysis. Q.Z. polished the paper. Z.C. and X.Z. participated in the development of the algorithm and benchmarked the methods. H.C., Q.Z., N.Z., X.F. and X.S. provided a lot of advice on KG construction and experimental design. All authors wrote the paper and read and approved the final paper.

Competing interests

The authors declare no competing interests.

Additional information

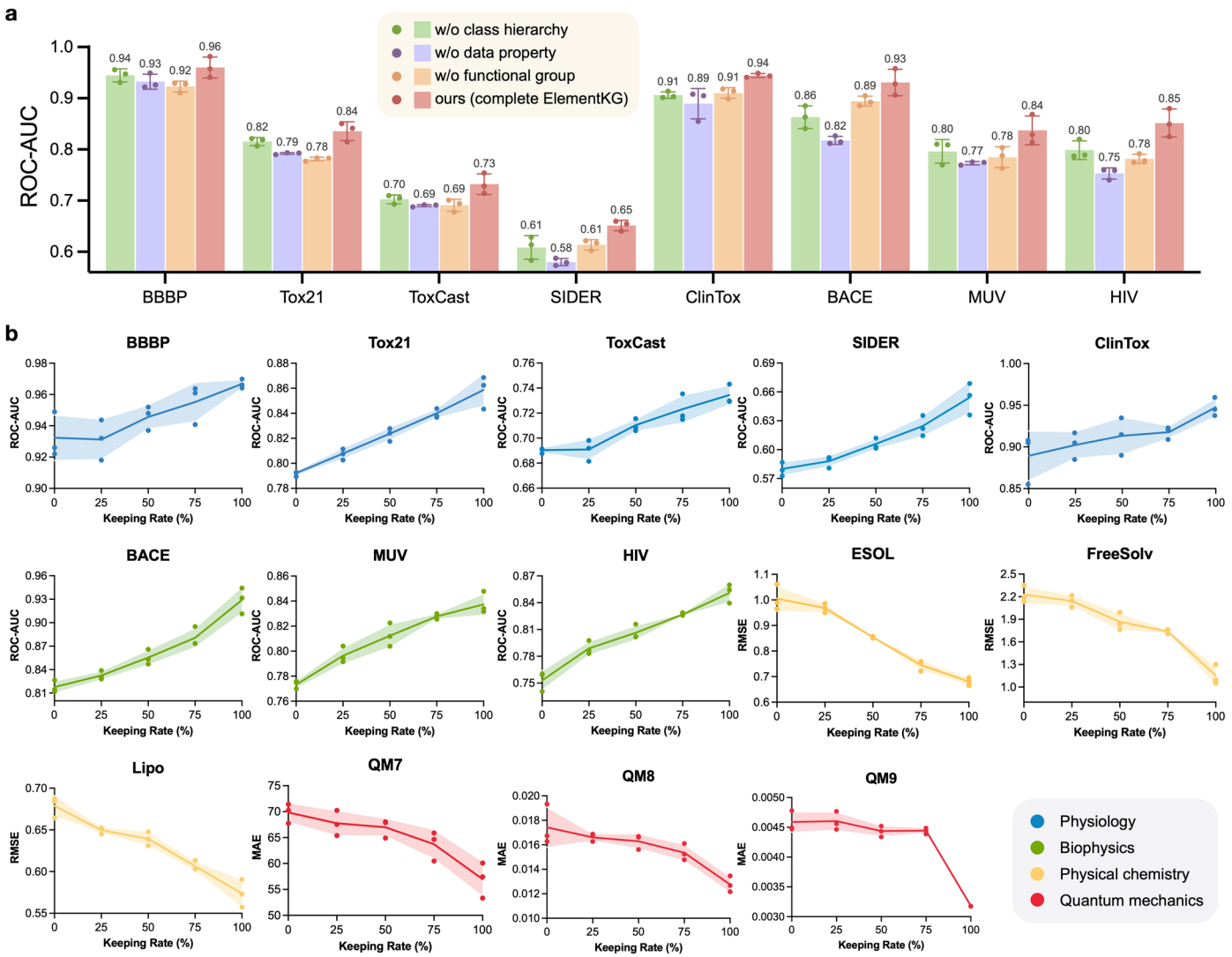
Extended data is available for this paper at https://doi.org/10.1038/s42256-023-00654-0.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-023-00654-0.

Correspondence and requests for materials should be addressed to Qiang Zhang, Xiaohui Fan or Huajun Chen.

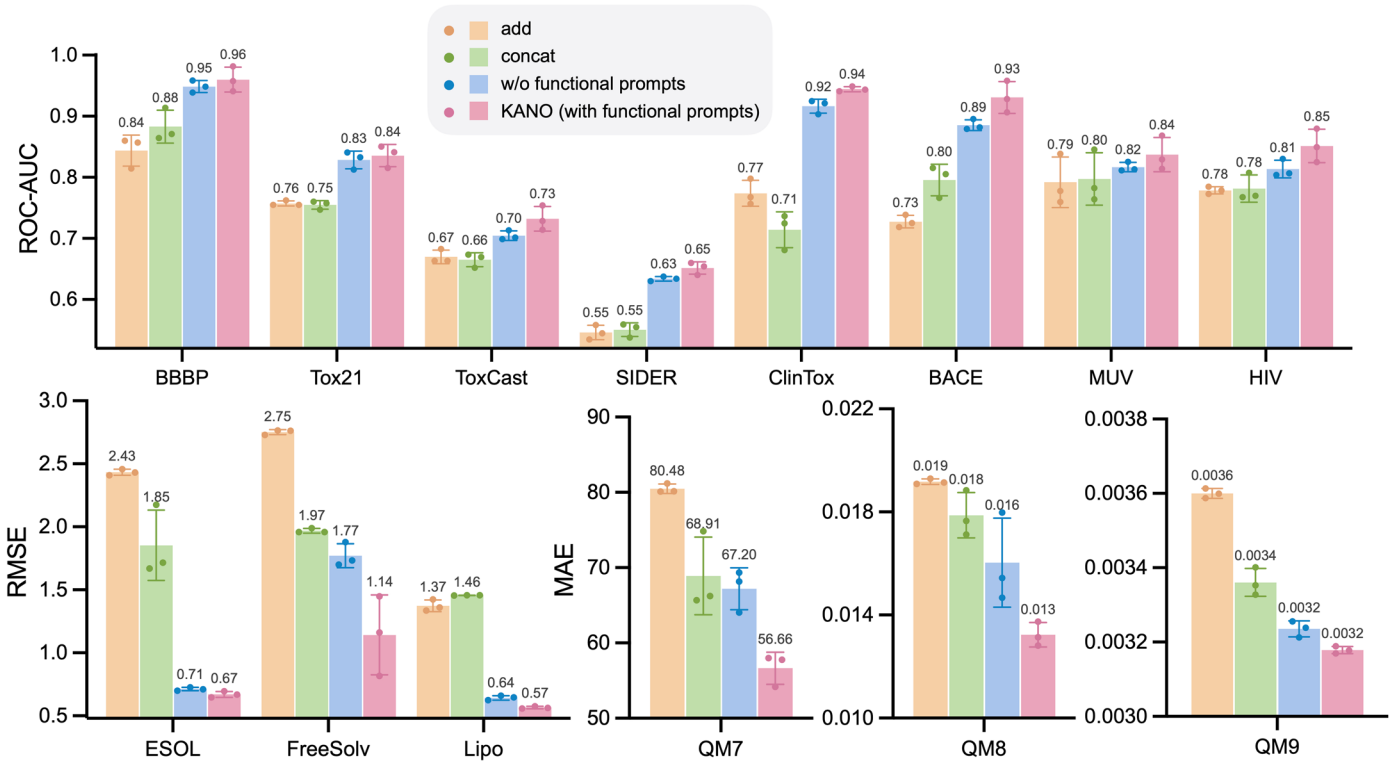
Peer review information *Nature Machine Intelligence* thanks Maria Andreina Francisco Rodriguez and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

https://doi.org/10.1038/s42256-023-00654-0



Extended Data Fig. 1 | Exploration of knowledge abundance in ElementKG. **a**, Performance of KANO with different ElementKG components. Green denotes the removal of class hierarchy from ElementKG, which removes various classes (except for the lowest-level classes directly connected with entities), as well as axioms *rdfs:subClassOf* and *owl:disjointWith*. It consists only of entities, lowest-level classes, data properties, and object properties. Purple denotes the deletion of data properties of each entity. Yellow represents the removal of the entire functional group component, including class hierarchy and entities of functional groups, and their relations with element entities. Red indicates the complete ElementKG with all components. The results are reported as mean values +/- SD on three independent runs. The error bars represent the SD, while the dots represent three individual data points. **b**, Performance of KANO with different keeping rates of data properties in ElementKG. We vary the proportion of data properties of element entities retained in ElementKG and report the corresponding performance trends across datasets in various domains, represented by different colors. The horizontal axis represents the keeping rate, which refers to the proportion of knowledge introduced. The vertical axis represents the performance measured by ROC-AUC on classification tasks (higher is better) and RMSE and MAE on regression tasks (lower is better). The results are reported as mean values +/- SD on three independent runs. The mean is represented by the lines, the SD is depicted by the error bars, and individual data points are marked with dots.

https://doi.org/10.1038/s42256-023-00654-0



Extended Data Fig. 2 | A closer look at functional prompts. Performance comparison of KANO with or without functional prompts, as well as architectures that incorporate functional group knowledge in different ways (addition or concatenation). The yellow bars indicate the addition of functional group knowledge to each atom, while the green bars signify the concatenation of this knowledge to the atom. The blue bars represent KANO without functional prompts, where the input molecules do not contain functional group knowledge from ElementKG. The pink bars represent injecting functional group knowledge to each atom using functional prompts. The results are reported as mean values +/- SD on three independent runs. The error bars represent the SD, while the dots represent three individual data points.