IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 June 2023, accepted 17 August 2023, date of publication 24 August 2023, date of current version 30 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3308203

RESEARCH ARTICLE

Predicting Molecule Toxicity via Descriptor-Based Graph Self-Supervised Learning

XINZE LI[1], ILYA MAKAROV[2,3], AND DMITRII KISELEV[1,2]
[1]School of Data Analysis and Artificial Intelligence, HSE University, 101000 Moscow, Russia
[2]Artificial Intelligence Research Institute (AIRI), 105064 Moscow, Russia
[3]AI Center, NUST MISiS, 119049 Moscow, Russia

Corresponding authors: Dmitrii Kiselev (kiselev@airi.net) and Ilya Makarov (makarov@airi.net)

The work in Section 3 on descriptor-based graph self-supervised learning prepared by D. Kiselev was made within the framework of the HSE University Basic Research Program. The work in Section 2 on modern graph neural networks prepared by I. Makarov was made under support from the strategic project "Digital Business" within the framework of the Strategic Academic Leadership Program "Priority 2030" at National University of Science and Technology (NUST) MISiS.

ABSTRACT Predicting molecular properties with Graph Neural Networks (GNNs) has recently drawn a lot of attention, with compound toxicity prediction being one of the biggest challenges. In cases where there is insufficient labeled molecule data, an effective approach is to pre-train GNNs on large-scale unlabeled molecular data and then fine-tune them for downstream tasks. Among pre-training strategies, node-level pre-training involves masking and predicting atom properties, while motif-based methods capture rich information in subgraphs. These approaches have shown effectiveness across various downstream tasks. However, current pre-training frameworks face two main challenges: (1) node-level auxiliary tasks do not preserve useful domain knowledge, and (2) the fusion of motif-based methods and node-level tasks is computationally extensive. To address these challenges, we propose Descriptor-based Graph Self-supervised Learning (DGSSL), a method that utilizes domain knowledge to enhance graph representation learning. We extract domain knowledge from a descriptor language known as fragmentary code of substructure superposition (FCSS), where molecules are described using substructures that can serve as centers for weak bonds. Specifically, DGSLL identifies descriptor centers in molecules and encodes motif-like information as special atomic numbers in the pre-training tasks. This enables node-level self-supervised pre-training frameworks for GNNs to also capture rich information in local subgraphs. Experimental results demonstrate that our method achieves state-of-the-art performance on three toxicity-related benchmarks and show their significance in an ablation experiment.

INDEX TERMS Graphs, molecule graphs, graph neural networks, molecule toxicity prediction, self-supervised learning.

I. INTRODUCTION

With the rapid application of deep learning in graph-structured data, a line of works focused on exploiting deep learning methods to accelerate the process of drug discovery, with molecular property prediction being an important branch [1], [2], [3]. The successful application of deep learning in this field can reduce the time-consuming wet-lab experiments, assist researchers in the chemistry domain to optimize candidate molecules, and enable high-throughput drug screening [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Giacomo Fiumara.

In recent years, Graph Neural Networks (GNNs) have shown remarkable success in graph representation learning [5], [6], [7]. Since molecules can be naturally represented by graphs, different variants of GNNs have been widely studied for molecular property prediction [2], [8], [9], [10]. However, GNNs usually have poor generalization capabilities when there is insufficient labeled training data in this domain [3]. Meanwhile, obtaining relevant labeled molecules requires time-consuming and expensive wet-lab experiments, making it difficult to increase labeled data for model training [4].

Recently, self-supervised learning (SSL) has emerged as a popular research topic in natural language processing (NLP) [11], [12], [13] and computer vision (CV) [14], [15], [16],

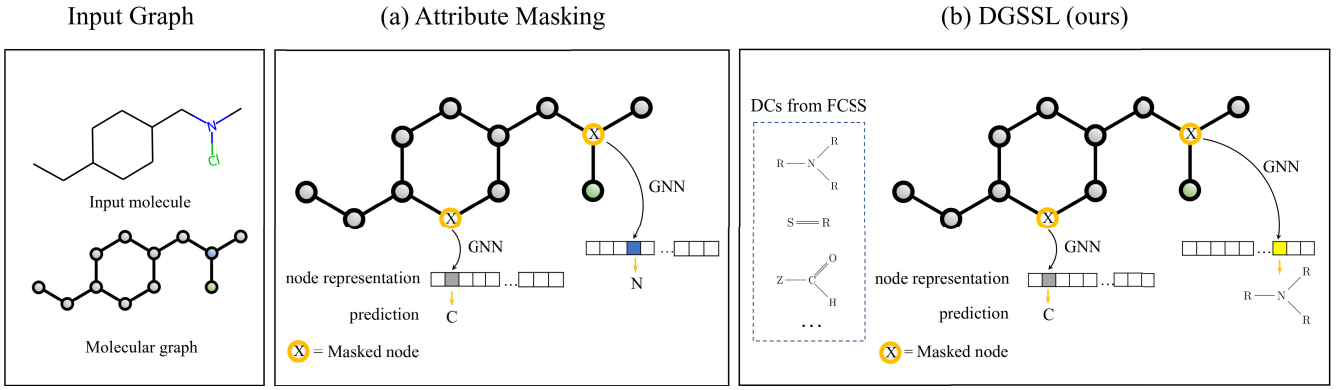


FIGURE 1. Illustration comparing Attribute Masking with our proposed method DGSSL: (a) Attribute Masking requires the GNN to predict the atom type. (b) In DGSSL, the prediction target can be either an atom type or a descriptor center.

[17], [18], [19]. Models are first pre-trained using a large amount of unlabeled data, and then the learned parameters are used to initialize models in downstream tasks, followed by a fine-tuning stage [20], [21]. This approach significantly improves the performance of models in downstream tasks. For example, the language model BERT [11] utilizes a large number of unlabeled texts for masked token prediction tasks. Researchers use various data augmentation methods such as color distortion, scaling, and cropping, along with contrastive methods to enhance performance in visual representation learning [14].

Inspired by the remarkable achievements of SSL in these domains, studies on various model architectures of GNNs in molecular property prediction have slowed down, and interest in studies has gradually shifted towards SSL on graphs. Hu et al. [22] proposed attribute masking methods that randomly mask some attributes of nodes/edges and then predict certain attributes, such as atom types, similar to masked token prediction in the NLP domain. However, Rong et al. [3] argue that serious ambiguity problems exist in this pre-training task since the number of atom types is too small. To address this problem, they construct statistical properties of the local subgraph and assign them to atoms and bonds as contextual properties. Then, instead of predicting atom types, they predict the contextual property in node-level pre-training tasks.

We argue that both methods are suboptimal. The approach proposed by Hu et al. [22] only focuses on predicting the atomic number, while the same atoms may have different chemical semantics along with their local subgraphs. when they are in specific chemical environments. Therefore, only predicting atom types in the pre-training task will make embeddings of atoms with the same atomic number in molecules tend to be consistent. If such knowledge is transferred to downstream tasks, it would be harder for GNNs to capture different semantic information of the same type of atoms. As for Grover [3], the statistical properties of the local subgraph are not necessarily related to chemical semantics, which increases the risk of GNNs considering unnecessary contextual information in downstream tasks.

In this paper, we propose Descriptor-based Graph Self-Supervised Learning (DGSSL). We improve the node-level pre-training task by designing a novel prediction target that combines the useful information in descriptor centers (DCs) and atom types. Specifically, we match DCs in molecules and utilize extra special atomic numbers to encode domain knowledge. We conducted pre-training on the ZINC dataset [23], and experiments have shown that our method achieves state-of-the-art performance on three toxicity-related benchmarks. The implementation is available at https://github.com/li-xinze/GNN-Tox.

II. RELATED WORK

Our work draws inspiration from the fields of molecular machine learning, graph neural networks, and SSL on graphs. In the following subsections, we provide molecular representations and relevant models in the field of molecular property prediction. Then, we delve into the preliminaries of GNNs. Finally, we provide an introduction to SSL methods on graphs.

A. MOLECULAR PROPERTY PREDICTION

Expressive representations of molecules play a significant role in molecular property prediction. Typically, a molecule can be encoded as a line of ASCII strings, a fixed-length vector, or a molecular graph.

For string-based representations (e.g., SMILES [24]), text processing models such as LSTM and Transformer have been employed to predict molecular properties [1], [25]. However, SMILES encoding does not directly capture important topology information in molecules. In cheminformatics, molecules can be encoded into fixed-length vectors using fingerprints or descriptors. Fingerprints focus on encoding structural information in molecules but are not specifically optimized for particular tasks. For example, ECFP [26] assigns an initial integer identifier to each non-hydrogen atom and iteratively updates identifiers of neighboring atoms until a specified diameter is reached. Descriptors consist of structural information or physiochemical properties selected by experts.

Random forests and deep neural networks can be utilized with these fingerprint or descriptor vectors to predict specific properties [27].

More recently, GNNs have been introduced in molecular representation learning, as molecules can be naturally represented as graphs. Reference [2] proposes a message passing framework, while [28] enhances molecular representation by strengthening message interactions between nodes and edges. Additionally, [9] incorporates attention mechanisms at both the atom and molecule levels to learn better embeddings.

B. GRAPH NEURAL NETWORKS (GNNs)

GNNs have been proposed in recent years as a means to learn effective representations for graph data. A general message passing framework can summarize most GNNs. In a GNN layer, each node will first aggregate information from its neighboring atoms and edges, then update its representation. Specifically, there are two key operations in GNNs: aggregate and update.

The aggregate operation can be further divided into two functions: the message function and the reduce function. At each layer, a message is generated on each edge using the message function. Subsequently, each node collects messages from its connected edges and reduces them using the reduce function, which can take the form of sum, mean, max, or even a neural network. Finally, the update function adjusts the node's representation using the aggregated messages and its own representation from the previous layer. In layer l, this procedure can be formally expressed as follows,

$$\begin{aligned} m_e^{(l+1)} &= \phi(x_v^{(l)}, x_u^{(l)}, w_e^{(l)}), \quad (u, v, e) \in E \\ x_v^{(l+1)} &= \varphi(x_v^{(l)}, \rho(\{m_e^{(l+1)} : (u, v, e) \in E\})) \end{aligned} \tag{1}$$

where ϕ is the message function, ρ is the reduce function, and φ is some update function. $m_e^{(l+1)}$ is the generated message, and $x_v^{(l+1)}$ is the updated hidden state of node v. In the case of graph-level tasks, a readout layer is necessary to obtain the representation of the entire graph. This is achieved by pooling node representations after the final layer L of the GNNs.

$$x_G = \text{READOUT}(x_v^{(L)} \mid v \in G\}) \tag{2}$$

The readout function should be designed as a permutation-invariant function, such as max, sum, averaging, or more complex pooling functions [29], [30].

GNNs have demonstrated remarkable accuracy in molecular property prediction. However, when applied to small labeled molecule datasets, GNNs often suffer from overfitting. One effective approach to tackle this challenge is self-supervised learning.

C. SELF-SUPERVISED LEARNING ON GRAPHS

Self-supervised learning (SSL) on graphs aims to extract knowledge from unlabeled graph data and enhance the performance of unknown downstream tasks. SSL generally utilizes internal data properties as labels instead of relying solely on external label information compared to supervised learning. More formally, graph-based SSL can be categorized into three types, depending on the specific internal properties used for learning: generative, predictive, and contrastive methods [31]. The specific self-supervised objective and pipeline design depend on the method type.

Generative methods focus on pre-training tasks designed to predict explicit features in graphs, such as masked features of nodes or edges. For example, Hu et al. [22] proposed a node attribute prediction task, while GPT-GNN [32] utilized an autoregressive framework to perform reconstruction tasks on randomly masked nodes and edges.

Predictive methods aim to predict generated labels, with many studies focusing on motif-based approaches [33], [34], [35]. Graph motifs are frequently occurring subgraph patterns, often representing functional groups in molecules. Grover [3] designed contextual property prediction and graph-level motif prediction tasks, although it overlooked the topological information among motifs. To address this limitation, MGSSL [36] introduced a generative pre-training framework that incorporates topological and motif-label prediction.

Contrastive methods involve contrasting two different views generated from graph augmentations [37]. General data augmentation techniques in graph contrastive learning (GraphCL), such as node dropping, edge permuting, and subgraph extracting, can significantly alter the chemical properties of molecules, leading to limited improvements or negative transfer on downstream tasks [38]. Various data augmentation strategies and pretext tasks have been proposed for molecular representation learning. For instance, MICRO-Graph [39] employed EM-clustering to learn motifs and then sample motif-like subgraphs for context-global contrasting. MoCL [40] introduced a novel augmentation scheme called substructure substitution, which aims to preserve the graph semantics during augmentations by incorporating local-domain knowledge. Additionally, You et al. [41] proposed a general framework for dynamically and automatically selecting augmentations in graph contrastive learning.

Our proposed DGSSL is a node-level self-supervised pre-training framework that incorporates domain knowledge into node-level prediction targets. We introduce a novel SSL task to overcome the limitations of other node-level SSL methods [3], [22].

III. DESCRIPTOR-BASED GRAPH SELF-SUPERVISED LEARNING

This section contains details of our well-designed node-level pre-training task. In the following, we first describe the extraction of domain knowledge from FCSS descriptor centers. Then we show the limitation of current existing node-level pre-training tasks and propose our newly defined prediction target infused with domain knowledge.

A. FCSS DESCRIPTOR CENTERS

Toxicity and other biological properties of molecules primarily depend on the weak bonds formed between the molecule

TABLE 1. List of 4 out of the 17 selected FCSS Descriptor Centers (Z represents any atom, R represents any atom except for H).

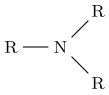
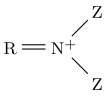
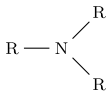
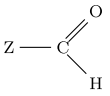
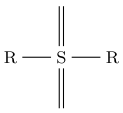
DC	Valence	DC	Valence	DC	Valence	DC	Valence
R═NH	3	Z—NH—Z	3		3	R═CH_2	4

TABLE 2. List of 17 selected FCSS Descriptor Centers (Z denotes any atom, R denotes any atom except hydrogen (H)).

DC	Valence	DC	Valence	DC	Valence	DC	Valence
	4		4	Z—NH—Z	3		3
R═NH	3	R═N—R	3	O═R	2	Z—SH	2
R—S—R	2		2	S═R	2	R═CH_2	4
R—OH	2	R—O—R	2		6	R≡CH	4
R≡N	3						

and the biological receptor during their interaction. These weak bonds are influenced by the presence of π-electrons in the molecule. That is why we extract domain knowledge from a descriptor language known as fragmentary code of substructure superposition (FCSS) [42]. In FCSS, molecules are described using substructures that can serve as centers for weak bonds. These centers referred to as active or descriptor centers (DCs) in FCSS, hold biological significance from an expert's perspective. They can be heteroatoms (N, O, S, P, metals, etc.), carbons connected by double or triple bonds, and aromatic systems. Since aromaticity can be determined by the properties of atoms or bonds (if an atom or bond is aromatic), we focus only on the first two cases and gather 17 types of DC patterns from FCSS.

In Table 1, the first three DCs are heteroatoms in different chemical environments, the last one is a carbon which is connected by double bonds. These DCs can be considered as motifs and utilized in motif-based graph-level pre-training tasks. However, they differ from typical motifs whose semantic information is based on the whole subgraphs. DCs are small ego-nets. The ego atoms are those which can be centers of weak interaction (i.e., N in the first three DCs and C in the last one in Table 1). It is meaningful to encode the domain knowledge within a DC into its corresponding ego atom as a contextual property. This characteristic enables the infusion of domain knowledge related to DCs into node-level pre-training tasks. To detect these DCs in molecules, we have developed a specialized algorithm. The complete list of selected DCs is presented in Table 2.

B. CONSTRUCTION OF SELF-SUPERVISED PRE-TRAINING TASK

Here, we present our descriptor-based node-level pre-training task. Based on the assumption that DCs are valuable domain knowledge that can contribute to predicting toxicity, Graph Neural Networks (GNNs) have the potential to learn such domain information from labeled data and improve performance. However, most existing node-level pre-training tasks may even limit the ability of GNNs. For example, Hu et al. [22] mask atom type and predict it, resulting in similar representations being assigned to atoms of the same kind in the final node embeddings. This can impact the ability of GNNs to capture different DCs in molecules during the fine-tuning stage. The context property prediction proposed by Grover [3] predicts statistical information about atoms and their local subgraphs. However, statistical information does not necessarily capture chemical semantic information. Therefore, a suitable pre-training task at the node level for molecular property prediction should satisfy the following criteria: i) The prediction target should reflect the contextual information of atoms when they exhibit specific chemical semantics in various local subgraphs. ii) For regular atoms without the aforementioned characteristics, atom types are sufficient as the prediction target. Guided by these criteria,

we design a node-level prediction task that integrates domain knowledge from DCs, where the semantics of local subgraphs are encoded into the prediction target. Specifically, we detect DCs in molecules and encode their local subgraph information as special atomic numbers. For example, if a 119-bit vector is defined to encode atomic symbols, it will be expanded to 136 bits after encoding 17 DCs as special atomic numbers. It means that atoms with the same atomic number will be further divided into different categories based on their chemical environment (if they are ego atoms in DCs). The newly defined prediction target resolves ambiguity issues regarding atom types and incorporates essential chemical semantics within local subgraphs.

With the prediction target defined, for each unlabeled compound in the pre-training dataset, we randomly mask its input node/edge features, replacing the masked features with a special indicator. After feeding the processed molecular graphs into the graph encoder in the pre-training model, we obtain embeddings of nodes/edges. Then, we apply a linear model on top of these embeddings to predict the masked node/edge attributes. Differing from previous node-level tasks, we predict the prediction target, which contains atom types or knowledge from DCs. This is a multi-label classification task, where each class corresponds to a node type or a DC.

Overall, the proposed method consists of the following steps:

1) Detect DCs in molecules.
2) Encode information about found DCs using one-hot encoding. Table 1 describes the dictionary of all possible DCs for one-hot encoding.
3) Concatenate the one-hot encoding to the vector of atomic symbols.
4) Pre-train the model using the novel DCs targets defined.

C. FINE-TUNING FOR DOWNSTREAM TASKS

A high-quality graph encoder should be obtained after pre-training DGSSL on a large number of unlabeled molecules. Then we fine-tune the pre-trained model using labeled data in downstream tasks. Since all tasks in molecular property prediction are graph-level tasks, we can incorporate an additional classification layer for the downstream tasks. The parameters of the graph encoder, learned during the pre-training stage, are used to initialize the parameters of the models trained in the fine-tuning stage. Through fine-tuning, we aim to obtain a well-performing model specifically tailored for the molecular property prediction task.

IV. EXPERIMENTS

A. EXPERIMENTAL SETUP

1) PRE-TRAINING DATASET

We pre-train DGSSL using a dataset of 2 million unlabeled compounds sampled from the ZINC15 dataset [23].

TABLE 3. Dataset information.

Dataset	#Compounds	#Tasks	Metric
tox21	7831	12	ROC-AUC
clintox	1478	2	ROC-AUC
toxcast	8575	617	ROC-AUC

TABLE 4. Test ROC-AUC (%) performance on toxicity-related benchmarks. The best result for each dataset is highlighted in bold, and the second-best result is underlined.

SSL Methods	tox21	toxcast	clintox	Avg.
No pretrain	74.2±0.7	62.5±0.9	59.8±6.2	65.5
Infomax	75.2±0.3	62.8±0.6	73.0±3.2	70.3
JOAO	75.0±0.3	62.9±0.5	81.3±2.5	73.1
Attribute masking	76.7±0.5	63.3±0.6	71.9±5.1	70.6
Grover	76.3±0.6	63.4±0.6	76.9±1.9	72.2
MGSSL	76.5±0.3	**64.1±0.7**	79.7±2.2	73.4
DGSSL (ours)	**76.9±0.4**	63.4±0.7	**89.5±2.0**	**76.6**

2) DOWNSTREAM BENCHMARK DATASETS

We selected three toxicity-related benchmarks from MoleculeNet [43] to perform the experiments:

- tox21: This dataset comprises toxicity measurements for 12 biological targets.
- clintox [44]: This dataset includes toxicity information from FDA clinical trials.
- toxcast [45]: This dataset contains toxicity data obtained through in vitro high-throughput screening.

3) DATA SPLITTING

To simulate a real-world use case, we employ the scaffold split strategy [46], which divides compounds based on their substructures (scaffolds). This splitting approach provides a more realistic distribution of compound structures among the train, validation, and test sets. The ratio for splitting the data into train, validation, and test sets is 8:1:1.

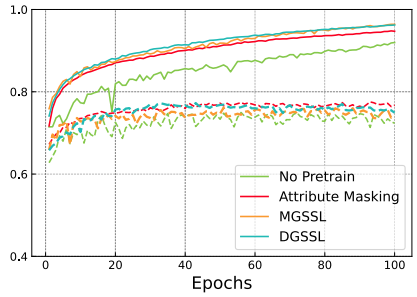
4) BASELINES

We compare DGSSL against five popular state-of-the-art self-supervised learning approaches for graphs in our evaluation.

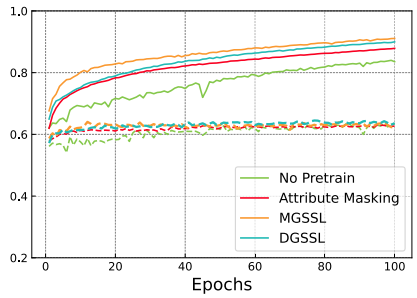
- Infomax [47] is a contrastive method that contrasts the embeddings of the entire graph and its substructures.
- JOAO [41] is an optimization framework that dynamically and automatically selects augmentations in GraphCL.
- Attribute Masking [22] masks the atom/edge types and predicts them in the pre-training task.
- Grover [3] includes a node-level contextual property prediction task and a graph-level motif prediction task.
- MGSSL [36] is a motif-based graph SSL method that utilizes a powerful motif generation pre-training task.

5) MODEL CONFIGURATION

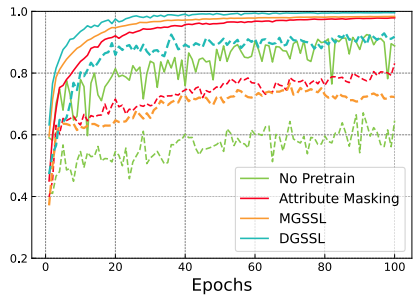
We use Graph Isomorphism Networks (GINs) [48] as our backbone in the following experiments, as they have been demonstrated to be the most expressive GNN models.



(a) tox21



(b) toxcast



(c) clintox

FIGURE 2. Training and testing curves of different pre-training methods on GIN. Solid and dashed lines indicate training and testing curves, respectively.

The update of the node hidden state in layer l can be written as:

$$x_v^{(l+1)} = \text{MLP}_\theta(x_v^l + \sum_{u \in N_v} \text{ReLU}(x_u^l + e_{u,v})) \quad (3)$$

where x_v^l is the hidden state of node v at l-th layer, $e_{u,v}$ denotes the feature of edge which is between node u and v, N_v is the the neighbor nodes of node v.

The GIN model uses atomic number and atom chirality as the initial node features and the bond type and bond direction as the initial edge features. We employ a 5-layer GIN with a hidden dimension of 300 for each GIN layer. During the fine-tuning stage, a dropout ratio of 0.5 is applied to the GIN layers. The batch size is set to 32 for all benchmark datasets and 256 for pre-training tasks.

In the pre-training stage, we use Adam as the optimizer with an initial learning rate of 0.001, and train the model for 100 epochs. For downstream tasks, we finetune the model for 100 epochs using 10 different seeds for mini-batch selection. The model is implemented using PyTorch and executed on an RTX 3090 GPU.

B. RESULTS AND ANALYSIS

1) RESULTS ON DOWNSTREAM TASKS

Table 4 presents the overall results of the downstream tasks. In general, various self-supervised pre-training methods demonstrate the ability to enhance the performance of the downstream tasks. Notably, our proposed DGSSL outperforms other baselines on the tox21 and clintox benchmarks, highlighting the effectiveness of our descriptor-based node-level pre-training task. Figure 2 illustrates the training and testing curves of different pre-training methods when transferred to downstream tasks. It reveals that all pre-training models contribute to faster convergence during the fine-tuning stage.

2) INFLUENCE OF THE BASE GNN

Table 5 demonstrates that DGSSL is independent of GNN architectures, as it successfully leverages three popular GNN models - GCN [5], GIN [48], and GraphSAGE [49] - as backbones. We report the average ROC-AUC on all three benchmarks. Notably, DGSSL exhibits larger relative gains compared to Attribute Masking [22] across all these GNN architectures.

TABLE 5. Compare pre-training gains with different GNN architectures, averaged ROC-AUC(%) on 3 toxicity-related benchmarks.

Model	GCN	GIN	GraphSAGE
No pretrain	70.0	65.5	66.1
Attribute Masking	68.3	70.6	68.9
DGSSL (ours)	**76.1**	**76.6**	**76.8**

TABLE 6. Test ROC-AUC (%) performance on toxicity-related benchmarks of GIN without pre-training. The best result for each dataset is highlighted in bold.

Model	tox21	toxcast	Avg.
No pretrain	74.2±0.7	62.5±0.9	68.4
No pretrain (DCs)	**75.5±1.0**	**62.6±1.1**	**69.1**

3) STUDIES ON GRAPH-LEVEL PRE-TRAINING TASK

To investigate the potential of incorporating DCs into graph-level pre-training tasks, we conducted additional experiments using DCs and ordinary motifs as prediction targets. Motifs in molecules were detected using the professional open-source package RDKit [50], which is the same method employed in Grover [3]. DCs are extracted using the algorithm mentioned earlier. A total of 87 motif labels and 17 DCs were defined. We separately use motifs and DCs as prediction targets in the graph-level task, where each motif or DC correspond to a single label.

In Table 7, we observe that using motif labels as pre-training targets resulted in minimal improvement on the toxicity benchmarks. Similarly, using DC labels in pre-training

tasks yielded limited improvement. This finding suggests that utilizing DCs in label prediction pre-training tasks leads to greater improvement compared to motifs, despite the smaller number of DCs compared to motifs. However, neither approach outperforms DGSSL.

Hu et al. [22] has reported negative transfer in some downstream tasks even after performing supervised graph-level prediction. This highlights the challenges involved in designing effective graph-level pre-training tasks. Firstly, motif-label or DC label prediction is a quite simple graph-level pre-training task. Secondly, these tasks solely focus on the presence of specific motifs in a molecule, disregarding important factors such as the quantity of each motif and the overall topology and structure of the molecule. As a result, the relationship between these prediction targets and downstream tasks is weak, leading to limited improvement. In contrast, MGSSL [36] incorporates topological information through a well-designed motif-tree generation task, significantly enhancing the performance of downstream tasks.

TABLE 7. Test ROC-AUC (%) performance on toxicity-related benchmarks of graph-level pre-training strategies with GIN. The best result for each dataset is highlighted in bold.

Model	tox21	toxcast	Avg.
No pretrain	74.2±0.7	62.5±0.9	68.4
Motifs	74.4±0.9	62.6±0.4	68.5
DCs	**75.1±0.5**	**63.4±0.4**	**69.3**

V. CONCLUSION AND FUTURE WORKS

This paper proposes a novel pre-training procedure for molecular graphs called Descriptor-based Graph Self-Supervised Learning (DGSSL). It is a powerful method that incorporates the descriptor centers into the node-level pre-training task solving the main challenges of existing pre-training frameworks: lack of domain information in the node-level auxiliary tasks and high computational complexity simultaneous training for motif-based and node-level methods. DGSSL identifies and matches DCs in molecules, enabling the encoding of local subgraph information into the features of center atoms as contextual properties. By combining atom type and atom contextual property, we propose a well-designed node-level pre-training target that facilitates the transfer of domain knowledge to downstream tasks. We demonstrate that DGSSL achieves state-of-the-art performance on toxicity-related benchmark datasets. Furthermore, we analyze how pre-training impacts downstream tasks, assessing its alignment with our expectations.

ACKNOWLEDGMENT

Author Contributions: Xinze Li: initial idea, model and experiment design, and paper preparation; Ilya Makarov: paper revision, help with experiment and model design, and research supervision; and Dmitrii Kiselev: paper revision, help with experiment and model design.

REFERENCES

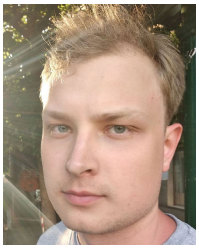
[1] E. J. Bjerrum, "SMILES enumeration as data augmentation for neural network modeling of molecules," 2017, *arXiv:1703.07076*.
[2] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1263–1272.
[3] Y. Rong, Y. Bian, T. Xu, W. Xie, Y. Wei, W. Huang, and J. Huang, "Self-supervised graph transformer on large-scale molecular data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 12559–12571.
[4] J. Dong, N.-N. Wang, Z.-J. Yao, L. Zhang, Y. Cheng, D. Ouyang, A.-P. Lu, and D.-S. Cao, "ADMETlab: A platform for systematic ADMET evaluation based on a comprehensively collected ADMET database," *J. Cheminformatics*, vol. 10, no. 1, pp. 1–11, Dec. 2018.
[5] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
[6] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.
[7] I. Makarov, D. Kiselev, N. Nikitinsky, and L. Subelj, "Survey on graph embeddings and their applications to machine learning problems on graphs," *PeerJ Comput. Sci.*, vol. 7, pp. 1–62, Feb. 2021.
[8] I. Wallach, M. Dzamba, and A. Heifets, "AtomNet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery," 2015, *arXiv:1510.02855*.
[9] Z. Xiong, D. Wang, X. Liu, F. Zhong, X. Wan, X. Li, Z. Li, X. Luo, K. Chen, H. Jiang, and M. Zheng, "Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism," *J. Med. Chem.*, vol. 63, no. 16, pp. 8749–8760, Aug. 2020.
[10] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley, "Molecular graph convolutions: Moving beyond fingerprints," *J. Comput.-Aided Mol. Des.*, vol. 30, no. 8, pp. 595–608, Aug. 2016.
[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[12] A. M. Grachev, D. I. Ignatov, and A. V. Savchenko, "Neural networks compression for language modeling," in *Proc. 7th Int. Conf. Pattern Recognit. Mach. Intell. (PReMI)*, Kolkata, India. Cham, Switzerland: Springer, Dec. 2017, pp. 351–357.
[13] A. V. Savchenko, "Phonetic words decoding software in the problem of Russian speech recognition," *Autom. Remote Control*, vol. 74, no. 7, pp. 1225–1232, Jul. 2013.
[14] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.
[15] I. Makarov, M. Bakhanova, S. Nikolenko, and O. Gerasimova, "Self-supervised recurrent depth estimation with attention mechanisms," *PeerJ Comput. Sci.*, vol. 8, pp. 1–25, Jan. 2022.
[16] I. Makarov and G. Borisenko, "Depth inpainting via vision transformer," in *Proc. 19th IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*. New York, NY, USA: IEEE, Oct. 2021, pp. 286–291.
[17] A. V. Savchenko and Y. I. Khokhlova, "About neural-network algorithms application in viseme classification problem with face video in audiovisual speech recognition systems," *Opt. Memory Neural Netw.*, vol. 23, no. 1, pp. 34–42, Jan. 2014.
[18] B. Tseytlin and I. Makarov, "Hotel recognition via latent image embeddings," in *Proc. 16th Int. Work-Conf. Artif. Neural Netw. (IWANN)*, Universitat Politecnica de Catalunya. Berlin, Germany: Springer, Jun. 2021, pp. 293–305.
[19] M. Golyadkin and I. Makarov, "Semi-automatic Manga colorization using conditional adversarial networks," in *Proc. 9th Int. Conf. Anal. Images, Social Netw. Texts (AIST)*, in Lecture Notes in Computer Science, Skoltech. Berlin, Germany: Springer, Oct. 2020, pp. 230–242.
[20] A. V. Savchenko, "EmotiEffNets for facial processing in video-based valence-arousal prediction, expression classification and action unit detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2023, pp. 5715–5723.
[21] A. V. Savchenko, "MT-EmotiEffNet for multi-task human affective behavior analysis and learning from synthetic data," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 45–59.
[22] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec, "Strategies for pre-training graph neural networks," 2019, *arXiv:1905.12265*.
[23] T. Sterling and J. J. Irwin, "ZINC 15—Ligand discovery for everyone," *J. Chem. Inf. Model.*, vol. 55, no. 11, pp. 2324–2337, Nov. 2015.

[24] D. Weininger, "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules," *J. Chem. Inf. Comput. Sci.*, vol. 28, no. 1, pp. 31–36, Feb. 1988.
[25] P. Karpov, G. Godin, and I. V. Tetko, "Transformer-CNN: Swiss knife for QSAR modeling and interpretation," *J. Cheminformatics*, vol. 12, no. 1, pp. 1–12, Dec. 2020.
[26] D. Rogers and M. Hahn, "Extended-connectivity fingerprints," *J. Chem. Inf. Model.*, vol. 50, no. 5, pp. 742–754, May 2010.
[27] M. Zaslavskiy, S. Jégou, E. W. Tramel, and G. Wainrib, "ToxicBlend: Virtual screening of toxic compounds with ensemble predictors," *Comput. Toxicol.*, vol. 10, pp. 81–88, May 2019.
[28] Y. Song, S. Zheng, Z. Niu, Z.-H. Fu, Y. Lu, and Y. Yang, "Communicative representation learning on attributed molecular graphs," in *Proc. IJCAI*, Jul. 2020, pp. 2831–2838.
[29] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.
[30] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.
[31] L. Wu, H. Lin, C. Tan, Z. Gao, and S. Z. Li, "Self-supervised learning on graphs: Contrastive, generative, or predictive," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 4216–4235, Apr. 2023.
[32] Z. Hu, Y. Dong, K. Wang, K.-W. Chang, and Y. Sun, "GPT-GNN: Generative pre-training of graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2020, pp. 1857–1867.
[33] I. Makarov, A. Savchenko, A. Korovko, L. Sherstyuk, N. Severin, D. Kiselev, A. Mikheev, and D. Babaev, "Temporal network embedding framework with causal anonymous walks representations," *PeerJ Comput. Sci.*, vol. 8, pp. 1–27, Jan. 2022.
[34] I. Makarov and O. Gerasimova, "Predicting collaborations in co-authorship network," in *Proc. 14th IEEE Int. Workshop Semantic Social Media Adaptation Personalization (SMAP)*, Cyprus University of Technology. New York, NY, USA: IEEE, Jun. 2019, pp. 1–6.
[35] I. Makarov and O. Gerasimova, "Link prediction regression for weighted co-authorship networks," in *Proc. 15th Int. Work-Conf. Artif. Neural Netw. (IWANN)*, Universitat Politecnica de Catalunya. Berlin, Germany: Springer, Jul. 2019, pp. 667–677.
[36] Z. Zhang, Q. Liu, H. Wang, C. Lu, and C.-K. Lee, "Motif-based graph self-supervised learning for molecular property prediction," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 15870–15882.
[37] I. Makarov, K. Korovina, and D. Kiselev, "JONNEE: Joint network nodes and edges embedding," *IEEE Access*, vol. 9, pp. 144646–144659, 2021.
[38] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Graph contrastive learning with adaptive augmentation," in *Proc. Web Conf.*, Apr. 2021, pp. 2069–2080.
[39] A. Subramonian, "Motif-driven contrastive learning of graph representations," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 18, pp. 15980–15981.
[40] M. Sun, J. Xing, H. Wang, B. Chen, and J. Zhou, "MoCL: Data-driven molecular fingerprint via knowledge-aware contrastive learning from molecular graph," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 3585–3594.
[41] Y. You, T. Chen, Y. Shen, and Z. Wang, "Graph contrastive learning automated," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 12121–12132.
[42] V. V. Avidon, I. A. Pomerantsev, V. E. Golender, and A. B. Rozenblit, "Structure-activity relationship oriented languages for chemical structure representation," *J. Chem. Inf. Comput. Sci.*, vol. 22, no. 4, pp. 207–214, Nov. 1982.
[43] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande, "MoleculeNet: A benchmark for molecular machine learning," *Chem. Sci.*, vol. 9, no. 2, pp. 513–530, 2018.
[44] P. A. Novick, O. F. Ortiz, J. Poelman, A. Y. Abdulhay, and V. S. Pande, "SWEETLEAD: An in silico database of approved drugs, regulated chemicals, and herbal isolates for computer-aided drug discovery," *PLoS ONE*, vol. 8, no. 11, Nov. 2013, Art. no. e79568.
[45] A. M. Richard, R. S. Judson, K. A. Houck, C. M. Grulke, P. Volarath, I. Thillainadarajah, C. Yang, J. Rathman, M. T. Martin, J. F. Wambaugh, T. B. Knudsen, J. Kancherla, K. Mansouri, G. Patlewicz, A. J. Williams, S. B. Little, K. M. Crofton, and R. S. Thomas, "ToxCast chemical landscape: Paving the road to 21st century toxicology," *Chem. Res. Toxicol.*, vol. 29, no. 8, pp. 1225–1251, Aug. 2016.
[46] B. Ramsundar, P. Eastman, P. Walters, and V. Pande, *Deep Learning for the Life Sciences: Applying Deep Learning to Genomics, Microscopy, Drug Discovery, and More*. Sebastopol, CA, USA: O'Reilly Media, 2019.
[47] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," in *Proc. ICLR*, 2019, vol. 2, no. 3, p. 4.
[48] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" 2018, *arXiv:1810.00826*.
[49] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
[50] G. Landrum et al., "RDKit contributors," Tech. Rep. 2022.03.1 (Q1 2022) Release, 2013, p. 4, vol. 1, nos. 1–79. [Online]. Available: https://www.rdkit.org

XINZE LI received the master's degree in applied mathematics and informatics from HSE University.

ILYA MAKAROV received the Specialist degree in mathematics from Lomonosov Moscow State University, Moscow, Russia, and the Ph.D. degree in computer science from the University of Ljubljana, Ljubljana, Slovenia.

Since 2011, he has been a Lecturer with the School of Data Analysis and Artificial Intelligence, HSE University, where he was the School Deputy Head, from 2012 to 2016, and he is currently an Associate Professor and a Senior Research Fellow. He was also the Program Director of the Bigdata Academy MADE, VK, and a Researcher with the Samsung-PDMI Joint AI Center, St. Petersburg Department, V.A. Steklov Mathematical Institute, Russian Academy of Sciences, Saint Petersburg, Russia. He is also a Senior Research Fellow of the Artificial Intelligence Research Institute (AIRI), Moscow, where he leads the research in industrial AI. He became the Head of the AI Center and Data Science Tech Master Program in NLP, National University of Science and Technology MISIS.



DMITRII KISELEV received the master's degree in applied mathematics and informatics and the Ph.D. degree in computer science from HSE University. He is currently a Researcher in the field of graph neural networks applications to the industrial AI with the Artificial Intelligence Research Institute, Moscow, Russia.