MORE: Molecule Pretraining with Multi-Level Pretext Task

Yeongyeong Son[1], **Dasom Noh**[1], **Gyoungyoung Heo**[1], **Gyoung Jin Park**[1], **Sunyoung Kwon**[1, 2, 3*]

[1]Department of Information Convergence Engineering, Pusan National University, Korea
[2]School of Biomedical Convergence Engineering, Pusan National University, Korea
[3]Center for Artificial Intelligence Research, Pusan National University, Korea
{vlddus123, ds.noh, qorskds, rudwls2717, sy.kwon}@pusan.ac.kr

Abstract

Foundation models, serving as pretrained fundamental bases for a variety of downstream tasks, try to learn versatile, rich, and generalizable representations that can be quickly adopted through fine-tuning or even in a zero-shot manner for specific applications. Foundation models for molecular representation are no exception. Various pretext tasks have been proposed for pretraining molecular representations, but these approaches have focused on only single or partial properties. Molecules are complicated and require different perspectives depending on purposes: insights from local- or global-level, 2D-topology or 3D-spatial arrangement, and low- or high-level semantics. We propose **M**ulti-level m**O**lecule g**R**aph pr**E**-train (**MORE**) to consider these multiple aspects of molecules simultaneously. Experimental results demonstrate that our proposed method effectively learns comprehensive representations by showing outstanding performance in both linear probing and full fine-tuning. Notably, in quantification experiments of forgetting the pretrained models, MORE consistently exhibits minimal and stable parameter changes with the smallest performance gap, whereas other methods show substantial and inconsistent fluctuations with larger gaps. The effectiveness of individual pretext tasks varies depending on the problems being solved, which again highlights the need for a multi-level perspective. Scalability experiments reveal steady improvements of MORE as the dataset size increases, suggesting potential gains with larger datasets as well.

Code — https://github.com/IT-fatica/MORE

Introduction

Foundation models are pretrained on massive amounts of diverse datasets, enabling effective adaptation to various downstream tasks. Supervised pretraining based on human labeling requires extensive labeled data, leading to drawbacks such as high cost, time consumption, label inconsistency, and limited scalability. Self-supervised learning (SSL)-based pretraining does not require human manual labeling, so can be free from many of the drawbacks caused by human labeling-based pretraining approaches. Recent foundation models like GPT (Radford et al. 2018), BERT (Devlin et al. 2018), and ViT (Dosovitskiy et al. 2020) leverage SSL techniques for efficient pretraining.

This trend in the molecular field is no exception. Obtaining labeled data for molecular tasks is challenging due to the reliance on costly and inconsistent wet lab experiments, whereas unlabeled molecular data is relatively abundant. Inspired by the success of SSL in NLP and CV, researchers have explored SSL for molecular tasks (Moon, Im, and Kwon 2023; Hu et al. 2019; Rong et al. 2020). SSL learns the representations from the data itself, guided by the predefined pretext task. Thus, the design of the pretext task is crucial for the performance of SSL (Fang et al. 2024).

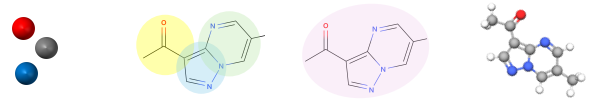
The current pretext tasks for molecular graphs often focus on partial molecular properties. Masked node/edge reconstruction (Hu et al. 2019; Hou et al. 2022; Tan et al. 2023) captures local features. Similarly, motif-based methods (Ji et al. 2022; Zhang et al. 2021) emphasize local structural information. Contrastive methods (You et al. 2020; Liu et al. 2021) exploit global graph agreements but often rely on 2D topological structures, overlooking 3D geometry. Low-level semantics such as masking and augmentation have been prevalently used, but high-level semantics such as molecular weight and polar surface area have been overlooked.

To serve as a fundamental base for the molecular domain, pretext tasks must exploit different perspectives simultaneously, and the learned representation must be versatile, rich, and highly generalizable. We propose a novel pretraining method, **M**ulti-level m**O**lecule g**R**aph pr**E**-train (**MORE**), which integrates four graph viewpoints: node-, subgraph-, graph-, and 3D-level, as shown in Figure 1. Our subgraph- and graph-level pretext task is different from conventional approaches, learning predefined local or global information.

To evaluate MORE, we maintain the same neural network architecture and hyperparameters across methods, varying only the pretext tasks during pretraining. We use scaffold split for downstream datasets to assess robustness and generalizability. We compare linear probing and full fine-tuning performance. Despite pretraining on large data, full fine-tuning can lead to a forgetting problem, where previously learned knowledge is lost (Zhou and Cao 2021). We analyze forgetting in pretrained models, examine individual pretext tasks, and investigate the scalability of dataset size in pretraining. Our contributions are as follows:

- We design a multi-level pretext task to learn compre-

*Corresponding author.



(a) Node (b) Subgraph (c) Graph (d) 3D

Figure 1: Illustrative examples of four-level viewpoint in a molecular graph.

hensive representation from various views of molecules: node-, subgraph-, graph-, and 3D geometric-levels. Experimental results show that MORE effectively learns generalizable and transferable representations, outperforming both in linear probing and full fine-tuning.

- We provide quantitative analysis of forgetting of pretrained models, showing that MORE exhibits minimal and stable parameter changes with the smallest performance gap, whereas other methods show substantial and inconsistent fluctuations with larger gaps.
- Analysis of individual pretext tasks reveals that the significance of pretext tasks may vary depending on downstream tasks. However, our graph-level task, leveraging molecular descriptors not extensively studied as predictive targets, shows the best performance on average.
- Scalability experiments demonstrate that increasing pretraining dataset size consistently improves the performance of MORE, highlighting its potential as a foundation model.

Preliminary and Related Work

Graph Neural Networks

Graph Neural Networks (GNNs) are powerful tools for graph-structured data. A molecular graph is represented as $\mathcal{G} = (\mathcal{V}, \mathbf{X}, \mathbf{A})$, where $\mathcal{V}$ is the set of nodes, $N = |\mathcal{V}|$ is the number of nodes, $\mathbf{A} \in \{0,1\}^{N \times N}$ is the adjacency matrix, and $\mathbf{X} \in \mathbb{R}^{N \times d}$ is the node feature matrix, with d as the number of features. GNNs aggregate information from node v's k-hop neighborhood over k iterations, and the graph representation $h_{\mathcal{G}}$ is obtained using the READOUT function, which aggregates node embeddings via operations like mean or sum pooling. GNNs are formalized as follows:

$$a_v^{(k)} = \text{AGGREGATE}\left(\left\{h_u^{(k-1)} | u \in \mathcal{N}(v)\right\}\right) \quad (1)$$

$$h_v^{(k)} = \text{UPDATE}\left(h_v^{(k-1)}, a_v^{(k)}\right) \quad (2)$$

$$h_{\mathcal{G}} = \text{READOUT}\left(h_v^{(k)} \mid v \in \mathcal{G}\right) \quad (3)$$

where $\mathcal{N}(v)$ is the set of neighbors of node v, and $h_v^{(k)}$ is the representation of node v at the k-th layer.

Pretrain on Molecular Graphs

Molecular graph pretraining is typically divided into contrastive and generative learning. Contrastive learning, such as GraphCL (You et al. 2020) and GraphMVP (Liu et al. 2021), aims to bring similar samples closer and push dissimilar samples apart in the embedding space. While it captures overall graph structure, but has limited learning of high-level semantic information. Generative learning, like EdgePred (Hamilton, Ying, and Leskovec 2017), AttrMasking (Hu et al. 2019), and GraphMAE (Hou et al. 2022), focuses on restoring the original input or generating new graphs, often emphasizing low-level features. Recently, new approaches have been proposed to utilize molecular features in diverse ways, such as KANO (Fang et al. 2023), which leverages functional group information by introducing a knowledge graph and prompts. Although various pretraining models for molecular graphs have been proposed, most focus on single or partial aspects, and comprehensive representation learning has been neglected.

Method

Overall Architecture

As shown in Figure 2, MORE is an encoder-decoder architecture. The encoder f_{E} takes a molecular graph $\mathcal{G}' = \left(\mathcal{V}, \widetilde{\mathbf{X}}, \mathbf{A}\right)$ as input, where $\widetilde{\mathbf{X}}$ is the node feature matrix with some masked entries. The four decoders f_{D}, each performing different pretext tasks, contribute to the encoder's comprehensive representation learning during pretraining.

$$\mathbf{H} = f_{\text{E}}\left(\widetilde{\mathbf{X}}, \mathbf{A}\right), \quad \mathbf{O} = f_{\text{D}}(\mathbf{H}) \quad (4)$$

where $\mathbf{H} \in \mathbb{R}^{N \times d_{\text{E}}}$ denotes the node representation matrix, and d_{E} is the embedding dimension. $\mathbf{O}$ denotes the decoder output, used to compute the loss for each pretext task.

After petraining, the encoder f_{E} is transferred to the downstream task. For fine-tuning, a downstream task is performed using the pretrained encoder f_{E} and downstream decoder $f_{\text{Downstream}}$.

$$\widehat{\mathbf{Y}} = f_{\text{Downstream}}\left(f_{\text{E}}\left(\mathbf{X}, \mathbf{A}\right)\right) \quad (5)$$

where $\widehat{\mathbf{Y}}$ denotes the prediction for the downstream task, used to calculate the downstream loss.

Pretext Tasks

Node-level Pretext Task We adopt the pretext task to learn information focused on a node, which is the most basic unit of a graph — the masked node reconstruction. We follow the method proposed by GraphMAE (Hou et al. 2022). GraphMAE, based on the Graph Autoencoder (GAE) (Kipf and Welling 2016), uses a re-masking strategy for a more expressive decoder and replaces Mean Squared Error (MSE) with the Scaled Cosine Error (SCE) loss to address reconstruction limitations.

As shown in Equation 4, we generate the masked feature $\widetilde{\mathbf{X}}$ by randomly masking n node features based on the masking ratio. Given the node-level decoder $f_{\text{D}_{\text{node}}}$, the output is as follows:

$$\mathbf{O}_{\text{node}} = f_{\text{D}_{\text{node}}}\left(\widetilde{\mathbf{H}}, \mathbf{A}\right) \quad (6)$$

where $\widetilde{\mathbf{H}}$ is the re-masked node representation, and $\mathbf{O}_{\text{node}} \in \mathbb{R}^{N \times 119}$ denotes the predicted atomic numbers for all nodes,

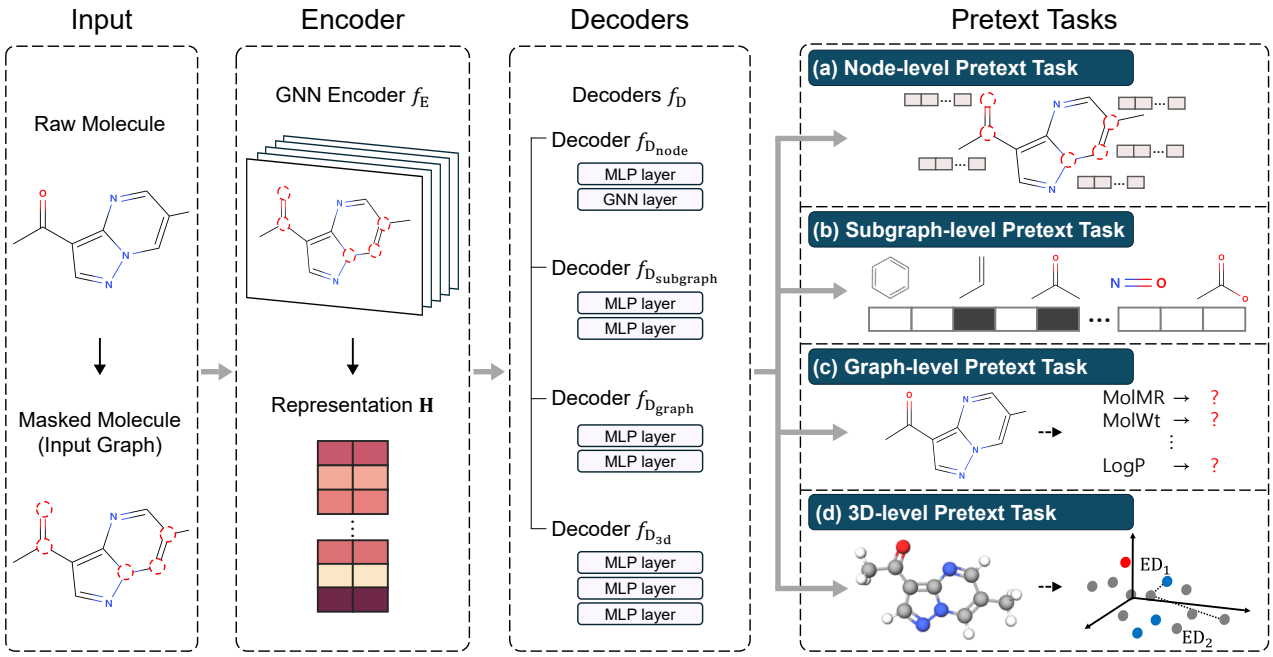


Figure 2: An illustration of our MORE. It consists of an encoder, which takes a molecular graph with some nodes masked as input and learns a meaningful representation, and the decoders, which learn multiple attributes of the molecule. In the decoders, (a) reconstructs node-level molecular structures, (b) predicts subgraph-level molecular structures, (c) predicts graph-level molecular attributes, and (d) learns 3D-level molecular structures.

with 119 indicating the total atom types. We calculate the loss only for the masked nodes.

Subgraph-level Pretext Task We design a pretext task to capture molecular structural characteristics — predicting MACCS (Molecular ACCess System) keys (Durant et al. 2002), a type of molecular fingerprint representing unique chemical patterns. MACCS keys encode the molecular structure as binary bits, where each bit represents the presence (1) or absence (0) of specific substructures. For example, if a molecule contains an aromatic ring, the value of the corresponding bit is 1. Using the RDKit library (Landrum et al. 2020), we employed 155 of the 166 sub-structure keys, excluding those with zero values across all molecules in the pretraining dataset.

The graph representation is derived from $\mathbf{H}$ via the READOUT function and used to predict 155 MACCS keys. Binary Cross Entropy loss is applied for learning. Given the subgraph-level decoder $f_{\mathrm{D}_{\mathrm{subgraph}}}$, the output $\mathbf{O}_{\mathrm{subgraph}} \in \mathbb{R}^{1\times 155}$ is as follows:

$$\mathbf{O}_{\mathrm{subgraph}} = f_{\mathrm{D}_{\mathrm{subgraph}}}\left(\mathrm{READOUT}\left(\mathbf{H}\right)\right) \tag{7}$$

Graph-level Pretext Task We design the pretext task to learn a molecule's high-level semantic and global information — the molecular descriptors (Xue and Bajorath 2000) prediction. Molecular descriptors numerically represent the physical, chemical, structural, and geometric properties of molecules. They are crucial in analyzing and predicting the properties of molecules in chemical and biological research (Barnard et al. 2020). 200+ molecular descriptors can be easily extracted via RDKit library (Landrum et al. 2020). An example is molecular weight and LogP, representing lipophilicity. In this work, we use 194 of them, excluding those with large value ranges and those with zero values across all molecules in the pretraining dataset.

We normalized them via standard scalar due to varying distributions of values across each molecular descriptors. The graph representation obtained through the READOUT function is used to predict the 194 molecular descriptors. We use the MSE loss function. Given the graph-level decoder $f_{\mathrm{D}_{\mathrm{graph}}}$, the output $\mathbf{O}_{\mathrm{graph}} \in \mathbb{R}^{1\times 194}$ is as follows:

$$\mathbf{O}_{\mathrm{graph}} = f_{\mathrm{D}_{\mathrm{graph}}}\left(\mathrm{READOUT}\left(\mathbf{H}\right)\right) \tag{8}$$

3D-level Pretext Task We design the pretext task to learn the geometry structure in 3D spaces — the node-wise relative distances in 3D spaces prediction. Molecular properties are largely determined by 3D structures (Crum-Brown and Fraser 1865; Hansch and Fujita 1964). Conformers represent 3D structures of molecules based on rota-table single bonds, with potential energy varying by rotation degree. The lower the energy, the more likely it is to exist in nature. Even with just five conformers, it is possible to represent nearly all molecules found in nature (Liu et al. 2021). Conformers can be generated via the RDKit library (Landrum et al. 2020), and the 3D coordinates of each atom (node) can also be obtained.

In this work, we generate five conformers using the random coordinate generation method and then optimize them to minimize energy using the MMFF function (Halgren 1996). Out of the five conformers, the three with the lowest

energies are utilized. At each iteration, one conformer is randomly selected from the three conformers as the target 3D coordinates of each node. This approach enables augmentation effects. We calculate the relative distances between every node by computing the Euclidean distances from the 3D coordinates of each node to generate the true distance matrix $\mathbf{ED}_{\text{true}} \in \mathbb{R}^{N \times N}$. Given the 3D-level decoder $f_{\text{D}_{\text{3d}}}$, the output is as follows:

$$\mathbf{O}_{\text{3d}} = f_{\text{D}_{\text{3d}}}(\mathbf{H}) \tag{9}$$

where $\mathbf{O}_{\text{3d}} \in \mathbb{R}^{N \times d_{\text{3d}}}$ is the node embeddings, meaning the coordinates of each node in d_{3d} dimensions. Compute the Euclidean distance from $\mathbf{O}_{\text{3d}}$ to generate the predicted distance $\mathbf{ED}_{\text{pred}} \in \mathbb{R}^{N \times N}$. Rather than directly predicting the node-wise distance in the model, we estimate the distance based on each state of the nodes in the embedding space. Note that $\mathbf{ED}_{\text{true}}$ and $\mathbf{ED}_{\text{pred}}$ are diagonal matrices. We optimize with only the lower triangular non-diagonal elements and use the MSE loss function.

Optimizing Multi-level Pretext Task Loss

MORE is updated based on following loss $\mathcal{L}$:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{node}} + \lambda_2 \mathcal{L}_{\text{subgraph}} + \lambda_3 \mathcal{L}_{\text{graph}} + \lambda_4 \mathcal{L}_{\text{3d}} \tag{10}$$

where λ_1, λ_2, λ_3, and λ_4 denote the hyperparameter for each loss function. $\mathcal{L}_{\text{node}}$, $\mathcal{L}_{\text{subgraph}}$, and $\mathcal{L}_{\text{graph}}$ denote the losses for node-, subgraph-, and graph-level pretext task computed based on the decoder output $\mathbf{O}$, respectively. $\mathcal{L}_{\text{3d}}$ denotes the loss for the 3D-level pretext task computed based on $\mathbf{ED}$. Since there is a difference in the distribution of each loss value, we adjust the hyperparameters so that the model learns all tasks evenly, without being focused on any particular task.

Experiments

Datasets

Pretraining Dataset. We use 2 million unlabeled molecules from ZINC15 (Sterling and Irwin 2015), excluding some for which conformers are not generated, resulting in $1,974,507$ molecules. The data are randomly split into training and validation sets in a 9:1 ratio. The model is trained on the training set, and the encoder with the lowest validation loss is saved.

Downstream Dataset. We use seven graph classification datasets from MoleculeNet (Wu et al. 2018), detailed in Table 1. BBBP predicts the probability of a molecule crossing the blood-brain barrier (BBB), which depends on physicochemical properties and overall size, making molecular descriptors advantageous (Martins et al. 2012). Toxicity prediction datasets (SIDER, ToxCast, Tox21, and ClinTox) require structural features and chemical indicators (*e.g.*, electron affinity, and polarity), often utilizing fingerprints (Cavasotto and Scardino 2022). HIV and BACE assess binding affinity, where 3D molecular structures, including atomic coordinates and bond angles, are crucial for determining compatibility with target binding sites (Li et al. 2021).

Dataset	# Tasks	# Compounds	# Atoms	# Bonds
BBBP	1	2,039	24.1	26.0
Tox21	12	7,831	18.6	19.3
ToxCast	617	8,575	18.8	19.3
SIDER	27	1,427	34.3	36.1
ClinTox	2	1,478	26.3	28.1
HIV	1	41,127	25.5	27.5
BACE	1	1,513	34.1	36.9

Table 1: Details of the dataset used in the experiments. # Tasks and # Compounds are the number of tasks to perform and molecules, respectively. # Atoms and # Bonds are the averages of the number of nodes and edges in all molecules, respectively.

Dataset Split. We adopt scaffold splitting (Wu et al. 2018), which separates molecules by structural differences, a more challenging method than random splitting. Since the molecular structures in the test set are likely to be unseen during training, we can evaluate model generalization on out-of-distribution samples. The downstream datasets are split into train/validation/test sets in an $8:1:1$ ratio based on scaffolds (molecular substructures).

Settings

Implementation Details. The encoder uses a 5-layer Graph Isomorphism Network (GIN) (Xu et al. 2018) with 300 hidden units. We set the masking ratio to 25%. The node-level decoder comprises a 1-layer Multi-Layer Perceptron (MLP) and a 1-layer GIN. The subgraph- and graph-level decoders are 2-layer MLP with 256 hidden units and output shapes. The 3D-level decoder is a 3-layer MLP with hidden units of 256, 128, and 30. The downstream task decoder is a 1-layer MLP.

For pretraining, hyperparameters are tuned using validation sets. Specifically, for the loss function, we set $\lambda_1 = 4.5$, $\lambda_2 = 5.0$, $\lambda_3 = 1.0$, and $\lambda_4 = 0.04$. For fine-tuning, we follow a commonly used default setting without any hyperparameter tuning.

Baselines. We compare our experiments to eight prior methods. Note that the encoder structure and hyperparameters during fine-tuning of the eight baseline models and MORE are the same; therefore, we can focus on the effects of the pretext task.

- **Infomax** (Veličković et al. 2018) maximizes the mutual information between the local and pooled global graph representations.
- **EdgePred** (Hamilton, Ying, and Leskovec 2017) predict the adjacency matrix of a graph.
- **AttrMasking** (Hu et al. 2019) predicts masked nodes, applying MLP decoder when masking predictions.
- **ContextPred** (Hu et al. 2019) predicts context graph structure using subgraphs.
- **GraphCL** (You et al. 2020) is a contrastive learning method using a combination of four graph augmenta-

	Local		Global	
	Node	Subgraph	Graph	3D
Infomax	✓	-	-	-
EdgePred	✓	-	-	-
AttrMasking	✓	-	-	-
ContextPred	-	✓	-	-
GraphCL	-	-	✓	-
GraphLoG	-	✓	✓	-
GraphMAE	✓	-	-	-
GraphMVP	-	-	✓	✓
MORE	✓	✓	✓✓	✓

Table 2: Comparison of viewpoints in various pretraining models. ✓✓ indicates consideration of broader high-level semantic properties beyond just topological structures.

tions: node deletion, edge perturbation, subgraph cropping, and feature masking.

- **GraphLoG** (Xu et al. 2021) utilizes clustering to build a hierarchical prototype of a graph sample and contrast each local instance with its parent prototype for contrastive learning.
- **GraphMAE** (Hou et al. 2022) reconstructs masked nodes using a re-mask strategy and a GNN decoder for prediction.
- **GraphMVP** (Liu et al. 2021) maximises mutual information between 2D and 3D views of a molecule.

Table 2 shows the viewpoints of each pretraining method. MORE considers all four levels simultaneously, whereas the baselines address only some of the levels and predominantly focus on learning local information. For the graph-level, the baseline uses a self-supervision approach that considers only global structural information through contrastive learning and clustering. In contrast, our method not only incorporates global structural information but also learns high-level semantic properties by exploiting molecular descriptors.

Results

We evaluate downstream task performance under two settings: **1) Linear probing**: the encoder parameters are frozen and only the decoder is updated. This setting is commonly used to evaluate pretrained models and to assess the quality of the learned representations. **2) Full fine-tuning**: with the pretrained encoder weights set as the initial values, and then all parameters are updated to fit better new data, which may disrupt the previously learned knowledge. Note that the structures of MORE and all the other models are identical. We report the mean and standard deviation (std) of the ROC-AUC scores from three experiments conducted with different random seeds.

Prediction Performance under Linear Probing and Full Fine-tuning

Table 3 shows the prediction performance of MORE and the other eight models under both linear probing and full fine-tuning. In linear probing (Table 3 (a)), MORE achieves superior performance on all seven datasets. Moreover, the average ROC-AUC of MORE, 68.24, is markedly higher than the second-best of Infomax, 63.33. In this setting, the parameters of the pretrained encoder are frozen, and only the downstream task decoder is trained. Achieving good performance across diverse datasets can indicate that MORE has learned comprehensive representations for a variety of tasks during the pretraining process, making it easier for the model to generalize. In full fine-tuning (Table 3 (b)), most methods outperform the baseline without pretraining, clearly demonstrating the effectiveness of pretraining. MORE still exhibits outstanding performance across most datasets and shows the highest average ROC-AUC.

Quantification of Forgetting in Pretrained Models

In order to quantitatively evaluate the knowledge forgetting of pretrained model after being fine-tuned, we empirically measure the prediction performance gap and degree of parameter changes.

Figure 3 illustrates the average performance gap between linear probing and full fine-tuning. We observe that MORE not only has the highest linear probing performance but also the smallest performance gap. The small performance gap indicates that pretraining was already effective for applying to various downstream tasks and that the model adapted to new tasks without significant forgetting of the learned knowledge.

Figure 4 illustrates encoder parameter changes after full fine-tuning, normalized per dataset. The color and size of the circles represent the mean and variance, respectively. The lighter and smaller the circle, the lower the mean and variance. MORE consistently shows minimal and stable parameter changes, whereas other methods exhibit substantial and inconsistent fluctuations. This stability change suggests that it can be optimized easily for a variety of downstream tasks. We assert that learning multiple molecular attributes is excellent for generalizable knowledge.

Effectiveness of Individual Pretext Tasks

To assess the effectiveness of each of the four pretext tasks, we conduct two types of experiments: (leave-one-out analysis) one where all tasks are used except for one, and (single-task analysis) another where only a single pretext task is employed.

(Leave-one-out analysis) Figure 5 displays the performance degradation of leave-one-out pretrained model compared to MORE in linear probing. For example, 'w/o Node' excludes the node-level task while using subgraph-, graph-, and 3D-level tasks. In a leave-one-out setup, the decreased performance indicates that the excluded task is significant and must not be overlooked. ClinTox and BBBP rely heavily on graph-level tasks, BACE on node-level. The task with significant effect varies depending on the datasets, this reminds the importance of exploiting comprehensive molecular information. On average, we observe a large performance degradation at the subgraph- and graph-level, suggesting the benefits of learning a broader range of chemically meaningful molecular structures and high-level semantic informa-

(a) Linear probing (freezing the encoder)

	BBBP	Tox21	ToxCast	SIDER	ClinTox	HIV	BACE	avg (↑)
Infomax	60.8±0.38	67.0±0.40	58.3±0.24	58.2±0.76	62.6±0.28	71.3±1.36	65.1±0.67	63.33
EdgePred	52.7±1.45	63.0±0.77	54.1±0.39	51.7±0.99	48.2±5.46	65.2±1.06	58.6±1.76	56.21
AttrMasking	51.8±0.23	69.3±0.04	57.7±0.08	51.3±0.09	54.5±0.44	60.5±0.31	61.8±0.62	58.13
ContextPred	58.8±0.57	68.3±0.24	58.8±0.36	59.2±0.19	40.0±1.51	67.0±0.62	59.6±2.53	58.81
GraphCL	63.0±0.29	67.6±0.39	57.4±0.48	52.8±0.95	54.7±5.10	64.8±1.69	66.3±0.29	60.94
GraphLoG	54.5±0.30	66.8±0.17	57.4±0.17	58.0±0.58	57.6±1.29	65.2±0.54	72.4±0.65	61.70
GraphMAE	56.5±0.41	66.7±0.45	57.6±0.10	52.0±0.82	44.3±0.57	60.5±0.54	61.8±5.53	57.06
GraphMVP	57.9±0.68	66.9±0.17	58.5±0.09	56.0±1.16	42.7±1.99	67.1±0.83	65.3±0.20	59.20
MORE	**67.9±0.28**	**70.6±0.33**	**62.2±0.16**	**60.7±0.28**	**63.5±1.26**	**72.8±0.32**	**80.0±0.62**	**68.24**

(b) Full fine-tuning (unfreezing the encoder)

	BBBP	Tox21	ToxCast	SIDER	ClinTox	HIV	BACE	avg (↑)
w/o pretrain	65.7±1.89	74.0±0.07	60.9±0.43	56.8±1.09	53.3±2.59	73.5±1.14	66.3±1.56	64.36
Infomax	68.0±1.05	75.0±0.55	62.4±0.57	59.7±0.47	71.1±3.88	76.8±1.38	76.8±1.32	69.97
EdgePred	65.3±1.96	76.3±0.26	63.6±0.28	61.3±0.54	64.5±2.67	75.7±1.05	79.9±1.27	69.51
AttrMasking	63.4±1.45	**76.4±0.19**	63.4±0.52	60.0±0.87	71.1±2.46	76.3±0.28	79.7±0.47	70.04
ContextPred	67.1±0.81	74.4±0.12	63.7±0.12	60.9±0.78	59.0±1.99	76.6±0.45	79.1±2.13	68.69
GraphCL	68.0±2.05	74.5±0.12	62.4±0.59	59.2±1.12	75.3±3.67	76.3±1.09	76.8±0.36	70.36
GraphLoG	66.8±2.55	74.7±0.25	62.4±0.55	59.6±0.67	64.2±1.27	76.6±0.77	82.3±0.42	69.51
GraphMAE	68.0±2.54	75.6±0.27	63.4±0.14	60.2±0.49	70.8±4.15	76.6±0.87	82.1±1.40	70.96
GraphMVP	71.3±1.13	75.0±0.37	63.4±0.26	**62.9±0.29**	68.2±5.95	75.8±1.44	78.8±4.61	70.77
MORE	**71.9±0.94**	75.6±0.54	**64.6±0.58**	60.9±0.62	**81.0±0.65**	**77.0±0.74**	**82.8±1.33**	**73.40**

Table 3: Prediction performance of pretrained models on seven downstream tasks and average performance with 3 repetitions, scaffold splitting, in terms of ROC-AUC (↑) (mean±std in %). We keep the neural network architecture the same, with only the pretraining methods (pretext tasks) being varied. The last column, avg, represents the average performance over the entire dataset. We have marked the best result in bold. (a) Linear probing: freezes the encoder and only the decoder parameters are fine-tuned, (b) Full fine-tuning: unfreezes the encoder, so both of the encoder and decoder parameters are fine-tuned, the first row represents the prediction performance of the model without pretraining, and the remaining rows are the results learning from pretrained models.

tion. Some datasets tend to improve performance when one pretext task is excluded.

(single-task analysis) Figure 6 shows the performance results of MORE and each pretext task executed alone in linear probing. In a single-task setup, the best performance means that the executed task is important. Each downstream task reveals varying trends because the important information is different. For example, ToxCast, SIDER, HIV, and BACE exhibit the best performance at the subgraph-level, BBBP and Tox21 at the graph level, and ClinTox at the 3D-level. We can also observe that the highest single task is similar to MORE performance. On average, MORE is the highest, followed by graph-level.

In both experiments, important pretext tasks vary based on downstream tasks, these results highlight that each task requires a different level of information. In particular, graph information based on molecular descriptors, which has not been actively explored, shows the best effects on average.

Scalability with Dataset Size in Pretraining

Several studies on molecular foundation models and large language models (LLMs) investigate scaling laws between performance and various factors (Ji et al. 2024; Beaini et al. 2023; Hormazabal et al. 2024; Kaplan et al. 2020). Notably, the performance improves as the dataset size increases, highlighting the positive correlation between dataset size and model effectiveness (Hoffmann et al. 2022). To analyze this effect, we prepare five downsampled datasets from ZINC15: around 10M, 25M, 50M, 100M, and 200M. Since GraphMVP is a pretrained method with GEOM (Axelrod and Gomez-Bombarelli 2022), it is excluded from the baseline.

Figure 7 illustrates the average performance as the size of pretraining dataset increases. While other methods exhibit fluctuating performance, MORE consistently improves as the pretraining dataset size increases. In particular, MORE exhibits the highest and most consistent performance improvement in linear probing and outperforms significantly in full fine-tuning. These results suggest the potential of MORE as a foundation model.

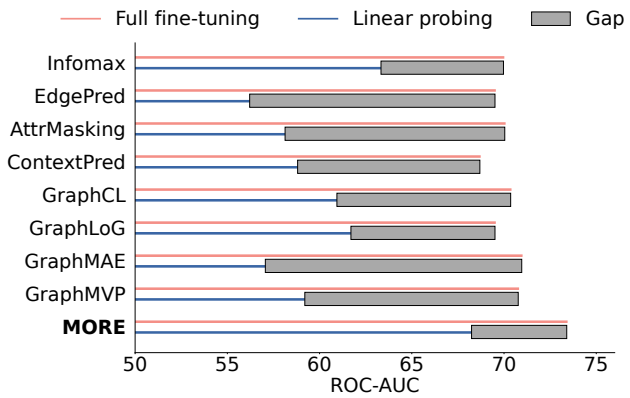


Figure 3: Performance gap between linear probing and full fine-tuning, corresponding to whether the encoder is frozen or unfrozen, respectively.

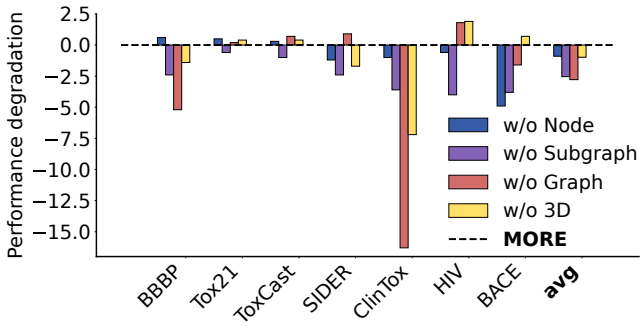


Figure 5: Performance degradation of leave-one-out pretrained model compared to MORE in linear probing. All results are averaged over three repetitions for each dataset, and the last 'avg' represents the average results for all datasets.

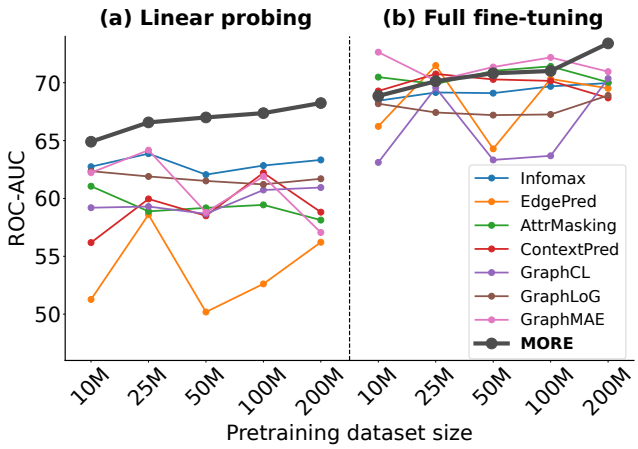


Figure 7: Results of scalability experiments with increasing the size of pretraining dataset. The average performance across 7 downstream tasks, evaluated over 3 repetitions, for (a) linear probing and (b) full fine-tuning. MORE is highlighted with a thick outline.

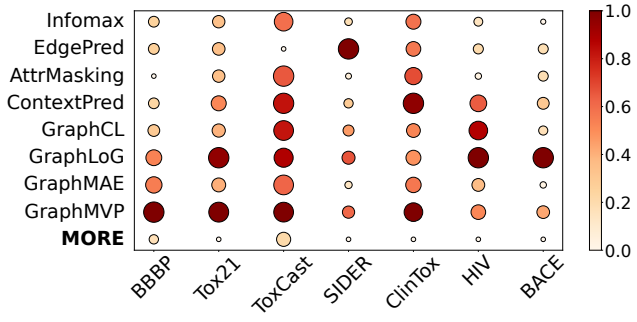


Figure 4: Quantification of changes in encoder parameters due to full fine-tuning. Circle color and size represent the mean and variance, respectively. The darker the color and larger circle, the greater the changes in parameters; the lighter the color and smaller circle, the lesser the change it indicates.

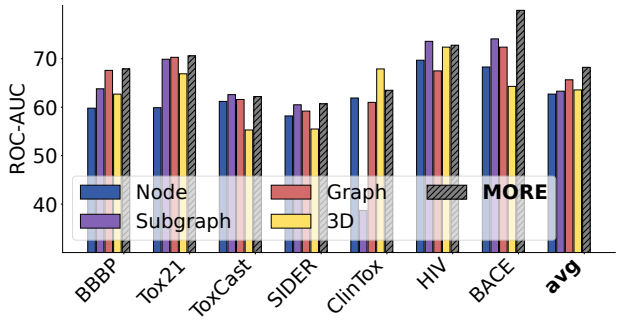


Figure 6: Performance of single-task pretrained model and MORE in linear probing. All results are averaged over three repetitions for each dataset, and the last 'avg' represents the average results for all datasets.

Conclusion

In this paper, we emphasize the importance of the pretext tasks that learn versatile, rich, and generalizable representations to serve as a basic foundation for the molecular domain. We propose **M**ulti-level m**O**lecule g**R**aph pr**E**-train (**MORE**), which integrates four levels of a molecular graph: node-, subgraph- graph-, and 3D-level. MORE learns high-level semantic properties by predicting molecular descriptors as well as considering both local- and global-level information. Compared to the baseline models with the same model structure, MORE demonstrates the ability to learn comprehensive representations, with the best performance. In the quantitative forgetting analysis of pretrained models, MORE shows consistently minimal and stable parameter changes with the smallest performance gap. The consistent performance gains with larger pretraining datasets indicate the potential for development as a foundation model.

This work highlights a multi-level pretext task method for learning generalizable and transferable representations. The proposed method can be applied to other well-known models besides GNNs, potentially leading to greater effects.

Acknowledgments

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2022R1C1C1005065); in part by Institute of Information & communications Technology Planning & Evaluation (IITP) under the Artificial Intelligence Convergence Innovation Human Resources Development (IITP-2025-RS-2023-00254177) and the Leading Generative AI Human Resources Development (IITP-2025-RS-2024-00360227) grant funded by the Korea government(MSIT).

References

Axelrod, S.; and Gomez-Bombarelli, R. 2022. GEOM, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1): 185.

Barnard, T.; Hagan, H.; Tseng, S.; and Sosso, G. C. 2020. Less may be more: an informed reflection on molecular descriptors for drug design and discovery. *Molecular Systems Design & Engineering*, 5: 317–329.

Beaini, D.; Huang, S.; Cunha, J. A.; Li, Z.; Moisescu-Pareja, G.; Dymov, O.; Maddrell-Mander, S.; McLean, C.; Wenkel, F.; Müller, L.; et al. 2023. Towards foundational models for molecular learning on large-scale multi-task datasets. *arXiv preprint arXiv:2310.04292*.

Cavasotto, C. N.; and Scardino, V. 2022. Machine learning toxicity prediction: latest advances by toxicity end point. *ACS omega*, 7: 47536–47546.

Crum-Brown, A.; and Fraser, T. R. 1865. The connection of chemical constitution and physiological action. *Trans R Soc Edinb*, 25: 257.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Durant, J. L.; Leland, B. A.; Henry, D. R.; and Nourse, J. G. 2002. Reoptimization of MDL keys for use in drug discovery. *Journal of chemical information and computer sciences*, 42(6): 1273–1280.

Fang, T.; Zhang, Y.; Yang, Y.; Wang, C.; and Chen, L. 2024. Universal prompt tuning for graph neural networks. *Advances in Neural Information Processing Systems*, 36.

Fang, Y.; Zhang, Q.; Zhang, N.; Chen, Z.; Zhuang, X.; Shao, X.; Fan, X.; and Chen, H. 2023. Knowledge graph-enhanced molecular contrastive learning with functional prompt. *Nature Machine Intelligence*, 5: 542–553.

Halgren, T. A. 1996. Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94. *Journal of computational chemistry*, 17: 490–519.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Hansch, C.; and Fujita, T. 1964. p-σ-π Analysis. A Method for the Correlation of Biological Activity and Chemical Structure. *Journal of the American Chemical Society*, 86: 1616–1626.

Hoffmann, J.; Borgeaud, S.; Mensch, A.; Buchatskaya, E.; Cai, T.; Rutherford, E.; Casas, D. d. L.; Hendricks, L. A.; Welbl, J.; Clark, A.; et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Hormazabal, R.; Ko, S. W.; Yoo, I.; Han, S.; and Bertens, P. 2024. Exploring Neural Scaling Laws in Molecular Pretraining with Synthetic Tasks. In *ICML 2024 AI for Science Workshop*.

Hou, Z.; Liu, X.; Cen, Y.; Dong, Y.; Yang, H.; Wang, C.; and Tang, J. 2022. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 594–604.

Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.; and Leskovec, J. 2019. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*.

Ji, X.; Wang, Z.; Gao, Z.; Zheng, H.; Zhang, L.; Ke, G.; et al. 2024. Uni-Mol2: Exploring Molecular Pretraining Model at Scale. *arXiv preprint arXiv:2406.14969*.

Ji, Z.; Shi, R.; Lu, J.; Li, F.; and Yang, Y. 2022. ReLMole: Molecular representation learning based on two-level graph similarities. *Journal of Chemical Information and Modeling*, 62(22): 5361–5372.

Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T. B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; and Amodei, D. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Kipf, T. N.; and Welling, M. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Landrum, G.; Tosco, P.; Kelley, B.; et al. 2020. rdkit/rdkit: 2020_03_1 (Q1 2020) Release.

Li, C.; Wei, W.; Li, J.; Yao, J.; Zeng, X.; and Lv, Z. 2021. 3DMol-Net: learn 3D molecular representation using adaptive graph convolutional network based on rotation invariance. *IEEE journal of biomedical and health informatics*, 26: 5044–5054.

Liu, S.; Wang, H.; Liu, W.; Lasenby, J.; Guo, H.; and Tang, J. 2021. Pre-training molecular graph representation with 3d geometry. *arXiv preprint arXiv:2110.07728*.

Martins, I. F.; Teixeira, A. L.; Pinheiro, L.; and Falcao, A. O. 2012. A Bayesian approach to in silico blood-brain barrier penetration modeling. *Journal of chemical information and modeling*, 52: 1686–1697.

Moon, K.; Im, H.-J.; and Kwon, S. 2023. 3D graph contrastive learning for molecular property prediction. *Bioinformatics*, 39: btad371.

Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al. 2018. Improving language understanding by generative pre-training. *pre-training*.

Rong, Y.; Bian, Y.; Xu, T.; Xie, W.; Wei, Y.; Huang, W.; and Huang, J. 2020. Self-supervised graph transformer on

large-scale molecular data. *Advances in neural information processing systems*, 33: 12559–12571.

Sterling, T.; and Irwin, J. J. 2015. ZINC 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55: 2324–2337.

Tan, Q.; Liu, N.; Huang, X.; Choi, S.-H.; Li, L.; Chen, R.; and Hu, X. 2023. S2gae: Self-supervised graph autoencoders are generalizable learners with graph masking. In *Proceedings of the sixteenth ACM international conference on web search and data mining*, 787–795.

Veličković, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2018. Deep graph infomax. *arXiv preprint arXiv:1809.10341*.

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

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.

Xu, M.; Wang, H.; Ni, B.; Guo, H.; and Tang, J. 2021. Self-supervised graph-level representation learning with local and global structure. In *International Conference on Machine Learning*, 11548–11558. PMLR.

Xue, L.; and Bajorath, J. 2000. Molecular descriptors in chemoinformatics, computational combinatorial chemistry, and virtual screening. *Combinatorial chemistry & high throughput screening*, 3: 363–372.

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

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

Zhou, F.; and Cao, C. 2021. Overcoming catastrophic forgetting in graph neural networks with experience replay. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 4714–4722.