Ai4science

Wu Hualong

HARBIN INSTITUTE OF TECHNOLOGY,SHEN ZHEN

MOLE-BERT: RETHINKING PRE-TRAINING GRAPH NEURAL NETWORKS FOR MOLECULES

ICLR 2023

Jun Xia,Chengshuai Zhao,Bozhen Hu

AI Lab, Research Center for Industries of the Future, Westlake University

Outline

Background

background

AttrMask: Typically,atom types as node attributes are randomly masked, and GNNs are then trained to predict masked types as in AttrMask

- Following the Masked Language Modeling (MLM) task of BERT
- However, unlike MLM with a large vocabulary, the AttrMask pre-training does not learn informative molecular representations due to small and unbalanced atom 'vocabulary'

To amend this problem, The artical propose a variant of VQ-VAE as a context-aware tokenizer to encode atom attributes into chemically meaningful discrete codes.
With the enlarged atom 'vocabulary',the author then propose a novel pre-training framework: Mole-BERT.

VQ-VAE

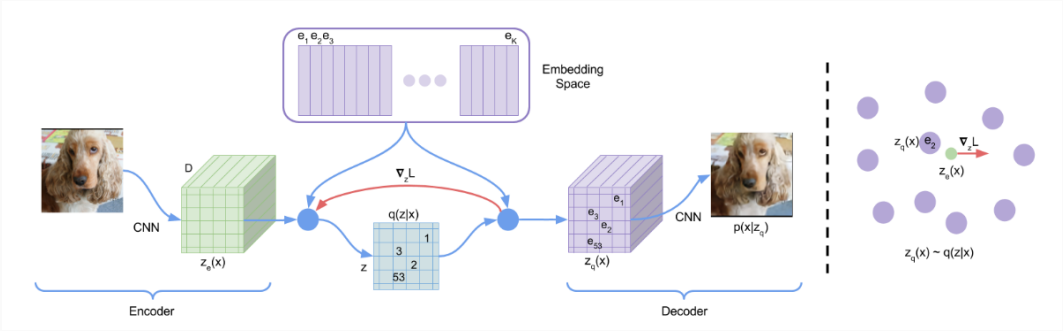
Variational Autoencoder (VAE):A Variational Autoencoder (VAE) is a type of generative model that learns to encode data into a latent space and then decodes it back to the original data.

- Encoder: Maps input data to a latent space, outputting a mean and variance for each latent dimension
- Decoder: Reconstructs the data from the latent space.

The training objective of a VAE includes a reconstruction loss (to ensure the output resembles the input) and a regularization term (KL divergence).
Vector Quantization (VQ):Vector Quantization is a classical signal processing technique that maps continuous input vectors to the nearest vector in a discrete codebook.

VQ-VAE

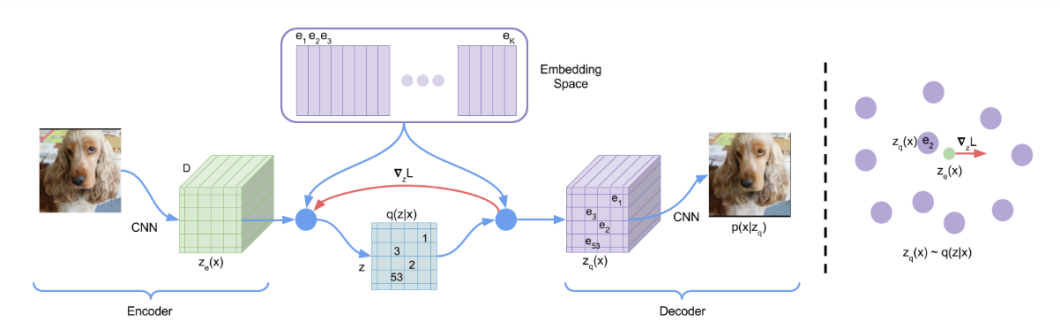
- encoder output:$z_e(x)$
- define a latent embedding space: $e \in R^{K \times D}$,the embedding keeps the same dimension with the encoder output



VQ-VAE

- The posterior categorical distribution $q(z|x)$ probabilities are defined as one-hot as follows:

$$q(z = k|x) = \begin{cases} 1 & \text{for k} = \text{argmin}_j \|z_e(x) - e_j\|_2 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

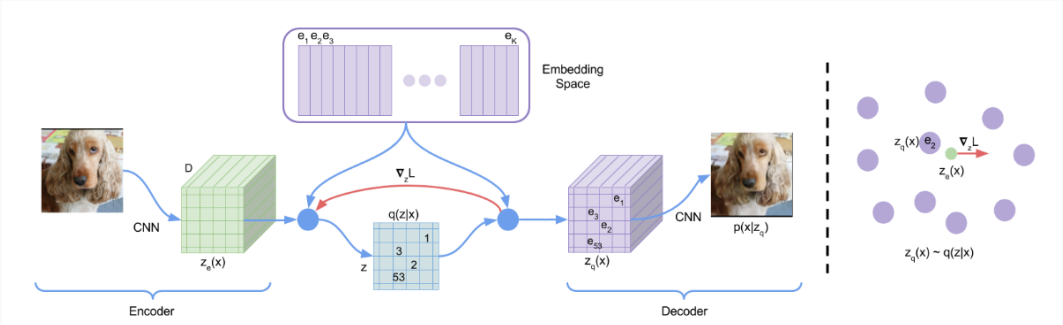


VQ-VAE

- The representation $z_e(x)$ is mapped onto the nearest element of embedding e:

$$z_q(x) = e_k, \quad \text{where} \quad k = \text{argmin}_j \|z_e(x) - e_j\|_2 \tag{2}$$

Note that there is no real gradient defined for equation, approximate the gradient similar to the straight-through estimator,just copy gradients from decoder input $z_q(x)$ to encoder output $z_e(x)$



VQ-VAE

Equation 3 specifies the overall loss function:

$$L = \log p(x|z_q(x)) + \|\mathrm{sg}[z_e(x)] - e\|_2^2 + \beta\|z_e(x) - \mathrm{sg}[e]\|_2^2 \tag{3}$$

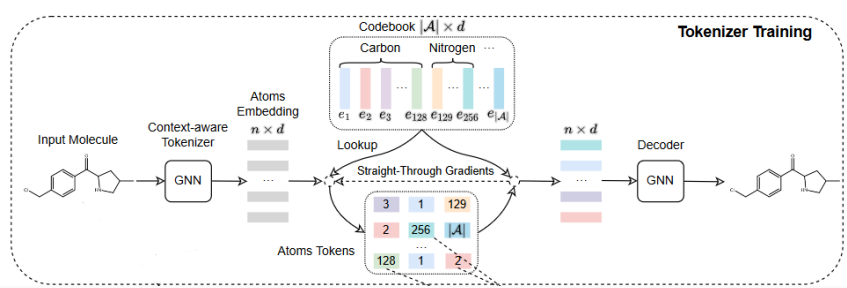
- The first term is the reconstruction loss which optimizes the decoder and the encoder.
- The second term uses the l_2 error to move the embedding vectors e_i towards the encoder outputs $z_e(x)$
- To make sure the encoder commits to an embedding and its output does not grow,the third term serves as a commitment loss.

Proposed Pre-Training Framework:Mole-BERT

Tokenizer Training

Formally,the atoms $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$ of a molecule graph G is tokenized to $z = \{z_1, z_2, \cdots, z_n\} \in \mathcal{A}^n$

- Firstly, the GNN encoder of group VQ-VAE encodes the atoms to atoms embeddings.
- Next, the vector quantizer (VQ) looks up the nearest neighbor in the codebook for each atom embedding h_i



Tokenizer Training

Let $\{e_1, e_2, \cdots, e_{|\mathcal{A}|}\}$ denote the codebook embeddings.

- The quantized code of the i-th atom is

$$z_i = \text{argmin}_j \left\| h_i - e_j \right\|_2 \tag{4}$$

- After quantizing the atoms to discrete tokens, we feed the corresponding codebook embeddings $\{e_{z_1}, e_{z_2}, \cdots, e_{z_n}\}$ to the decoder to reconstruct the input molecule graph
- With the input attributes v_i and reconstructed attributes $\widehat{v}_i$, the training loss of the tokenizer for the molecule graph G is:

$$\mathcal{L}_{\mathbf{VQ}} = \frac{1}{n}\sum_{i=1}^{n}\left(1 - \frac{v_i^T \widehat{v}_i}{\|v_i\| \cdot \|\widehat{v}_i\|}\right)^{\gamma} + \frac{1}{n}\sum_{i=1}^{n}\|\text{sg}[h_i] - e_{z_i}\|_2^2 + \frac{\beta}{n}\sum_{i=1}^{n}\|\text{sg}\left[e_{z_i}\right] - h_i\|_2^2 \tag{5}$$

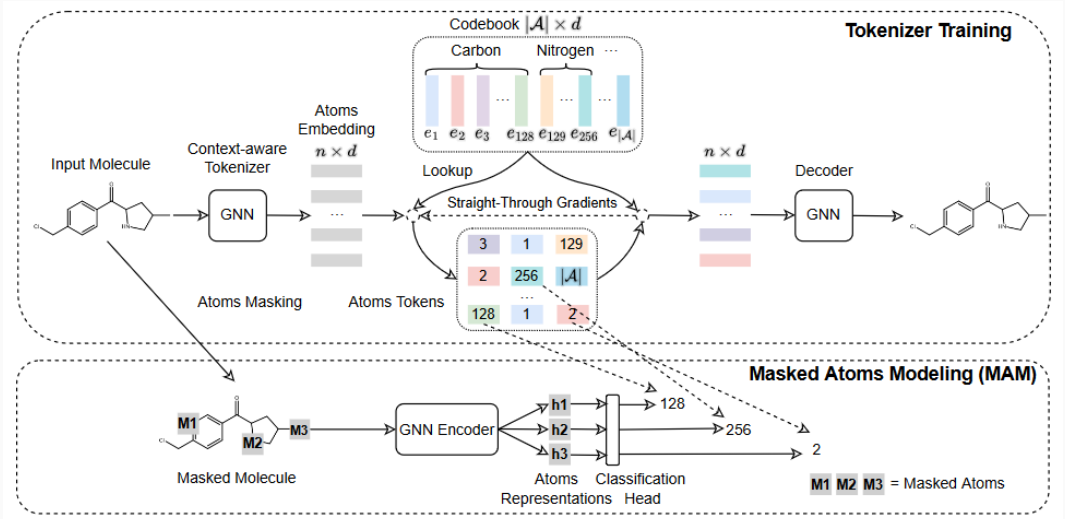
Tokenizer Training

Observe that atoms of different types might be allocated with the same token id with vanilla VQ-VAE.As a remedy, author introduce a group VQ-VAE to address this issue.

- Specifically,divide the codebook embeddings into several groups, each of which corresponds to specific atom types.
- For example, the quantized codes of carbon, nitrogen and oxygen are restricted to [1, 128], [129, 256] and [257, 384], respectively. The left rare atoms are restricted to [385, 512] because they are less likely to conflict with each other.

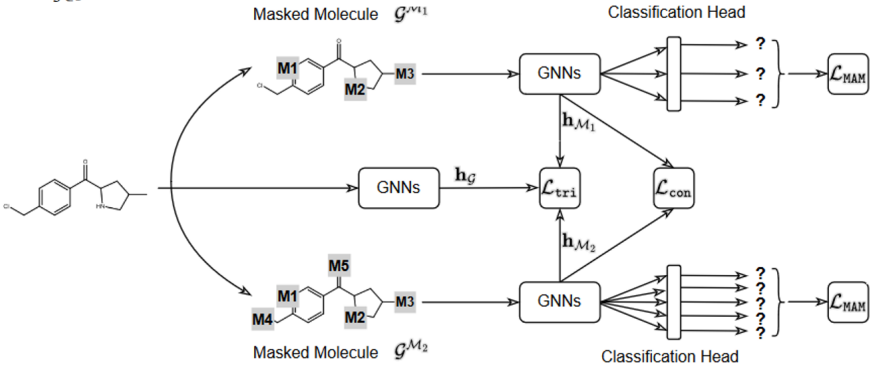
MASKED ATOMS MODELING (MAM)

With the new tokenizer, author propose MAM to pre-train GNNs.



TRIPLET MASKED CONTRASTIVE LEARNING (TMCL)

$$\mathcal{L}_{\texttt{tri}} = \sum_{\mathcal{G}\in\mathcal{D}} \max\left(sim\left(\mathbf{h}_{\mathcal{G}}, \mathbf{h}_{\mathcal{M}_2}\right) - sim\left(\mathbf{h}_{\mathcal{G}}, \mathbf{h}_{\mathcal{M}_1}\right), 0\right)$$



$$\mathcal{L}_{\texttt{TMCL}} = \mathcal{L}_{\texttt{con}} + \mu\mathcal{L}_{\texttt{tri}}, \quad \text{where} \quad \mathcal{L}_{\texttt{con}} = -\sum_{\mathcal{G}\in\mathcal{D}} \log \frac{e^{sim(\mathbf{h}_{\mathcal{M}_1}, \mathbf{h}_{\mathcal{M}_2})/\tau}}{\sum_{\mathcal{G}'\in\mathcal{B}} e^{sim(\mathbf{h}_{\mathcal{M}_1}, \mathbf{h}_{\mathcal{G}'})/\tau}}$$

$$\mathcal{L}_{\texttt{Mole-BERT}} = \mathcal{L}_{\texttt{MAM}} + \mathcal{L}_{\texttt{TMCL}}$$

Experiments

Experiments

Datasets:

- For the pre-training stage, author uses 2 million molecules sampled from the ZINC15 database
- The main downstream task is molecular property prediction, where author adopts the widely-used 8 binary classification datasets contained in MoleculeNet

Author uses a 5-layer Graph Isomorphism Networks (GINs) whose hidden dimension is 300 as the backbone architecture, which is one of the state-of-the-art GNNs for graph-level tasks.

Results for molecular property prediction (classification)

	Tox21	ToxCast	Sider	ClinTox	MUV	HIV	BBBP	Bace	Average
# Molecules	7,831	8,575	1,427	1,478	93,087	41,127	2,039	1,513	-
No pretrain	74.6 (0.4)	61.7 (0.5)	58.2 (1.7)	58.4 (6.4)	70.7 (1.8)	75.5 (0.8)	65.7 (3.3)	72.4 (3.8)	67.15
InfoGraph (Sun et al., 2020b)	73.3 (0.6)	61.8 (0.4)	58.7 (0.6)	75.4 (4.3)	74.4 (1.8)	74.2 (0.9)	68.7 (0.6)	74.3 (2.6)	70.10
GPT-GNN (Hu & others., 2020)	74.9 (0.3)	62.5 (0.4)	58.1 (0.3)	58.3 (5.2)	75.9 (2.3)	65.2 (2.1)	64.5 (1.4)	77.9 (3.2)	68.45
EdgePred (Hamilton et al., 2017)	76.0 (0.6)	64.1 (0.6)	60.4 (0.7)	64.1 (3.7)	75.1 (1.2)	76.3 (1.0)	67.3 (2.4)	77.3 (3.5)	70.08
ContextPred (Hu et al., 2020)	73.6 (0.3)	62.6 (0.6)	59.7 (1.8)	74.0 (3.4)	72.5 (1.5)	75.6 (1.0)	70.6 (1.5)	78.8 (1.2)	70.93
GraphLoG (Xu et al., 2021a)	75.0 (0.6)	63.4 (0.6)	59.6 (1.9)	75.7 (2.4)	75.5 (1.6)	76.1 (0.8)	68.7 (1.6)	78.6 (1.0)	71.56
G-Contextual (Rong et al., 2020b)	75.0 (0.6)	62.8 (0.7)	58.7 (1.0)	60.6 (5.2)	72.1 (0.7)	76.3 (1.5)	69.9 (2.1)	79.3 (1.1)	69.34
G-Motif (Rong et al., 2020b)	73.6 (0.7)	62.3 (0.6)	61.0 (1.5)	77.7 (2.7)	73.0 (1.8)	73.8 (1.2)	66.9 (3.1)	73.0 (3.3)	70.16
AD-GCL (Suresh et al., 2021)	74.9 (0.4)	63.4 (0.7)	61.5 (0.9)	77.2 (2.7)	76.3 (1.4)	76.7 (1.2)	70.7 (0.3)	76.6 (1.5)	72.16
JOAO (You et al., 2021)	74.8 (0.6)	62.8 (0.7)	60.4 (1.5)	66.6 (3.1)	76.6 (1.7)	76.9 (0.7)	66.4 (1.0)	73.2 (1.6)	69.71
SimGRACE (Xia et al., 2022b)	74.4 (0.3)	62.6 (0.7)	60.2 (0.9)	75.5 (2.0)	75.4 (1.3)	75.0 (0.6)	71.2 (1.1)	74.9 (2.0)	71.15
GraphCL (You et al., 2020)	75.1 (0.7)	63.0 (0.4)	59.8 (1.3)	77.5 (3.8)	76.4 (0.4)	75.1 (0.7)	67.8 (2.4)	74.6 (2.1)	71.16
GraphMAE (Hou et al., 2022)	75.2 (0.9)	63.6 (0.3)	60.5 (1.2)	76.5 (3.0)	76.4 (2.0)	76.8 (0.6)	71.2 (1.0)	78.2 (1.5)	72.30
3D InfoMax (Stärk et al., 2022)	74.5 (0.7)	63.5 (0.8)	56.8 (2.1)	62.7 (3.3)	76.2 (1.4)	76.1 (1.3)	69.1 (1.2)	78.6 (1.9)	69.69
GraphMVP (Liu et al., 2022a)	74.9 (0.8)	63.1 (0.2)	60.2 (1.1)	**79.1** (2.8)	77.7 (0.6)	76.0 (0.1)	70.8 (0.5)	79.3 (1.5)	72.64
MGSSL (Zhang et al., 2021)	75.2 (0.6)	63.3 (0.5)	61.6 (1.0)	77.1 (4.5)	77.6 (0.4)	75.8 (0.4)	68.8 (0.6)	78.8 (0.9)	72.28
AttrMask (Hu et al., 2020)	75.1 (0.9)	63.3 (0.6)	60.5 (0.9)	73.5 (4.3)	75.8 (1.0)	75.3 (1.5)	65.2 (1.4)	77.8 (1.8)	70.81
MAM (with vanilla VQ-VAE)	75.8 (0.6)	63.1 (0.5)	60.7 (1.5)	74.2 (2.7)	76.5 (1.6)	76.2 (0.9)	66.4 (0.7)	78.2 (0.8)	71.39
TMCL (w/o $\mathcal{L}_{\texttt{con}}$)	73.5 (1.0)	61.8 (0.3)	58.7 (1.6)	61.1 (4.1)	71.6 (1.3)	73.5 (1.3)	65.4 (2.6)	73.7 (2.4)	67.41
TMCL (w/o $\mathcal{L}_{\texttt{tri}}$)	74.1 (0.4)	62.4 (0.8)	58.7 (3.0)	75.6 (2.2)	75.7 (1.1)	74.6 (1.1)	66.8 (1.4)	74.2 (1.3)	70.26
MAM	76.2 (0.5)	63.9 (0.3)	61.4 (1.9)	75.1 (3.0)	77.4 (2.1)	77.5 (1.0)	66.8 (1.5)	78.9 (1.1)	72.16
TMCL	74.9 (0.7)	63.2 (0.7)	59.6 (1.4)	77.0 (4.2)	77.2 (0.3)	75.3 (1.1)	67.6 (1.3)	75.1 (1.2)	71.24
Mole-BERT	**76.8** (0.5)	**64.3** (0.2)	**62.8** (1.1)	78.9 (3.0)	**78.6** (1.8)	**78.2** (0.8)	**71.9** (1.6)	**80.8** (1.4)	**74.04**

Experiments

Model	GCN	GIN	R-GCN	GraphSAGE
No pretrain	68.77	67.15	68.32	68.46
MAM	71.35	72.16	70.76	71.55
TMCL	68.93	71.24	69.25	69.58
Mole-BERT	73.22	74.04	73.51	73.74
Relative gain	+6.47 %	+10.26 %	+7.60 %	+7.71 %

Figure 2: different GNN architectures

Vocabulary size	128	256	512	1, 024	2, 048
MAM	71.42	71.65	72.16	**72.21**	71.66
Mole-BERT	73.23	73.56	**74.04**	74.02	73.86

Figure 3: The performance with various vocabulary sizes