Graph2Token: Make LLMs Understand Molecule Graphs

Runze Wang [1] **Mingqi Yang** [2] **Yanming Shen** [1]

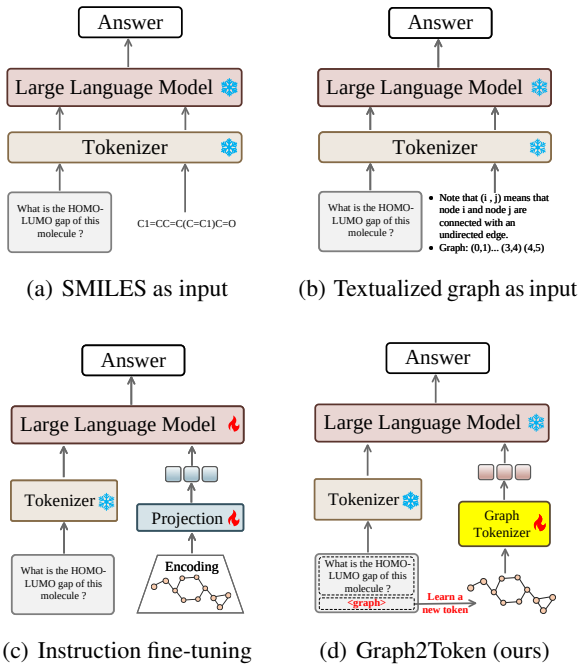
Abstract

Large language models (LLMs) excel at various text-related tasks. However, it is still challenging for them to process graph data such as molecules. To bridge this gap, this paper proposes Graph2Token, an efficient solution that aligns a graph token to LLM tokens. The key idea is to represent a graph token with the LLM token vocabulary, without finetuning the backbone of LLM. In this way, we can unleash the potential of existing LLMs, which helps the downstream molecule prediction tasks. Extensive experiments demonstrate the effectiveness of our proposed Graph2Token. Code is available at https://github.com/ZeLeBron/Graph2Token.

1. Introduction

Large language models (LLMs) are primarily designed for textual data processing. Their capability to handle graph-structured data such as molecules is not clearly defined. This presents a challenge as graph data requires a different approach compared to textual data. Extending the functionality of LLMs to effectively process and analyze molecules will open up opportunities for molecule related tasks.

To apply LLMs for molecule tasks, existing solutions often involve converting molecule structures into a format that can be processed by the model. One common approach is to use Simplified Molecular Input Line Entry System (SMILES) notation, which represents molecules as text strings (Fig. 1.(a)). Zhao et al. (2023b) employs SMILES as a molecule representation and utilizes in-context learning to guide ChatGPT in understanding molecule structures. However, a significant limitation of LLMs is their lack of understanding of molecule representations in SMILES strings, which in many cases leads to inaccurate or inconsistent results (Zhao et al., 2023b). This highlights the challenge of using SMILES as a representation for molecule data within the context of LLMs.



(a) SMILES as input (b) Textualized graph as input

(c) Instruction fine-tuning (d) Graph2Token (ours)

Figure 1. Different approaches of feeding molecules to LLMs.

Another line of methods involves converting molecule graphs into textual representations before feeding it to the model (Fig. 1.(b)). These methods typically involve describing the adjacency relationships between nodes of the graph and representing the properties of nodes using text (Wang et al., 2024; Fatemi et al., 2023; Zhao et al., 2023a; Liu & Wu, 2023). Combined with zero-shot or more advanced few-shot learning techniques, as well as prompting methods, they guide LLMs in understanding complex topological representations of the graph. This approach leverages textual representations to bridge the gap between graph structures and the language understanding capabilities of large language models, thereby facilitating the comprehension of intricate molecule architectures. However, the pure textual representation of structured data is insufficient for conducting graph reasoning using LLMs.

[1]School of Computer Science and Technology, Dalian University of Technology, China [2]School of Computing, National University of Singapore, Singapore. Correspondence to: Yanming Shen <shen@dlut.edu.cn>.

Proceedings of the ICML 2024 Workshop on Accessible and Efficient Foundation Models for Biological Discovery, Vienna, Austria. 2024. Copyright 2024 by the author(s).

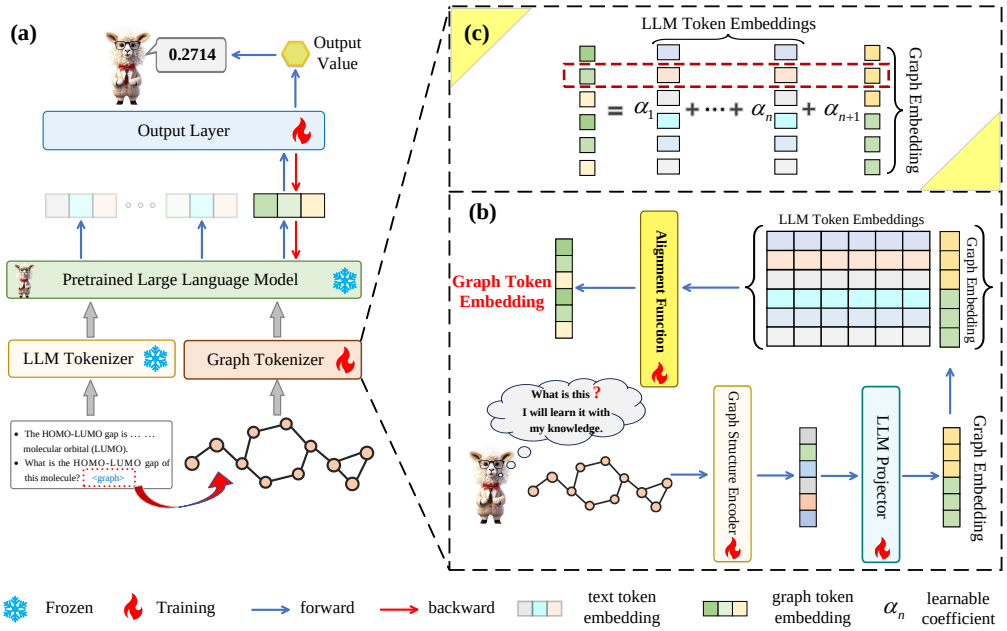


Figure 2. The model architecture of Graph2Token. Given an input instruction and a graph, we first (a) tokenize and embed them via LLM tokenizer and designed graph tokenizer. The graph tokenizer aligns the unknown graph token to the LLM embedding space through (b) encoding graph structures and projecting them into joint representation space with LLM. After that, the trainable alignment function (c) is used to learn a set of coefficients to weight LLM token vocabulary, aiming to translate the graph token embedding from the joint space to LLM space. Then, the graph token embedding and instruction embeddings are collectively fed into the frozen LLM backbone. Finally, the graph token embedding is extracted and fed into the output layer to make graph-level predictions.

Given the limitations of structured data in textual representations, researchers are exploring the use of instruction fine-tuning (Fig. 1.(c)). This involves leveraging the relationship between structured data and textual descriptions to align them in embedding space by fine-tuning a small number of parameters. This approach aims to address the inadequacies of representing structured data in text by aligning it more closely with textual descriptions through targeted adjustments in the model. Cao et al. (2023) leverages a linear mapping alignment approach, with the aim of fully mapping molecule graph structures into the embedding space of LLM. When addressing downstream tasks, they finetune the alignment model and selectively generalize some parameters of LLM to different tasks. Similarly, Liu et al. (2023b) employs the more complex mapping technique Qformer (Li et al., 2023) to achieve cross-modal mapping from molecule graphs to text embedding space. The aforementioned work relies heavily on high-quality molecule-text paired datasets. However, in the field of biology, relevant data is often scarce, hindering the ability to provide the required quantity and quality of data as in the field of computer vision. As a result, the full potential of large language models in tasks related to biological molecules cannot be realized.

In this paper, we propose Graph2Token, a simple and efficient solution, which aligns graph tokens to large language model tokens. *The key idea is to learn a graph token representation using the LLM token vocabulary.* In this way, a graph token can be naturally adapted by the LLM, without the costly fine-tuning. Intuitively, for LLMs to comprehend an unseen graph token from scratch, the ideal scenario is to generate a new representation rooted in their existing knowledge. Building upon this insight, we propose a novel alignment strategy that utilizes a learnable combination of tokens pre-trained by LLM to represent the graph tokens. Graph2Token achieves the SOTA performance on molecule property prediction tasks with only a fraction of the trainable parameters (fewer than 4.2 million) typically required by existing methods. Our main contributions include:

- We design a lightweight token alignment approach that can adapt a molecule graph token to LLMs.
- By extensive experiments, we show that the proposed approach achieves superior performance.

2. Method

Our model architecture is illustrated in Figure 2, which leverages the vocabulary of LLM to learn a graph token representation, enabling the LLM to understand a graph token and accomplish molecule-level tasks without the need

of fine-tuning the backbone model. The input can be formalized as texts like: "*The HOMO-LUMO gap is the energy difference between the highest occupied molecule orbital (HOMO) and the lowest unoccupied molecule orbital (LUMO) in a molecule. What is the HOMO-LUMO gap of this molecule? <graph>*". It can be seen that the input to LLMs consist of task-relevant instructions and a special graph token. The instruction serves as a straightforward yet effective way to task-specific activation of LLMs. It can be directly translated into token embedding by LLM tokenizer. For the placeholder *<graph>*, it represents the molecule graph token that corresponds to the graph structure embedding. However, the mismatch between graph embeddings and the semantic space of LLM has become a barrier to the understanding of graph token. To bridge the gap, we define a graph tokenizer towards translating graph structure into representations that LLMs can understand.

2.1. Graph Encoder

Considering a molecule graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$. Let $\boldsymbol{X}$ and $\boldsymbol{E}$ be the node and edge feature matrix. The graph tokenizer maps the molecule graph into the LLM embedding space with three steps. First, since a graph structure encoder needs to effectively extract node representations while preserving connectivity information of the molecule graphs to accurately capture these features, we apply a graph neural network (GNN) (Zhou et al., 2022) as the initial encoder:

$$\mathbf{g} = \text{GNN}(\mathcal{V}, \mathcal{E}, \boldsymbol{X}, \boldsymbol{E}), \tag{1}$$

where $\mathbf{g} \in \mathbb{R}^d$ is the graph embedding with d dimensions. It's worth noting that our method is not limited to the use of specific GNN. In general, GNN models with greater expressive power possess better capabilities to represent graph topology (Zhang et al., 2023), enabling them to encode structural information more relevant to molecular property.

2.2. Graph Token Alignment

Next, in order to project graph features to the joint representation space of LLM, a lightweight layer is introduced:

$$\mathbf{h} = \mathbf{W}_g * \mathbf{g} + \mathbf{b}_g, \tag{2}$$

where $\mathbf{h} \in \mathbb{R}^{d^*}$ is the graph token embedding with d^* being the dimension of LLM token embeddings.

Initially, a graph token embedding is incomprehensible for LLMs. To address this issue, we propose an innovative alignment strategy for aligning graph tokens with the semantic space of large language models. This approach harnesses the pre-trained token embeddings from LLM and trains a linear combination function to redefine the representation of a graph token, which effectively bridges the gap between graph structure and the semantic understanding of LLMs.

The proposed alignment strategy starts by augmenting the vocabulary of the large language model with graph token embeddings from a joint representation space. Subsequently, all token embeddings in the augmented vocabulary are retrieved, and each is multiplied by a learnable coefficient. Finally, the weighted token embeddings are aggregated across corresponding feature channels to yield a new representation of the graph token. Given the LLM token vocabulary $\mathbf{C}$ and graph token embedding $\mathbf{h}$, the learnable alignment process is defined as follows:

$$\mathbf{z} = (\mathbf{W}_{align}([\mathbf{C}^\top || \mathbf{h}^\top]) + \mathbf{b}_{align})^\top, \tag{3}$$

where $\mathbf{C} \in \mathbb{R}^{N_{|\mathbf{C}|} \times d^*}$ is the LLM token vocabulary with $N_{|\mathbf{C}|}$ indicates the vocabulary size. $[\cdot||\cdot]$ represents the concatenation operation. A detailed illustration of the above process is shown in Figure 2(c). $\mathbf{z} \in \mathbb{R}^{d^*}$ is the final translated graph token embedding in the LLM embedding space. After the alignment, the embedding of graph token is concatenated with the instruction and then fed into the LLMs.

2.3. Output Layer

Upon packaging and forwarding the instructions and graph structure embeddings through the frozen LLM backbone, we discard the prefix portion to obtain the output representation aiming to adapt the graph level tasks. Subsequently, we flatten these representations and apply a linear projection to derive the final predictions.

We can see that the trainable parameters in our Graph2Token primarily consist of the graph tokenizer module and the output layer, which are negligible compared to the parameters of LLMs. By having the original parameters of the LLMs frozen, Graph2Token preserves their inherent semantics and functionality.

3. Experiment

In this section, we evaluate Graph2Token on the graph-level tasks. Details of the datasets are given in Appendix A.

Table 1. Performance (Mean Absolute Error) comparison with LLM-based Generalist Models on the three molecule regression tasks (HOMO, LUMO, $\Delta\epsilon$) on QM9 dataset. The best results are in bold, and the second best are underlined.

Method	HOMO ↓	LUMO ↓	$\Delta\epsilon$ ↓	AVG ↓
LLama2-7B	0.7367	0.8641	0.5152	0.7510
Vicuna-13B	0.7135	3.6807	1.5407	1.9783
Mol-Instruction	0.0210	0.0210	0.0203	0.0210
InstructMol-G	0.0060	0.0070	0.0082	0.0070
InstructMol-GS	<u>0.0048</u>	<u>0.0050</u>	**0.0061**	<u>0.0050</u>
Ours	**0.0040**	**0.0039**	<u>0.0063</u>	**0.0047**

Table 2. Results (ROC-AUC) on molecule classification tasks on three types of datasets: pharmacokinetic (BBBP), bio-activity (BACE, HIV), toxicity (TOX21). The best results are in bold, and the second best are underlined.

Method Type	Method	BBBP ↑	BACE ↑	HIV ↑	TOX21 ↑	Avg↑
Supervised Learning	GIN	67.8	76.8	76.5	73.9	73.8
	GT	68.7	77.2	74.2	75.5	73.9
Graph Pretrain Finetuning	GraphMVP-C	72.4	81.2	77.0	74.4	76.3
	Mole-BERT	70.8	79.3	76.0	75.9	75.5
	MolFM	<u>72.9</u>	83.9	<u>78.8</u>	<u>77.2</u>	<u>78.2</u>
	SimSGT	72.3	83.6	77.7	75.7	77.3
LLM-Based Tuning	Llama-2-7B-chat	65.6	74.8	62.3	-	67.6
	Vicuna-v1.3-7B	60.1	68.3	58.1	-	62.6
	MolCA-S	70.8	79.3	-	76.0	75.4
	MolCA-GS	70.0	79.8	-	<u>77.2</u>	75.7
	InstructMol-G	64.0	**85.9**	74.0	-	74.6
	InstructMol-GS	70.0	82.3	68.9	-	73.7
	Ours	**73.5**	<u>85.0</u>	**79.4**	**79.2**	**79.3**

As explained, Graph2Token serves as a general graph token alignment method that does not rely on specific graph encoders or LLMs. In this work, we employ the Graph Isomorphism Network (GIN) (Xu et al., 2018) as the graph encoder and Vicuna-v1.5 7B (Chiang et al., 2023) as LLM backbone. To ensure a fair comparison with previous GNN-based baselines, we follow the same setup as in GraphMVP (Liu et al., 2021), which used single features of atoms and bonds. For the molecule datasets, we adopt the scaffold splitting way to divide the data into training, validation, and test sets with a ratio of 0.8, 0.1, and 0.1, respectively. Additional setup is given in Appendix C.

3.1. Experiment 1: Overall Performance

In this experiment, we train Graph2Token on molecule datasets for graph regression and classification tasks. Details of baseline models are provided in Appendix B. All baseline results are quoted from (Liu et al., 2024; Cao et al., 2023). As we can see, Graph2Token achieves the superior performance on both graph classification tasks and regression tasks compared with the existing methods, especially for the methods based on LLM-Based tuning. It demonstrates the advantage of Graph2Token in considering the alignment of graph structures to the LLM tokens.

3.2. Experiment 2: Few-shot Performance

LLMs have exhibited remarkable capabilities in few-shot learning (Liu et al., 2023a). In this section, we assess whether Graph2Token can maintain the few-shot learning capabilities when translating graph structures into LLM embedding space. The experiments are conducted to use only 5% and 10% of the full training set. The results are shown in Tab. 3. It can be observed that Graph2Token has a significant advantage in few-shot learning scenario on graph tasks, achieving average improvements of 8.63% and 6.11% compared to GIN, respectively, when using 5% and 10% of the training samples. The ability of Graph2Token that generalize to few-shot graph learning is indeed attributed to graph tokens alignment and the inherent generalization capabilities of LLMs. Our approach of aligning graph tokens to the LLM embedding space can unlock its potential for graph-level tasks.

Table 3. Few-shot learning on 5% and 10% training data. Results on classification dataset using roc-auc as metric.

Ratio	Dataset	Ours	GIN	GCN
5%	BBBP	**64.7**	61.8	64.4
	BACE	**73.2**	64.4	65.1
	HIV	**68.5**	66.2	62.7
	TOX21	**70.6**	62.6	58.4
10%	BBBP	**69.5**	66.9	67.0
	BACE	**74.6**	68.1	64.6
	HIV	**69.7**	66.9	60.0
	TOX21	**71.2**	66.7	68.4

4. Conclusion

In this work, we propose Graph2Token, which aims to make LLMs understand a graph via aligning a graph token to LLM token. Graph2Token is a lightweight solution that adapts a graph token to LLMs without costly fine-tuning while achieving competitive performance.

References

Cao, H., Liu, Z., Lu, X., Yao, Y., and Li, Y. Instructmol: Multi-modal integration for building a versatile and reliable molecular assistant in drug discovery. *arXiv preprint arXiv:2311.16208*, 2023.

Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

Fang, Y., Liang, X., Zhang, N., Liu, K., Huang, R., Chen, Z., Fan, X., and Chen, H. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. *arXiv preprint arXiv:2306.08018*, 2023.

Fatemi, B., Halcrow, J., and Perozzi, B. Talk like a graph: Encoding graphs for large language models. *arXiv preprint arXiv:2310.04560*, 2023.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Li, J., Li, D., Savarese, S., and Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Liu, C. and Wu, B. Evaluating large language models on graphs: Performance insights and comparative analysis. *arXiv preprint arXiv:2308.11224*, 2023.

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

Liu, X., McDuff, D., Kovacs, G., Galatzer-Levy, I., Sunshine, J., Zhan, J., Poh, M.-Z., Liao, S., Di Achille, P., and Patel, S. Large language models are few-shot health learners. *arXiv preprint arXiv:2305.15525*, 2023a.

Liu, Z., Li, S., Luo, Y., Fei, H., Cao, Y., Kawaguchi, K., Wang, X., and Chua, T.-S. Molca: Molecular graph-language modeling with cross-modal projector and uni-modal adapter. *arXiv preprint arXiv:2310.12798*, 2023b.

Liu, Z., Shi, Y., Zhang, A., Zhang, E., Kawaguchi, K., Wang, X., and Chua, T.-S. Rethinking tokenizer and decoder in masked graph modeling for molecules. *Advances in Neural Information Processing Systems*, 36, 2024.

Luo, Y., Yang, K., Hong, M., Liu, X., and Nie, Z. Molfm: A multimodal molecular foundation model. *arXiv preprint arXiv:2307.09484*, 2023.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Wang, H., Feng, S., He, T., Tan, Z., Han, X., and Tsvetkov, Y. Can language models solve graph problems in natural language? *Advances in Neural Information Processing Systems*, 36, 2024.

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

Wu, Z., Jain, P., Wright, M., Mirhoseini, A., Gonzalez, J. E., and Stoica, I. Representing long-range context for graph neural networks with global attention. *Advances in Neural Information Processing Systems*, 34:13266–13279, 2021.

Xia, J., Zhao, C., Hu, B., Gao, Z., Tan, C., Liu, Y., Li, S., and Li, S. Z. Mole-bert: Rethinking pre-training graph neural networks for molecules. In *The Eleventh International Conference on Learning Representations*, 2022.

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

Zhang, B., Fan, C., Liu, S., Huang, K., Zhao, X., Huang, J., and Liu, Z. The expressive power of graph neural networks: A survey. *arXiv preprint arXiv:2308.08235*, 2023.

Zhao, J., Zhuo, L., Shen, Y., Qu, M., Liu, K., Bronstein, M., Zhu, Z., and Tang, J. Graphtext: Graph reasoning in text space. *arXiv preprint arXiv:2310.01089*, 2023a.

Zhao, L., Edwards, C., and Ji, H. What a scientific language model knows and doesn't know about chemistry. In *NeurIPS 2023 AI for Science Workshop*, 2023b.

Zhou, Y., Zheng, H., Huang, X., Hao, S., Li, D., and Zhao, J. Graph neural networks: Taxonomy, advances, and trends. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(1):1–54, 2022.

A. Datasets Details.

All the datasets used in the experiment are from MoleculeNet (Wu et al., 2018) and can be classified as those for pharmacokinetic, bio-activity, toxicity and quantum chemistry. The detailed information is as follows:

- BBBP: Blood-brain barrier penetration (membrane permeability).
- BACE: Qualitative binding results for a set of inhibitors of human β-secretase 1.
- HIV: Experimentally measured abilities to inhibit HIV replication.
- TOX21: Toxicity data on 12 biological targets, including nuclear receptors and stress response pathways.
- QM9: A quantum chemistry dataset containing approximately 134,000 molecules in equilibrium states, covering various physical and chemical properties.

B. Baselines.

The baseline models used for comparisons include:

- Supervised learning: GIN (Xu et al., 2018); GT (Wu et al., 2021).
- GNN Pretrain Finetuning:
  Graph Pre-training method GraphMVP-C (Liu et al., 2021) combined with 3D geometric information;
  Graph pre-training methods (Xia et al., 2022; Liu et al., 2024) based on attribute masking;
  Multi-modal Graph Pre-training Method MolFM (Luo et al., 2023).
- LLM-based Tuning:
  Lora finetuning (Hu et al., 2021) for Llama-2-7B-chat (Touvron et al., 2023), Vicuna-v1.3-7B (Chiang et al., 2023) and Mol-Instruction (Fang et al., 2023); Cross-modal mapping methods based on LLMs: MolCA (Liu et al., 2023b) and InstructMol (Cao et al., 2023). '-GS' indicates the use of SMILES or SELFIES as prompt in LLM inputs.

Table 4. Hyperparameter settings on classification and regression tasks.

Hyperparameter	BBBP, BACE, TOX21	HIV	QM9
GNN Hidden Dim.	300	300	300
GNN Num. Layers	5	5	5
GNN Readout	mean	mean	mean
Output Hidden Dim.	256	256	256
Output Activate Func.	relu	relu	relu
Batch Size	16	16	16
Initial LR	1e-4	1e-5	1e-4
Min LR	1e-6	1e-6	1e-5
Warm. LR	1e-7	1e-7	1e-6
LR Dec. Rate	0.9	0.9	0.9
Warm. Steps	1000	1000	1000
Optim.	adamw	adamw	adamw

C. Experiment Setup.

All experiments are conducted on NVIDIA GEFORCE GTX4090 GPU servers. The molecule datasets are divided into training, validation, and test sets according to the scaffold splitting way with a ratio of 8:1:1. As mentioned in Section 2,

the inputs to LLMs consist of a task-related instruction and a special graph token that corresponds to a molecule graph representation. For the settings of atom features and bond features in molecule graph, we follow the configuration in the baseline works (Liu et al., 2021; Xia et al., 2022; Liu et al., 2024) and only use [atom type, chirality tag] and [bond type, bond direction]. The trainable parameters of Graph2Token are within the graph tokenizer and the output layer. The graph structure encoder utilizes Graph Isomorphism Network (GIN), with the hidden layer dimension set to 300, 5 layers of graph convolutional network, and the mean function for graph pooling. The alignment function, which is the core operation of the graph tokenizer, comprises 32,001 learnable coefficients. In the output layer, a simple nonlinear multi-layer perceptron (MLP) is applied, with a hidden layer feature dimension of 256 and the ReLU function as the nonlinear activation function. For graph classification tasks, cross-entropy loss is employed as the training objective, and roc_auc score is used as the evaluation metric during testing. On the other hand, mean absolute error (MAE) is adopted as both the training target and the evaluation metric for graph regression tasks. Other hyperparameter configurations used in our experiments are shown in Tab. 4.

It can be seen that Graph2Token is a lightweight solution for LLMs to understand graph. Compared to recent molecule graph mapping approaches that have about 100M parameters (Liu et al., 2023b), or even parameter-efficient fine-tuning of LLMs models that have about 8M parameters, Graph2Token only has less than 4.2M trainable parameters.

D. Ablation Study.

In this section, we attempt to remove the alignment part in the graph tokenizer and explore the effectiveness of utilizing LLM vocabulary for mapping graph embedding into LLM semantic space. The results are shown in Figure 3. It clearly shows that when utilizing the LLM vocabulary to learn graph embeddings in the graph tokenizer, there is a significant performance improvement in graph-level tasks. Specifically, the Graph2Token approach achieves an improvement of 3.8% and 5.5% on the BBBP and BACE datasets, respectively. The results indicate that the alignment strategy proposed in this paper plays a crucial role in effectively mapping graph embeddings into the semantic space of LLM.

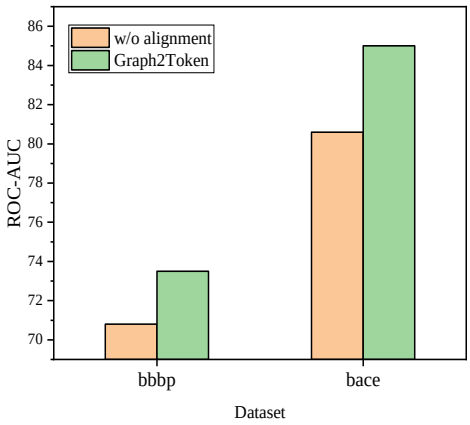


Figure 3. Performance comparison with the removal of the alignment part in the graph tokenizer on molecule graph classification tasks on BBBP and BACE datasets.