

TEXT-TO-GRAPH GENERATION WITH CONDITIONAL DIFFUSION MODELS GUIDED BY GRAPH-ALIGNED LLMs

Anonymous authors

Paper under double-blind review

ABSTRACT

Text-to-graph generation, aiming for controlled graph generation based on natural language instructions, holds significant application potentials in real-world scenarios such as drug discoveries. However, existing generative models fail to achieve text-to-graph generation in the following two aspects: i) language model-based generative models struggle with generating complex graph structures, and ii) [graph-based generative models mainly focus on unconditional graph generation or conditional generation with simple conditions](#), falling short in understanding as well as following human instructions. In this paper, we tackle the text-to-graph generation problem by employing graph diffusion models with guidance from large language models (LLMs) for the first time, to the best of our knowledge. The problem is highly non-trivial with the following challenges: 1) How to align LLMs for understanding the irregular graph structures and the graph properties hidden in human instructions, 2) How to align graph diffusion models for following natural language instructions in order to generate graphs with expected relational semantics from human. To address these challenges, we propose a novel LLM-aligned Graph Diffusion Model (**LLM-GDM**), which is able to generate graphs based on natural language instructions. In particular, we first propose the self-supervised text-graph alignment to empower LLMs with the ability to accurately understand graph structures and properties by finetuning LLMs with several specially designed alignment tasks involving various graph components such as nodes, edges, and subgraphs. Then, we propose a structure-aware cross-attention mechanism guiding the diffusion model to follow human instructions through inherently capturing the relational semantics among texts and structures. Extensive experiments on both synthetic and real-world molecular datasets demonstrate the effectiveness of our proposed **LLM-GDM** model over existing baseline methods.

1 INTRODUCTION

Graph generation is widely adopted in many real-world applications, such as molecule design (Xu et al., 2023; Gruver et al., 2023), social network analysis (Grover et al., 2019), code completion (Brockschmidt et al., 2019), neural architecture search (NAS) (Lee et al., 2021), *etc.*, yet its accessibility remains limited for users unfamiliar with graph concepts or coding skills since the models require expert knowledge to interact. Text-to-graph generation, that is to generate graphs following natural language instructions, holds significant application potentials in real-world scenarios. In molecular design, for instance, a researcher might specify a molecule as "soluble in water, stable at high temperatures, and effective against a specific enzyme." A text-to-graph generation model would interpret these instructions to create a graph, with nodes representing atoms and edges symbolizing bonds, thereby constructing a molecule that potentially fulfills all specified conditions. The development of such text-to-graph generation technologies can significantly accelerate the discovery and optimization of new molecules (Bhowmik et al., 2024; Moret et al., 2023; Flam-Shepherd et al., 2022; Hsu et al., 2022).

With the recent rise of large language models (LLMs), researchers (Edwards et al., 2022; Christofidellis et al., 2023; Fang et al., 2024) have begun exploring their application to text-to-graph generation. As illustrated in Figure 1, these models generate graphs by describing nodes and edges in

054
055
056
057
058
059
060
061
062
063
064
065
066
067
068
069
070
071
072
073
074
075
076
077
078
079
080
081
082
083
084
085
086
087
088
089
090
091
092
093
094
095
096
097
098
099
100
101
102
103
104
105
106
107

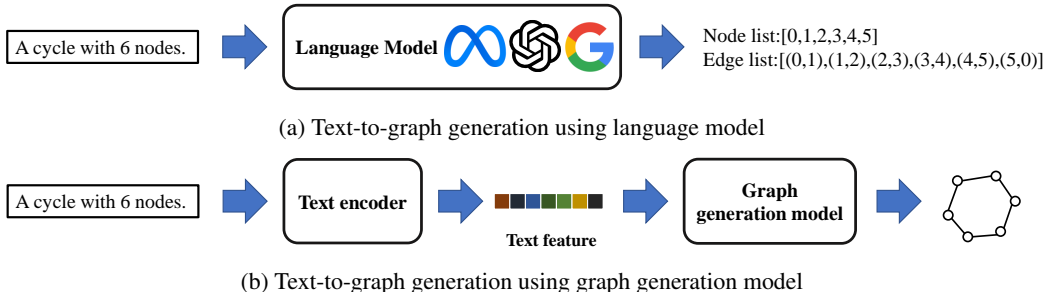


Figure 1: A comparison between text-to-graph generation method using language model and graph generation model. Language models generate graphs by describing nodes and edges in textual form. The graph generation model directly generate graphs under the guidance of the text features extracted by the text encoder.

textual form. However, while text is inherently sequential, graphs exhibit more complex topological structures that capture diverse relationships between entities. This discrepancy makes it challenging for text-only models to fully grasp and generate intricate graph structures. Consequently, we argue that integrating graph-based models is crucial to advancing the task of text-to-graph generation.

Diffusion models are a class of generative models that have garnered considerable attention recently. Notably, diffusion models have been increasingly utilized in graph generation (Vignac et al., 2023; Jo et al., 2022; Kong et al., 2023) to comprehend graph structures and generate diverse graphs through learning and sampling from a given data distribution. However, their current capabilities are limited to unconditional graph generation or conditional generation with simple conditions, where the generation distribution can not be controlled by complex conditions like human natural language instructions, textual descriptions of the graph properties, etc., limiting their applications in real-world scenarios.

In this paper, to bridge the gap, we study the problem of text-to-graph generation via guiding graph diffusion with large language models (LLMs), which remains unexplored in the literature. The problem is highly non-trivial with the following challenges:

- How to align large language models to understand the *irregular* graph structures and the *implied* graph properties in human instructions, where the instructions over graphs could be ambiguous.
- How to align graph diffusion models to follow natural language instructions to generate graphs with expected *relational* semantics, where nodes are inter-dependent with edges on graphs.

To address these challenges, we propose a novel LLM-aligned Graph Diffusion Model (LLM-GDM), which is able to generate graphs based on natural language instructions. Specifically, we first propose *self-supervised text-graph alignment* to empower LLM with better understanding of graph structures and properties by finetuning LLMs with several specially designed alignment tasks from the levels of nodes, edges and subgraphs. The finetuned LLM can extract graph-aligned features from text descriptions that capture implied graph structures in the text. Additionally, we propose a *structure-aware cross-attention mechanism* to guide the diffusion model to follow the human instructions by inherently capturing the relational semantics among texts and structures. It allows the model to generate diverse graphs that are consistent with the text input. Extensive experiments on synthetic and molecular datasets demonstrate the effectiveness of our proposed method. The contributions of this paper are summarized as follows:

- We study the problem of text-to-graph generation via guiding graph diffusion with large language models (LLMs), for the first time, to the best of our knowledge.
- We propose a novel LLM-aligned Graph Diffusion Model to generate graphs based on natural language instructions, which include two modules: i) self-supervised text-graph alignment to empower LLM with better understanding of graph structures and properties; ii) structure-aware cross-attention mechanism to capture the relational semantics among texts and structures.

- Extensive experiments on synthetic and molecular datasets demonstrate the effectiveness of our proposed method. The detailed ablation studies verify the effectiveness of each module.

2 PRELIMINARIES

2.1 GRAPH DIFFUSION MODEL

Diffusion models are a class of generative models that recently gained popularity for their excellent performance in computer vision. Recently, several works have successfully applied diffusion models to graph generation. In this paper, we focus on graph diffusion models based on stochastic differential equations (SDEs), and provide a brief description of them below.

A graph diffusion model consists of a forward process and a reverse process, both of which are defined as SDEs that operate on graph data. Consider a graph $G = (X, E)$, where X represents the node features and E represents the edge features. The forward process introduces Gaussian noise into G as the time variable t increases from $t = 0$ to $t = T$:

$$dG = f(G, t)dt + g(t)dw, \quad (1)$$

where $f(G, t)$ and $g(t)$ are predetermined functions, and dw is a standard Wiener process. We denote the value of G at time t as $G(t)$. With appropriate choices of f and g , $G(T)$ becomes indistinguishable from Gaussian-distributed values and contains nearly no information about the original graph G .

The reverse process goes backwards in time and describes the reverse of the forward SDE. It takes the following form as demonstrated in earlier works (Song et al., 2021).

$$dG = [f(G, t)dt - g(t)^2 \nabla_G \log p_t(G)]dt + g(t)dw, \quad (2)$$

where $p_t(G)$ is the marginal distribution of $G(t)$, and $\nabla_G p_t(G)$ is called its *score*.

Since the score is intractable, diffusion models use neural networks to approximate it as $s_\theta(G, t) \approx \nabla_G p_t(G)$, which can be trained using denoising score matching as follows:

$$\mathcal{L}_{\text{score}} = \mathbb{E}_t \mathbb{E}_{G(0)} \mathbb{E}_{G(t)|G(0)} [\lambda(t) \|s_\theta(G(t), t) - \nabla_{G(t)} \log p_{0t}(G(t) | G(0))\|^2], \quad (3)$$

where $p_{0t}(G(t) | G(0))$ is the transition probability of the forward process, and $\lambda(t)$ is a weighting function. After training, new graphs can be generated with graph diffusion models by starting with $G(T)$ sampled from Gaussian distributions and solving the reverse process using methods like Euler-Maruyama or Predictor-Corrector methods (Song et al., 2021).

2.2 CLASSIFIER-FREE DIFFUSION GUIDANCE

Classifier-free guidance (Ho & Salimans, 2021) is a widely used technique for conditional generation using diffusion models. It augments the score prediction network in diffusion models with the conditional information as an additional input. Let c be the condition, the conditional score network $s_\theta(G, t, c)$ is trained to approximate the conditional score $\nabla_G p_t(G | c)$.

To improve the quality of conditional generation, classifier-free guidance introduces a scale factor that controls the influence of the condition. Let k be the scale factor, classifier-free guidance modifies the estimated score function in the reverse process by scaling the difference between the predicted conditional and unconditional scores:

$$s_{\text{cfg}}(G, t, c) = s_\theta(G, t) + k(s_\theta(G, t, c) - s_\theta(G, t)). \quad (4)$$

In practice, the scale factor is large than 1, so the influence of conditional data is amplified. Data generated with higher scale factors tend to be more consistent with the provided condition, while lower scale factors can result in increased generation diversity.

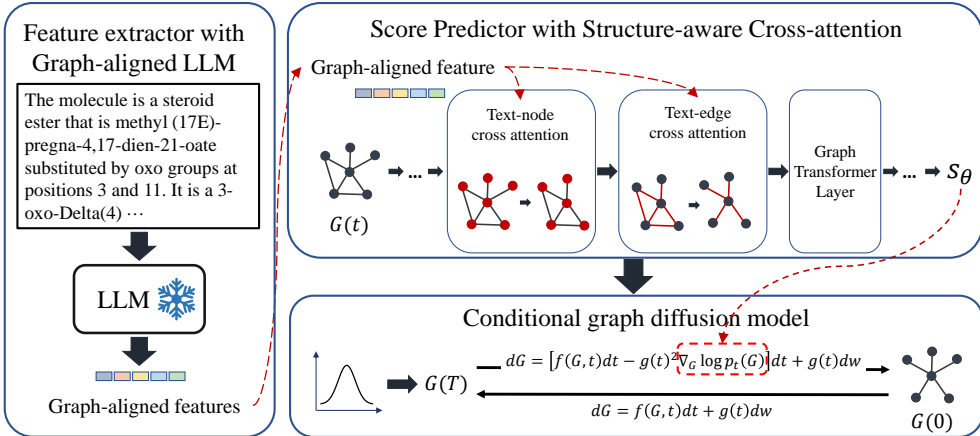
3 METHOD

In this section, we present our text-to-graph generation method. We first describe the overall framework of the proposed method, and then introduce the self-supervised text-graph alignment task for finetuning LLMs and the structure-aware cross-attention mechanism for incorporating text features into graph diffusion models.

162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215



Stage 1: Finetune LLM using self-supervised text-graph alignment task



Stage 2: Use graph-aligned features to train conditional graph diffusion model

Figure 2: An overview of our method. We use a condition graph diffusion model to perform text-to-graph generation. In the first stage, as shown in the upper part of the figure, we use *self-supervised text-graph alignment* task to finetune LLM to extract graph-aligned features from the text description. In the second stage, we use the graph-aligned LLM to extract graph-aligned features from the text description, and apply them to the score predictor in the conditional graph diffusion model using *structure-aware cross-attention mechanism*.

3.1 FRAMEWORK

We use a diffusion model for graph generation in our method. We represent a graph G by its node types $X = \{x_i\}$ and adjacency matrix $E = \{e_{ij}\}$. Here, $x_i \in \{1, 2, \dots, C_{\text{node}}\}$ is the type of the i -th node in G , and $e_{ij} \in \{0, 1, 2, \dots, C_{\text{edge}}\}$ is the type of edge between nodes i and j , with $e_{ij} = 0$ indicating no edge. Our method aims to generate corresponding graph data G for a given text description T by learning the conditional distribution $p_{\theta}(G|T)$.

The framework of our method is illustrated in Figure 2. In the first stage of our method, we use the self-supervised text-to-graph alignment task to finetune the Llama-3-8B model, obtaining a graph-aligned LLM. In the second stage, we construct a conditional graph diffusion model to generate graphs according to text description, and use a graph transformer (Yun et al., 2019) with structure-aware cross-attention as its conditional score predictor.

For a given text description T , the conditional score predictor uses a feature extractor with a graph-aligned LLM to extract text features from T , as introduced in Section 3.2. Then, the score predictor uses structure-aware cross-attention to modify the prediction results based on the extracted text features, as detailed in Section 3.3. During the generation process, we use classifier-free guidance to generate graphs according to text description T .

3.2 SELF-SUPERVISED TEXT-GRAPH ALIGNMENT

To incorporate text descriptions into the generation process, we need to extract features from the text. Pretrained language models are known to produce high-quality text features for downstream tasks. With billions of parameters, LLMs demonstrate strong performance in various graph tasks (Li et al., 2023), indicating their ability to understand the graph structure contained in text. This makes it feasible to use LLMs to extract text features relevant for graph tasks.

To ensure that the text features extracted by LLMs are more focused on the graph generation task, we finetune LLMs to obtain text features that are more relevant to the graph structure. For this purpose, we design a graph structure prediction task to finetune LLMs. Specifically, we aim for the extracted text features to reflect the structure of the graphs corresponding to the text description, including information about nodes, edge, and subgraphs. Therefore, the graph structure prediction task includes the prediction of three categories of targets: the number of nodes, edges, and subgraphs of various types in the graph, denoted by $c_{\text{node},i}$, $c_{\text{edge},i}$ and $c_{\text{sub},i}$ respectively. They are defined as follows:

$$\begin{aligned} c_{\text{node},i} &= |\{j \mid x_j = i\}|, \\ c_{\text{edge},i} &= |\{(j, k) \mid e_{jk} = i\}|, \\ c_{\text{sub},i} &= |\{G' \mid G' \text{ is a subgraph of } G \text{ and } G' \text{ is isomorphic to } G_i\}|. \end{aligned} \quad (5)$$

The objective function for finetuning is:

$$\mathcal{L}_{\text{align}} = \sum_i (c_{\text{node},i}^* - c_{\text{node},i})^2 + \sum_i (c_{\text{edge},i}^* - c_{\text{edge},i})^2 + \sum_i (c_{\text{sub},i}^* - c_{\text{sub},i})^2, \quad (6)$$

where $c_{\text{node},i}^*$, $c_{\text{edge},i}^*$, $c_{\text{sub},i}^*$ are the value predicted by LLM for the number of nodes, edges, subgraphs of various types in the graph, respectively.

It is worth mentioning that in the actual generation process, for each piece of text description, we input it into the finetuned LLM and use the output of the last hidden layer as the extracted features. The result will have features for each token in the input text.

3.3 STRUCTURE-AWARE CROSS-ATTENTION MECHANISM

To allow text-to-graph generation using diffusion models, it is necessary to incorporating text features into the conditional score predictor. While there are methods like affine conditioning or cross-attention for constructing the score predictors of conditional diffusion models, these approaches do not account for the complex nature of graph data. Directly applying them to nodes and edges in graph diffusion models can lead to suboptimal results. Since the edges in graphs represent the relationship between nodes, an effective conditioning method for graphs should respect this property and ensure the edge conditioning accounts for the nodes. Additionally, the size of the adjacency matrix is quadratic in the number of nodes. For more computationally expensive conditioning methods like cross-attention, calculating edge conditioning for each edge independently is costly.

In this section, we propose a structure-aware cross-attention method for graph diffusion models, which integrates a sequence of conditional text features into the score predictor network by computing attention between the graph data and the text features.

Specifically, we use the structure-aware cross-attention mechanism in the score predictor module of the graph diffusion model, where the results of node attention and edge attention are added into the network using residual connections. Let X and E be the node and edge features in some layer of the score predictor, and C be the sequence of text features. Structure-aware cross-attention first computes the cross-attention between node features and text features as follows, and use the attention results for node conditioning:

$$Q = XW_Q, \quad K = CW_K, \quad V = CW_V, \quad (7)$$

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right), \quad (8)$$

$$X_{\text{cond}} = AV, \quad (9)$$

where Q , K , and V are weight matrices for queries, keys, and values, d is the dimensionality of keys, A is the attention score for nodes, and X_{cond} is the node conditioning result.

Then, structure-aware cross-attention computes the attention scores for edges based on the node attention results. For each edge (u, v) , its attention score should be related to the attention scores of node u and v . We compute two scalar values $G_{1,uv}$ using a gating mechanism for each edge based on its features E_{uv} , which represents the influence of two endpoints u and v in the edge:

$$G_{1,uv} = \sigma(E_{uv}^T W_{G1}), \quad G_{2,uv} = 1 - G_{1,uv}, \quad (10)$$

where W_{G_1} is trainable weights, and σ is the logistic sigmoid function. The attention score of edge (u, v) is computed as a weighted mixture of the attention scores for node u and v :

$$A_{\text{edge},uv} = G_{1,uv}A_u + G_{2,uv}A_v, \quad (11)$$

where $A_{\text{edge},uv}$ is the attention score for edge (u, v) , and A_u and A_v are attention scores for node u and v respectively.

Finally, the edge conditioning result is determined by mixing the attention values according to the edge attention scores:

$$E_{\text{cond},uv} = A_{\text{edge},uv}V. \quad (12)$$

By deriving the edge conditioning from the node conditioning, structure-aware cross-attention can utilize the text features in the conditional score predictor with relatively low computational costs, and captures the relational semantics among texts and structures more efficiently. In our experiments, we apply structure-aware cross-attention in each layer of the score predictor.

4 EXPERIMENT

To demonstrate the effectiveness of our method, we constructed three parts of experiments. In the first part, we compare language model-based and graph-based methods on a synthetic graph dataset, which includes graphs with multiple types of structures. In the second part, we compare various methods on the task of text-conditional molecular graph generation using a real-world molecular dataset. In the third part, we conduct extensive ablation experiments to explore the roles of the two modules we proposed and the impact of hyper-parameter settings on the performance of text-to-graph generation.

4.1 GRAPH GENERATION ON SYNTHETIC DATASET

We conducted experiments on synthetic dataset, using the model to generate corresponding graphs according to the rules described in the text. We explored the ability of LLMs to understand graph structures through rules of different difficulty levels, as well as the ability of our method to understand graph structures described in the text.

Datasets We construct a synthetic dataset with 7 kinds of graphs, Tree, Cycle, Wheel, Bipartite, K-regular, Component, and Mix. The details of these kinds of graphs are available in the appendix. Each graph is accompanied by its corresponding text description, which specifies its kind and important properties, like ‘‘A tree with 8 nodes’’ or ‘‘A graph with 10 nodes and 2 connected components’’. There are 10000 graphs in the training set and 1000 graphs in the test set. During testing, we ask the model to generate a new graph that matches the text description.

Baselines We compare our method with several LLMs, including Qwen2.5-7B, Qwen2.5-72B, Gemma-2-9B, Gemma-2-27B, Llama-3.1-8B, and Llama-3.1-70B. We use the LLMs in a zero-shot way, asking the model to generate the text representation of graphs according to text descriptions with a prompt.

Metrics For each kind of graph in the test set, we report the proportion of generated graphs that matches the given text descriptions. It is worth noting that there are generally more than one graph matching a piece of text description, and any matching graph will count towards the metric.

Results and analysis The experimental results are shown in Table 1. We can find that: 1) Overall, LLMs with a larger number of parameters perform better than those with a smaller number of parameters. Moreover, LLMs other than Llama-3.1-8B perform well for generating graphs with simple structures such as trees and cycles, but perform poorly on tasks with more complex graph structures. This indicates that **LLMs struggle with generating complex graph structures**. 2) In addition, our method performs well on various tasks, indicating that **graph-based methods can better generate graphs with complex structures**.

Table 1: The result of graph generation on the synthetic dataset. The values in the table are the proportions of generated graphs that matches the given text descriptions.

Model	Tree	Cycle	Wheel	Bipartite	K-regular	Component	Mix
Qwen2.5-7B	0.778 \pm 0.010	1.000 \pm 0.000	0.020 \pm 0.016	0.088 \pm 0.038	0.295 \pm 0.046	0.178 \pm 0.078	0.178 \pm 0.019
Qwen2.5-72B	1.000 \pm 0.000	1.000 \pm 0.000	0.259 \pm 0.061	0.371 \pm 0.047	0.767 \pm 0.017	0.448 \pm 0.060	0.319 \pm 0.021
Gemma-2-9B	0.942 \pm 0.018	1.000 \pm 0.000	0.000 \pm 0.000	0.040 \pm 0.014	0.286 \pm 0.024	0.146 \pm 0.005	0.235 \pm 0.025
Gemma-2-27B	1.000 \pm 0.000	1.000 \pm 0.000	0.000 \pm 0.000	0.383 \pm 0.037	0.020 \pm 0.017	0.098 \pm 0.006	0.231 \pm 0.048
Llama-3.1-8B	0.178 \pm 0.037	0.136 \pm 0.042	0.000 \pm 0.000	0.000 \pm 0.000	0.007 \pm 0.010	0.007 \pm 0.010	0.121 \pm 0.006
Llama-3.1-70B	1.000 \pm 0.000	1.000 \pm 0.000	0.597 \pm 0.069	0.154 \pm 0.006	0.353 \pm 0.037	0.450 \pm 0.026	0.347 \pm 0.067
Ours	0.992 \pm 0.012	0.636 \pm 0.018	0.669 \pm 0.029	0.916 \pm 0.002	1.000 \pm 0.000	0.962 \pm 0.018	0.589 \pm 0.068

4.2 TEXT CONDITIONAL MOLECULAR GRAPH GENERATION

We conducted experiments on the text to molecule dataset and demonstrated the effectiveness of our method in generating molecular graphs based on text descriptions through comparison with the text to molecule text model.

Datasets Generating molecules according to text is an emerging task, and currently the only widely used and public dataset in this task is the ChEBI-20 dataset (Edwards et al., 2021). This dataset contains 33010 molecules and their corresponding text descriptions. We follow the approach of MolT5 (Edwards et al., 2022) for dataset splits and preprocessing.

Baselines We compare with the following methods.

- RNN: A 4-layer GRU recurrent neural network with bidirectional encoder trained on the ChEBI-20 dataset.
- Transformer: A vanilla transformer model consisting of six encoder and decoder layers trained on the ChEBI-20 dataset.
- MolT5: An encoder-decoder Transformer model initialized with a public checkpoint of T5, then pretrained on the combined dataset of C4 and ZINC, finally finetuned on the ChEBI-20 dataset.
- Llama-3.1: An decoder-only Transformer model published by Meta. Llama-3.1-8B has 8B parameters, while Llama-3.1-70B has 70B parameters. We use the model in a zero-shot way, asking the model to generate the SMILES representation of molecules according to text descriptions with a prompt.

Metrics We use the following metrics to evaluate our method and compare it with the baseline.

- Fingerprint-based molecule similarity metrics: We measure the fingerprint Tanimoto similarity (FTS) between each generated molecule and the corresponding ground truth molecule. MACCS, RDK, and Morgan represent the three different extraction methods for molecular fingerprints. We consider these to be the primary metrics measuring the quality of generated graphs in our experiments.
- FCD: We measure the Fréchet ChemNet Distance (FCD) (Preuer et al., 2018) between the generated molecules and ground truth molecules. It reflects the distance of chemical and biological information between the two sets of molecules.
- Exact: We measure the proportion of generated molecules that are exact matches of their corresponding ground truth molecules.
- Validity: We report the validity of the generated molecules as measured by RDKit sanitization.
- Diversity: We measure the diversity of generated molecules, defined as the average pairwise differences between multiple molecules generated by the model guided by the same text description. The pairwise differences are calculated by subtracting their FTS from 1. In the experiments, we select the first 100 text descriptions in the test set and generate 10 graphs for each description to calculate the diversity metrics.

Results and analysis From Table 2, we can observe that: 1) our method has similar performance to MolT5 in terms of FTS metrics, outperforming MolT5 in MACCS and RDK FTS, and has much smaller trainable parameter sizes. 2) Our method generates relatively few exact molecule matches,

Table 2: The result of molecule generation guided by the text in the test split of ChEBI-20. “MACCS”, “RDKit”, and “Morgan” are fingerprint Tanimoto similarity metrics. The result of RNN, Transformer, and MolT5 are sourced from Edwards et al. (2022), and other methods maintained the same settings as Edwards et al. (2022) during testing. “Param.” denote the number of trainable parameters of the model. The number of parameters for RNN and Transformer are unknown.

Model	MACCS \uparrow	RDKit \uparrow	Morgan \uparrow	FCD \downarrow	Exact \uparrow	Validity \uparrow	Param.
RNN	0.591	0.400	0.362	4.55	0.005	0.542	—
Transformer	0.480	0.320	0.217	11.32	0.000	0.906	—
MolT5	0.721	0.588	0.529	2.18	0.081	0.772	220M
Llama-3.1-8B	0.545 \pm 0.004	0.305 \pm 0.004	0.238 \pm 0.002	5.86 \pm 0.15	0.007 \pm 0.001	0.370 \pm 0.005	—
Llama-3.1-70B	0.683 \pm 0.003	0.450 \pm 0.004	0.390 \pm 0.004	3.27 \pm 0.10	0.049 \pm 0.003	0.563 \pm 0.008	—
Ours	0.787 \pm 0.001	0.638 \pm 0.002	0.470 \pm 0.002	2.96 \pm 0.06	0.050 \pm 0.002	1.000 \pm 0.000	16.5M

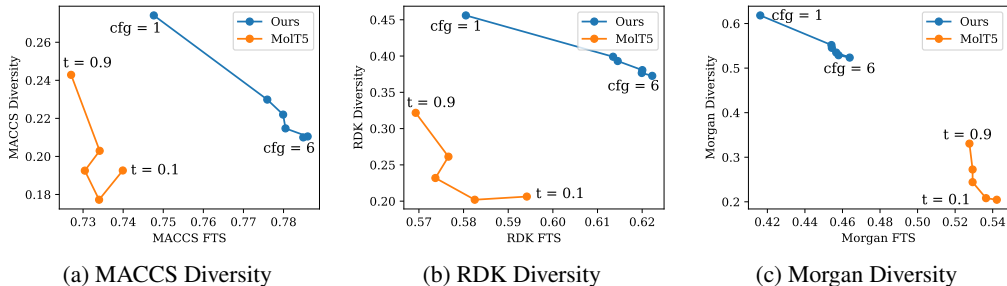


Figure 3: The diversity results of our method and MolT5. The horizontal axis represents the FTS between the generated and real molecules, the vertical axis represents the diversity between the generated molecules. The orange dots represent the values measured by MolT5 when setting different temperature values, and the blue dots represent the values measured by our method when setting different classifier-free guidance scale values.

because the generated graphs of diffusion models are diverse and it is difficult to ensure full matches. 3) At the same time, we can also observe that zero-shot Llama-3.1 performs poorly on various metrics, indicating that directly using LLMs for molecular generation tasks does not result in good performances.

In addition, we report the diversity results of our method and MolT5. Since diversity can be affected by the choice of parameters like the temperature value for MolT5 and the classifier-free guidance scale for our method, we test the models using several configurations and report their FTS and diversity metrics. The results can be seen in Figure 3. It can be seen that 1) the FTS metrics generally increases when the temperature is reduced (for MolT5) or the CFG scale is increased (for our method), while the diversity metrics generally decreases. 2) Overall, our method has higher diversity scores than MolT5. When the two methods have similar FTS metrics, our method achieves better diversity. While our method has lower performance than MolT5 in the Morgan FTS metric, the diversity of our method is significantly higher. The result demonstrates that our proposed method can generate graphs with higher diversity for the task of text-to-graph generation.

4.3 ABLATION STUDY

We conducted ablation experiments to explore the effects of different settings. All ablation experiments are performed on the ChEBI-20 dataset. The metrics are the same as the molecule generation experiments in Section 4.2.

4.3.1 THE EFFECT OF DIFFERENT TEXT CONDITIONING METHODS

We compare our structure-aware cross-attention mechanism with other text conditioning methods for diffusion models.

Compared methods We compare the following methods:

Table 3: Molecule generation results of different text conditioning methods.

Model	MACCS \uparrow	RDK \uparrow	Morgan \uparrow	FCD \downarrow	Exact \uparrow
Affine	0.734	0.570	0.391	3.67	0.028
Cross-attention	0.770	0.615	0.440	3.28	0.040
Ours	0.789	0.639	0.473	2.98	0.048

Table 4: Molecule generation results of LLM finetuning.

Model	MACCS \uparrow	RDK \uparrow	Morgan \uparrow	FCD \downarrow	Exact \uparrow
No LLM finetune	0.728	0.574	0.385	4.23	0.031
LLM finetune	0.789	0.639	0.473	2.98	0.048

- Affine: The features of the last token in the text description are inserted into the score predictor using feature-wise affine transformations, also known as feature-wise linear modulation (FiLM).
- Cross-attention: The text features are inserted into the score predictor using cross-attention between node features and text features. The edge features are not modified directly.
- Ours: The text features are inserted into the score predictor using our proposed structure-aware cross-attention mechanism.

Results and analysis The experimental results are shown in Table 3. We can find that structure-aware cross-attention achieves the best performance among compared methods in terms of molecular similarity to the ground truth. This indicates that our proposed method can better incorporate text features into graph diffusion models.

4.3.2 THE EFFECT OF LLM FINETUNING

Compared methods We compare with the following experimental settings:

- No LLM finetune: The model uses pretrained Llama-3-8B directly to extract features from text.
- LLM finetune: The model uses our finetuned LLM to extract graph-aligned features from text.

Results and analysis The experimental results are shown in Table 4. We can find that using the finetuned LLM for feature extraction is crucial for improving the quality of graph generation. Although the validity has slightly decreased, other metrics have significantly improved with the finetuned LLM.

4.3.3 THE EFFECT OF CLASSIFIER-FREE GUIDANCE SCALE

It is known that the classifier-free guidance scale has a significant impact on the performance of text-to-image diffusion models. In this section, we explore the effect of different guidance scales on our text-to-graph model.

Results and analysis The experimental results are shown in Table 5. It can be seen that: 1) The validity of generated molecules decreases as the guidance scale increases, indicating that strong

Table 5: Molecule generation results using different classifier-free guidance scales.

Model	MACCS \uparrow	RDK \uparrow	Morgan \uparrow	FCD \downarrow	Exact \uparrow
CFG scale = 1	0.752	0.584	0.411	3.355	0.037
CFG scale = 3	0.787	0.635	0.465	2.98	0.048
CFG scale = 5	0.789	0.639	0.473	2.98	0.048
CFG scale = 7	0.575	0.432	0.297	3.005	0.026
CFG scale = 9	0.336	0.210	0.111	2.97	0.002

486 guidance from the text can lead to generating inconsistent graph structures. 2) The influence of
487 guidance scale on molecular similarity metrics are minimal, indicating that our method is robust
488 to the choice of guidance scale. Generally, choosing a guidance scale around 5 leads to the best
489 performance.

491 5 RELATED WORK

494 5.1 DIFFUSION-BASED GRAPH GENERATION

495 Diffusion model has achieved great success in the field of computer vision. Recently, some re-
496 searchers have used diffusion models to solve graph generation task. For example, EDP-GNN (Niu
497 et al., 2020) is the first work using Score Matching with Langevin Dynamics (SMLD) (Song & Er-
498 mon, 2019) diffusion model to generate graphs, which learns the score function of the adjacency
499 matrices distributions of the graphs. GDSS (Jo et al., 2022) proposes a graph generation method
500 using continuous-time diffusion models (Song et al., 2021), which models the joint distribution of
501 the nodes and edges through stochastic differential equations (SDEs). DiGress (Vignac et al., 2023)
502 uses a diffusion model over discrete data space for graph generation, and additionally preserves the
503 marginal distribution of node and edge types and incorporates auxiliary graph-theoretic features.
504 These methods have demonstrated excellent performance on the task of graph generation.

505 In order to generate graphs that match specific requirements, conditional graph generation has re-
506 ceived attention in recent years. For example, DiGress (Vignac et al., 2023) uses classifier guidance
507 to perform graph generation guided by several graph-level properties, like the dipole moment and
508 highest occupied molecular orbit of molecular graphs. However, existing methods have not ex-
509 plored the task of text-guided graph generation, which is necessary for the popularization of graph
510 generation methods in various fields.

512 5.2 LARGE LANGUAGE MODELS FOR GRAPHS

514 LLMs have achieved good results on various natural language process tasks. Recently, many works
515 explored the application of LLMs in graph tasks. For example, Chen et al. (2023) uses LLMs to
516 predict node categories on text attribute graphs. Wang et al. (2023) proposes a benchmark framework
517 to evaluate the performance of LLMs with several graph algorithmic tasks, including topological
518 sort, maximum flows, *etc.* Zhang et al. (2023) evaluates the abilities of LLMs to handle spatial-
519 temporal information on dynamic graphs. Yao et al. (2024) designs a series of tasks to evaluate
520 the graph generation ability of LLMs. These and other studies have demonstrated the potential of
521 LLMs for processing graph tasks, showing the possibility to extract graph structure features from
522 text description using LLMs.

523 [Christofidellis et al. \(2023\)](#) and [Fang et al. \(2024\)](#) are text-to-molecule methods that leverage lan-
524 guage models for generation, where molecules are represented in the SMILES format. [Gong et al.](#)
525 [\(2024\)](#) employs a text diffusion model to generate the SMILES representation of molecules. While
526 these methods focus on generating molecules from text representations, they do not extend to other
527 graph generation tasks. [Zhu et al. \(2024\)](#) utilizes a latent space diffusion model to generate latent
528 features, which are then decoded into molecule graphs using a graph decoder. In contrast, this pa-
529 per explores for the first time the use of graph-aligned LLMs to extract text features for guiding
530 conditional graph diffusion models.

532 6 CONCLUSION

534 This paper addresses the critical gap in text-to-graph generation by proposing the LLM-aligned
535 Graph Diffusion Model (LLM-GDM), which integrates large language models with graph diffusion
536 model to generate graphs from natural language instructions. By developing a self-supervised text-
537 graph alignment process and introducing a structure-aware cross-attention mechanism, our approach
538 enhances the model’s understanding of graph structures and properties and ensures that generated
539 graphs adhere to the specified relational semantics in the text. Extensive experimental results on
synthetic and molecular datasets confirm the efficiency of our method.

REFERENCES

- 540
541
542 Debsindhu Bhowmik, Pei Zhang, Zachary Fox, Stephan Irle, and John Gounley. Enhancing molecu-
543 lar design efficiency: Uniting language models and generative networks with genetic algorithms.
544 *Patterns*, 5(4), 2024.
- 545
546 Marc Brockschmidt, Miltiadis Allamanis, Alexander L. Gaunt, and Oleksandr Polozov. Gen-
547 erative code modeling with graphs. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL
548 <https://openreview.net/forum?id=Bke4KsA5FX>.
- 549
550 Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei
551 Yin, Wenqi Fan, Hui Liu, and Jiliang Tang. Exploring the potential of large language mod-
552 els (llms) in learning on graphs. *SIGKDD Explor.*, 25(2):42–61, 2023. doi: 10.1145/3655103.
553 3655110. URL <https://doi.org/10.1145/3655103.3655110>.
- 554
555 Dimitrios Christofidellis, Giorgio Giannone, Jannis Born, Ole Winther, Teodoro Laino, and Mat-
556 teo Manica. Unifying molecular and textual representations via multi-task language modelling.
557 In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and
558 Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29*
559 *July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Re-*
560 *search*, pp. 6140–6157. PMLR, 2023. URL [https://proceedings.mlr.press/v202/](https://proceedings.mlr.press/v202/christofidellis23a.html)
561 [christofidellis23a.html](https://proceedings.mlr.press/v202/christofidellis23a.html).
- 562
563 Carl Edwards, ChengXiang Zhai, and Heng Ji. Text2mol: Cross-modal molecule retrieval with nat-
564 ural language queries. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-
565 tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language*
566 *Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November,*
567 *2021*, pp. 595–607. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.
EMNLP-MAIN.47. URL <https://doi.org/10.18653/v1/2021.emnlp-main.47>.
- 568
569 Carl Edwards, Tuan Manh Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. Translation
570 between molecules and natural language. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang
571 (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Pro-*
572 *cessing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 375–413.
573 Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.26.
574 URL <https://doi.org/10.18653/v1/2022.emnlp-main.26>.
- 575
576 Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and
577 Huajun Chen. Mol-instructions: A large-scale biomolecular instruction dataset for large language
578 models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vi-*
579 *enna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL [https://openreview.net/](https://openreview.net/forum?id=Tlstdsb6l9n)
[forum?id=Tlstdsb6l9n](https://openreview.net/forum?id=Tlstdsb6l9n).
- 580
581 Daniel Flam-Shepherd, Kevin Zhu, and Alán Aspuru-Guzik. Language models can learn complex
582 molecular distributions. *Nature Communications*, 13(1):3293, 2022.
- 583
584 Haisong Gong, Qiang Liu, Shu Wu, and Liang Wang. Text-guided molecule generation with
585 diffusion language model. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan
586 (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth*
587 *Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Sym-*
588 *posium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024,*
589 *Vancouver, Canada*, pp. 109–117. AAAI Press, 2024. doi: 10.1609/AAAI.V38I1.27761. URL
<https://doi.org/10.1609/aaai.v38i1.27761>.
- 590
591 Aditya Grover, Aaron Zweig, and Stefano Ermon. Graphite: Iterative generative modeling of graphs.
592 In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International*
593 *Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*,
volume 97 of *Proceedings of Machine Learning Research*, pp. 2434–2444. PMLR, 2019. URL
<http://proceedings.mlr.press/v97/grover19a.html>.

- 594 Nate Gruver, Samuel Stanton, Nathan C. Frey, Tim G. J. Rudner, Isidro Hötzel, Julien
595 Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew Gordon Wilson. Pro-
596 tein design with guided discrete diffusion. In Alice Oh, Tristan Naumann, Amir
597 Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neu-
598 ral Information Processing Systems 36: Annual Conference on Neural Information Pro-
599 cessing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16,
600 2023, 2023*. URL [http://papers.nips.cc/paper_files/paper/2023/hash/
601 29591f355702c3f4436991335784b503-Abstract-Conference.html](http://papers.nips.cc/paper_files/paper/2023/hash/29591f355702c3f4436991335784b503-Abstract-Conference.html).
- 602 Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on
603 Deep Generative Models and Downstream Applications*, 2021.
- 604
605 Yu-Chuan Hsu, Zhenze Yang, and Markus J Buehler. Generative design, manufacturing, and molec-
606 ular modeling of 3d architected materials based on natural language input. *APL Materials*, 10(4),
607 2022.
- 608
609 Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the sys-
610 tem of stochastic differential equations. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba
611 Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning,
612 ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine
613 Learning Research*, pp. 10362–10383. PMLR, 2022. URL [https://proceedings.mlr.
614 press/v162/jo22a.html](https://proceedings.mlr.press/v162/jo22a.html).
- 615
616 Lingkai Kong, Jiaming Cui, Haotian Sun, Yuchen Zhuang, B. Aditya Prakash, and Chao Zhang.
617 Autoregressive diffusion model for graph generation. In Andreas Krause, Emma Brunskill,
618 Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International
619 Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, vol-
620 ume 202 of *Proceedings of Machine Learning Research*, pp. 17391–17408. PMLR, 2023. URL
621 <https://proceedings.mlr.press/v202/kong23b.html>.
- 622
623 Hayeon Lee, Eunyoung Hyung, and Sung Ju Hwang. Rapid neural architecture search by learning
624 to generate graphs from datasets. In *9th International Conference on Learning Representations,
625 ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL [https://
626 openreview.net/forum?id=rkQuFUUmUOg3](https://openreview.net/forum?id=rkQuFUUmUOg3).
- 627
628 Yuhan Li, Zhixun Li, Peisong Wang, Jia Li, Xiangguo Sun, Hong Cheng, and Jeffrey Xu Yu. A sur-
629 vey of graph meets large language model: Progress and future directions. *CoRR*, abs/2311.12399,
630 2023. doi: 10.48550/ARXIV.2311.12399. URL [https://doi.org/10.48550/arXiv.
631 2311.12399](https://doi.org/10.48550/arXiv.2311.12399).
- 632
633 Michael Moret, Irene Pachon Angona, Leandro Cotos, Shen Yan, Kenneth Atz, Cyrill Brunner,
634 Martin Baumgartner, Francesca Grisoni, and Gisbert Schneider. Leveraging molecular structure
635 and bioactivity with chemical language models for de novo drug design. *Nature Communications*,
636 14(1):114, 2023.
- 637
638 Chenhao Niu, Yang Song, Jiaming Song, Shengjia Zhao, Aditya Grover, and Stefano Ermon.
639 Permutation invariant graph generation via score-based generative modeling. In Silvia Chi-
640 appa and Roberto Calandra (eds.), *The 23rd International Conference on Artificial Intelligence
641 and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108
642 of *Proceedings of Machine Learning Research*, pp. 4474–4484. PMLR, 2020. URL [http:
643 //proceedings.mlr.press/v108/niu20a.html](http://proceedings.mlr.press/v108/niu20a.html).
- 644
645 Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Günter Klambauer.
646 Fréchet chemnet distance: A metric for generative models for molecules in drug discovery. *J.
647 Chem. Inf. Model.*, 58(9):1736–1741, 2018. doi: 10.1021/ACS.JCIM.8B00234. URL [https:
648 //doi.org/10.1021/acs.jcim.8b00234](https://doi.org/10.1021/acs.jcim.8b00234).
- 649
650 Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the
651 data distribution. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence

- 648 d’Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Informa-*
649 *tion Processing Systems 32: Annual Conference on Neural Information Processing Sys-*
650 *tems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 11895–
651 11907, 2019. URL [https://proceedings.neurips.cc/paper/2019/hash/](https://proceedings.neurips.cc/paper/2019/hash/3001ef257407d5a371a96dcd947c7d93-Abstract.html)
652 [3001ef257407d5a371a96dcd947c7d93-Abstract.html](https://proceedings.neurips.cc/paper/2019/hash/3001ef257407d5a371a96dcd947c7d93-Abstract.html).
- 653 Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and
654 Ben Poole. Score-based generative modeling through stochastic differential equations. In
655 *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria,*
656 *May 3-7, 2021*. OpenReview.net, 2021. URL [https://openreview.net/forum?id=](https://openreview.net/forum?id=PXTIG12RRHS)
657 [PXTIG12RRHS](https://openreview.net/forum?id=PXTIG12RRHS).
- 658 Clément Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal
659 Frossard. Digress: Discrete denoising diffusion for graph generation. In *The Eleventh Inter-*
660 *national Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.
661 OpenReview.net, 2023. URL <https://openreview.net/pdf?id=UaAD-Nu86WX>.
- 662 Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov.
663 Can language models solve graph problems in natural language? In Alice Oh, Tris-
664 tan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Ad-*
665 *vances in Neural Information Processing Systems 36: Annual Conference on Neural Infor-*
666 *mation Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16,*
667 *2023, 2023*. URL [http://papers.nips.cc/paper_files/paper/2023/hash/](http://papers.nips.cc/paper_files/paper/2023/hash/622afc4edf2824a1b6aaf5afe153fa93-Abstract-Conference.html)
668 [622afc4edf2824a1b6aaf5afe153fa93-Abstract-Conference.html](http://papers.nips.cc/paper_files/paper/2023/hash/622afc4edf2824a1b6aaf5afe153fa93-Abstract-Conference.html).
- 669 Minkai Xu, Alexander S. Powers, Ron O. Dror, Stefano Ermon, and Jure Leskovec. Geometric latent
670 diffusion models for 3d molecule generation. In Andreas Krause, Emma Brunskill, Kyunghyun
671 Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference*
672 *on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of
673 *Proceedings of Machine Learning Research*, pp. 38592–38610. PMLR, 2023. URL [https:](https://proceedings.mlr.press/v202/xu23n.html)
674 [/proceedings.mlr.press/v202/xu23n.html](https://proceedings.mlr.press/v202/xu23n.html).
- 675 Yang Yao, Xin Wang, Zeyang Zhang, Yijian Qin, Ziwei Zhang, Xu Chu, Yuekui Yang, Wenwu Zhu,
676 and Hong Mei. Exploring the potential of large language models in graph generation. *CoRR*,
677 [abs/2403.14358](https://arxiv.org/abs/2403.14358), 2024. doi: 10.48550/ARXIV.2403.14358. URL [https://doi.org/10.](https://doi.org/10.48550/arXiv.2403.14358)
678 [48550/arXiv.2403.14358](https://doi.org/10.48550/arXiv.2403.14358).
- 679 Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J. Kim. Graph
680 transformer networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Flo-
681 rence d’Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Infor-*
682 *mation Processing Systems 32: Annual Conference on Neural Information Processing Sys-*
683 *tems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 11960–
684 11970, 2019. URL [https://proceedings.neurips.cc/paper/2019/hash/](https://proceedings.neurips.cc/paper/2019/hash/9d63484abb477c97640154d40595a3bb-Abstract.html)
685 [9d63484abb477c97640154d40595a3bb-Abstract.html](https://proceedings.neurips.cc/paper/2019/hash/9d63484abb477c97640154d40595a3bb-Abstract.html).
- 686 Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, Yijian Qin, Simin Wu, and Wenwu
687 Zhu. Llm4dyg: Can large language models solve problems on dynamic graphs? *CoRR*,
688 [abs/2310.17110](https://arxiv.org/abs/2310.17110), 2023. doi: 10.48550/ARXIV.2310.17110. URL [https://doi.org/10.](https://doi.org/10.48550/arXiv.2310.17110)
689 [48550/arXiv.2310.17110](https://doi.org/10.48550/arXiv.2310.17110).
- 690 Huaisheng Zhu, Teng Xiao, and Vasant G Honavar. 3m-diffusion: Latent multi-modal diffusion
691 for language-guided molecular structure generation. In *First Conference on Language Modeling*,
692 2024.
- 693
694
695
696
697
698
699
700
701

A EXPERIMENTAL SETTING

Our code is based on GDSS-Transformer (<https://github.com/DongkiKim95/GDSS-Transformer/>). We modified the code to allow text-conditional graph generation. We initialized the parameters of the base architecture with the checkpoint of GDSS-Transformers’s model trained on ZINC250k, and then trained our structure-aware cross-attention along with the base architecture with a learning rate of $2e-4$ on ChEBI-20. For LLM finetuning, we use LoRA with rank set to 8 and alpha set to 64, and trained for 6 epochs.

All experiments are performed with a single NVIDIA A100-SXM4-40GB. Finetuning the LLM takes about 5 hours, and training our diffusion model on ChEBI-20 takes about 1 day per run. Evaluating the diffusion model on ChEBI-20 takes about 4 hours per run.

B DATASETS

The synthetic dataset we used in the experiment was constructed based on the following rules:

- **Tree**: A tree with the specified number of node. An example of text description in the dataset is “A tree with 12 nodes”.
- **Cycle**: A cycle with the specified number of nodes. An example of text description in the dataset is “A tree with 12 nodes”.
- **Wheel**: A graph formed by connecting a single node to all nodes of a cycle. An example of text description in the dataset is “A wheel with 12 nodes”.
- **Bipartite**: A complete bipartite graph with the specified number of nodes. An example of text description in the dataset is “A complete bipartite graph with 6 nodes and 8 nodes in each split”.
- **K -regular graphs**: A graph whose every node have the same number of neighbors. An example of text description in the dataset is “A 2-regular graph with 12 nodes”.
- **Component**: A graph with the specified number of nodes and connected components. An example of text description in the dataset is “A graph with 12 nodes and 3 connected components”.
- **Mix**: A graph with the specified number of nodes and connected components, and every component is a graph that satisfies a certain rule. An example of text description in the dataset is “A graph with 12 nodes and 3 components. 2 components are trees. 1 component is a cycle.”.

C COMPUTATIONAL COSTS

Compared to unconditional graph diffusion models, our method only requires a small increase in training and inference time. For training our model, our method has the additional step of LLM finetuning, which takes a few days but only needs to be done once. For inference, our method, with the addition of text feature extraction and structure-aware cross-attention, only increased the inference time by about 30%.

D MORE EXPERIMENT RESULTS

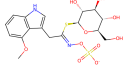
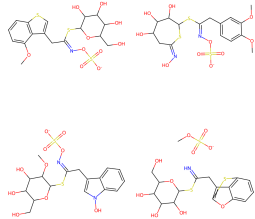
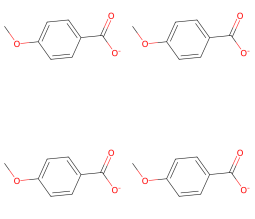
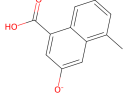
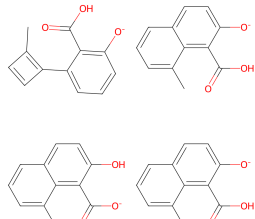
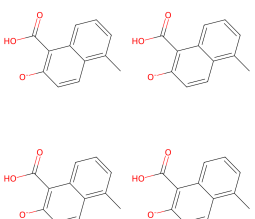
We list the numerical results of the diversity metrics in Table 6 in complement to the visualization in Figure 3.

In order to better illustrate the quality of generated graphs of our method, we list in Table 7 some comparison results with MolT5. It can be seen from the table that our method can generate diverse results while maintaining the basic molecular structure, while the diversity and accuracy of MolT5 is generally worse. This indicates that our method has more potential in the application of text-to-graph generation.

Table 6: The values of diversity and FTS under different parameters when generating molecules using various methods. "MACCS", "RDK", "Morgan" are three types methods calculating molecule fingerprint. "Diver." is diversity metric value. "FTS" is fingerprint-based molecule similarity metric.

Model	MACCS Diver.	MACCS FTS	RDK Diver.	RDK FTS	Morgan Diver.	Morgan FTS
Ours (CFG = 1)	0.748	0.726	0.581	0.544	0.416	0.382
Ours (CFG = 2)	0.776	0.770	0.613	0.601	0.454	0.448
Ours (CFG = 3)	0.780	0.778	0.615	0.607	0.454	0.455
Ours (CFG = 4)	0.780	0.785	0.620	0.619	0.457	0.465
Ours (CFG = 5)	0.786	0.790	0.620	0.623	0.464	0.476
Ours (CFG = 6)	0.785	0.790	0.622	0.627	0.458	0.471
Ours (CFG = 7)	0.474	0.543	0.332	0.397	0.213	0.275
Ours (CFG = 8)	0.321	0.398	0.195	0.264	0.099	0.158
Ours (CFG = 9)	0.314	0.374	0.191	0.243	0.098	0.140
Ours (CFG = 10)	0.359	0.338	0.230	0.217	0.137	0.119
MolT5 (t = 0.1)	0.740	0.807	0.594	0.794	0.542	0.795
MolT5 (t = 0.2)	0.746	0.807	0.596	0.784	0.543	0.784
MolT5 (t = 0.3)	0.734	0.823	0.583	0.798	0.536	0.792
MolT5 (t = 0.4)	0.740	0.821	0.585	0.786	0.538	0.777
MolT5 (t = 0.5)	0.730	0.807	0.574	0.768	0.529	0.756
MolT5 (t = 0.7)	0.736	0.807	0.580	0.757	0.536	0.744
MolT5 (t = 0.8)	0.734	0.797	0.577	0.739	0.529	0.727
MolT5 (t = 0.9)	0.734	0.798	0.574	0.740	0.525	0.719
MolT5 (t = 1.0)	0.727	0.757	0.569	0.678	0.528	0.669

Table 7: Visualization of generated molecules by our method and MolT5.

Text description	Ground truth	Our method	MolT5
<p>The molecule is an indolylmethylglucosinolate that is the conjugate base of 4-methoxyglucobrassicin, obtained by deprotonation of the sulfo group. It is a conjugate base of a 4-methoxyglucobrassicin.</p>			
<p>The molecule is a member of the class of naphthoates that is 1-naphthoate substituted at positions 3 and 5 by hydroxy and methyl groups respectively; major species at pH 7.3. It has a role as a bacterial metabolite. It is a conjugate base of a 3-hydroxy-5-methyl-1-naphthoic acid.</p>			
<p>The molecule is a myricetin O-glucuronide that is myricetin with a beta-D-glucosiduronic acid residue attached at the 5-position. It has a role as a metabolite. It is a myricetin O-glucuronide, a pentahydroxyflavone, a member of flavonols and a monosaccharide derivative.</p>	