EFFICIENT GRAPH GENERATION: BRIDGING COMPRESSION AND DIFFUSION MODELS FOR LARGE-SCALE GRAPHS

Dongyeong Hwang(20223735) *
Graduate School of AI
dy.hwang@kaist.ac.kr

Gunwook Nam(20235116)*
Department of Chemical and Biomolecular Engineering
gunwook@kaist.ac.kr

Yeongwoo Song(20228070)*
Department of Physics
ywsong1025@kaist.ac.kr

ABSTRACT

Graph generation poses challenges in handling large graphs due to their complexity. This paper presents a framework that effectively bridges existing graph compression methods with powerful generative models. Our key contribution lies in leveraging compression techniques and integrating them with the denoising diffusion model to address these challenges. By employing graph compression, we enable efficient diffusion processes and mitigate permutation problems. The framework facilitates the diffusion process by compressing the graph in terms of node count, allowing for the generation of large graphs with reduced computational complexity. Importantly, our approach ensures lossless decompression to preserve information during reconstruction. Experimental evaluation demonstrates the superior performance of our method compared to the base model, specifically on atom-level graphs. Our framework holds promise in advancing graph generation techniques and enabling their application across diverse domains, striking a balance between performance and efficiency while leveraging the power of the diffusion model.

1 INTRODUCTION

In recent years, graph generation has emerged as a highly popular and rapidly advancing research area, finding applications across diverse fields such as drug discovery and protein design. With the increasing use of machine learning models in this domain, there has been a growing interest in employing these models for graph generation tasks (You et al., 2018; Simonovsky & Komodakis, 2018). However, a significant challenge that persists in this realm is the generation of large graphs, primarily due to the $O(N^2)$ complexity involved.

On a different note, the denoising diffusion model (Sohl-Dickstein et al., 2015; Ho et al., 2020) has demonstrated exceptional performance as a generative model. At a high level, these models are trained to denoise diffusion trajectories and generate novel samples by iteratively sampling noise and denoising it. The success of diffusion models in various settings, particularly in the domain of image and video data, has ignited optimism for their application in graph generation tasks (Jo et al., 2022; Vignac et al., 2023). However, the implementation of the diffusion model for graph generation introduces a notable setback—the diffusion process typically relies on maintaining an adjacency matrix of $O(N^2)$ throughout, which exacerbates the scalability issue. While latent diffusion (Rombach et al., 2022) in the context of images has proven to be efficient, thanks to the diffusion process taking place in the latent space, the decoding process becomes nontrivial when applied to graphs. Furthermore, the measurement of reconstruction loss introduces a permutation problem that needs to be addressed.

*equal contribution

In light of these challenges, our proposed framework aims to address the limitations of existing approaches by introducing efficient graph compression techniques. By compressing the graph in terms of the number of nodes, we can effectively facilitate the diffusion process, making it more feasible to handle graphs with a large number of nodes and alleviating the burden of the $O(N^2)$ complexity. Importantly, our framework enables lossless decompression, ensuring that no information is lost during the reconstruction process. Since we measure the loss at the compressed level, we don't have to suffer from order-related permutation issues when decompressing.

In summary, our framework seeks to strike a balance between performance and efficiency by incorporating efficient graph compression techniques. This approach allows us to leverage the power of the diffusion model in graph generative tasks, even for larger molecules with a substantial $O(N^2)$ complexity. By overcoming the scalability and permutation challenges, our framework holds promise for advancing graph generation techniques and enabling their application in various domains.

2 RELATED WORKS

To generate molecules with desired properties using deep learning, it is necessary to determine the representation of the molecules and the structure of the generation model. In order for the model to accurately grasp the property distribution of molecules, a molecular representation is required that properly expresses the properties while accurately distinguishing between different molecules. On the other hand, the structure of the generation model is represented by variational autoencoders (VAEs) that form a latent space and generate new molecules from the distribution, diffusion model that learns the process of denoising noisy data and then generates denoised data from random data, and reinforcement learning that performs environment-optimized actions while obtaining rewards through the interaction of the environment.

Many deep learning-based chemical studies use the Simplified Molecular Input Line Entry System (SMILES) representation to represent molecules. While SMILES has the advantage of utilizing a highly developed language model, it has the unnecessary and difficult limitation of requiring the model to learn the SMILES grammar independently of understanding the distribution of chemical properties. If the grammar is not learned properly, it will generate invalid molecules, which is a fatal limitation of the model's performance. It also fails to uniquely represent a molecule because there is no single SMILES string that can represent a molecule. Olivecrona et al. (2017) used deep reinforcement learning to generate molecules from SMILES, and Gómez-Bombarelli et al. (2018) used a variational autoencoder to generate molecules represented by SMILES.

Recent work represents molecules as graphs with atoms as nodes and bonds as edges. This makes it easier for models to capture the properties of molecules by explicitly representing the bonds between atoms, which can more accurately describe the properties of the molecule. However, graph generative models have the difficulty of considering the permutation invariance property and discrete nature of graphs. Liu et al. (2018) generated graph-represented molecules with VAE, and Hoogeboom et al. (2022) successfully generated 3D molecules using molecule position as a continuous feature and atom type as a categorical feature. However, large molecules have $O(N^2)$ complexity, which means that the training process is long and the model is complex.

A strategy that bypasses this problem is to generate molecules at the molecular fragment-level rather than at the atom-level. Jin et al. (2018) and Kwon et al. (2020) showed how to form a latent space at the fragment level and generate molecules from it, and Gottipati et al. (2020) used reinforcement learning to generate molecules by adding fragments one by one.

3 DENOISING DIFFUSION MODEL

3.1 DENOISING DIFFUSION MODEL

Denoising diffusion models are a generative model that is gaining attention because of its excellent performance in generating images. The sample x0 is obtained from the dataset and then transformed into pure noise through a timestep T. The noise data is generated through the noising process

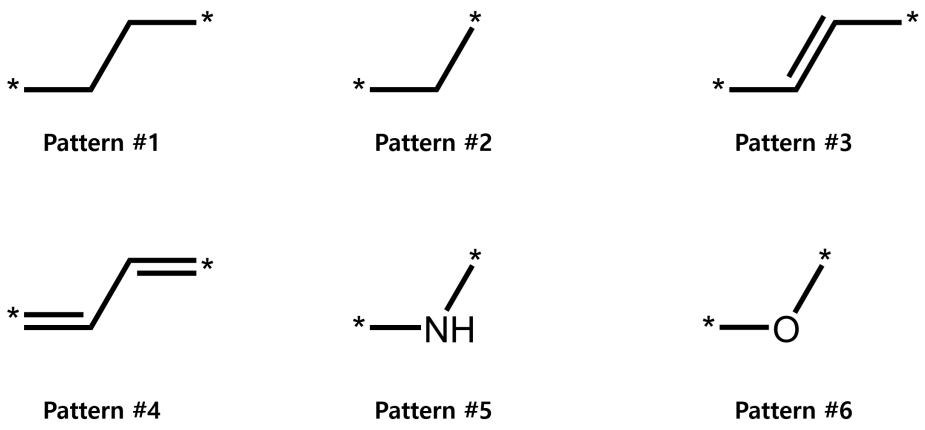

Figure 1: Substructural patterns that commonly appear between two atoms in molecules

$q(x^t|x^{t-1})$, and then the denoising process $p_\theta(x^{t-1}|x^t)$ is trained to a neural network with parameter θ. The denoising process is then applied to the data.

According to Hoogeboom et al. (2022), the transition matrix allows you to learn the final x^T instead of learning the next step in the process of denoising $p_\theta(x^{t-1}|x^t)$ of categorical features. This has the advantage that the learning is stable and fast. A transition matrix Q^t is follows:

$$Q^t_{ij} = q(x^t = j|x^{t-1} = i)$$

Therefore, we can define noising process as:

$$q(x^t|x^0) = Cat(x^t; p = x^{t-1} \cdot Q^t)$$

where Cat is the categorical distribution and x is feature vector.

4 Graph compression for scalable diffusion

4.1 Graph compression strategy

In this work, we propose a framework that leverages graph compression to apply the diffusion model to graphs with reduced node counts. Specifically, we adopt the compressed strategy introduced in Kwon et al. (2020). This strategy involves compressing the graph representation by reducing the number of nodes. We utilize six small substructural patterns commonly found between two heavy atoms, which are listed in Fig. 1. Each pattern consists of one or two heavy atoms, representing atom types such as C, N, and O, which are prevalent in real-world molecules. To represent the occurrences of these six substructural patterns, we incorporate additional edge features, which prove to be adequate for most real-world datasets. Formally, we define a compression function, ϕ, that compresses an input graph. Given the original graph, G, applying the ϕ function results in the corresponding compressed graph, G'.

$$G' = \phi(G) \tag{1}$$

The compression process involves identifying the substructures relevant to the designated patterns within the input graph, G. These substructures are sequentially transformed into edges by representing their occurrences using the corresponding edge feature. By doing so, we were able to reduce the average number of nodes by about 33.72% and the maximum number of nodes by about 40.91%(See Table 1.)

It is important to note that these substructural patterns can exhibit overlap between atoms, and in some cases, a pattern may appear multiple times. For instance, the simultaneous presence of patterns 3 and 4 between two atoms signifies the formation of a benzene ring. Moreover, the occurrence of pattern 1 twice indicates the formation of a hexagonal ring. To address this, the original paper introduces edge features in the form of concatenated one-hot counts for each pattern, in addition to

Table 1: Node reduction results

.

Statistics	Original	Compressed	Reduction rate(%)
AVG	27.70	18.36	33.72
MAX	88	52	40.91

representing single, double, and triple bond types. This results in a multivariate categorical vector representation. Check the original paper (Kwon et al., 2020) for further details.

However, a challenge arises when using DiGress, the diffusion model we intend to utilize, as it relies on categorical features. Consequently, we cannot employ the aforementioned representation as is. To overcome this challenge, we made the decision to only use edge feature types that occur more frequently than triple bonds and exclude molecules that cannot be expressed using this edge representation from the dataset. As a result, approximately 10% of the molecules were removed from the dataset.

4.2 GRAPH DIFFUSION MODEL

We utilize the discrete denoising diffusion model on our compressed-level graphs, which is the same as the one employed in Vignac et al. (2023). In this section, we provide a description of the model. To generate noisy samples, denoted as G^t, we select a timestep, t, and apply a cumulative transition matrix, $\bar{Q}^t$, to each node in X and each edge in E. The structure of the transition matrices can be represented as follows:

$$\bar{q}_X^t = \bar{\alpha}^t I + \bar{\beta}^t 1_{dx} m_X \quad \text{and} \quad \bar{Q}_E^t = \bar{\alpha}^t I + \bar{\beta}^t 1_{de} m_E \tag{2}$$

In these equations, $\bar{\alpha}^t$ and $\bar{\beta}^t$ are time-dependent scheduling variables, while m_X and m_E represent the marginal probabilities of each node and edge type, respectively, derived from the training data. The values of $\bar{\alpha}$ and $\bar{\beta}$ are adjusted over time using a popular cosine schedule $\bar{\alpha}^t = \cos(0.5\pi \frac{t/T+s}{1+s})^2$.

The denoising process is governed by a graph transformer neural network (Dwivedi & Bresson, 2020), which takes a noisy graph, G^t, as input and predicts the original unperturbed graph, $\hat{G}^0$. Training the denoising network involves utilizing a simple cross-entropy loss function that compares the predicted distribution of node types, $\hat{p}^X$, and edge types, $\hat{p}^E$, with the ground truth values, X_0 and E_0. The contributions of these comparisons are weighted by a hyperparameter, λ.

$$l(\hat{p}^G, G) = \sum_{1 \leq i \leq n} \text{cross-entropy}(x_i, \hat{p}_i^X) + \lambda \sum_{1 \leq i,j \leq n} \text{cross-entropy}(e_{ij}, \hat{p}_{ij}^E) \tag{3}$$

During the generation process, we begin by initializing a compressed-level graph randomly. The number of nodes in the graph is sampled according to the node count distribution observed in the training data. Additionally, the node types and edges are sampled based on the marginal probabilities of each node and edge type, represented as m_X and m_E, respectively. This stochastic sampling procedure results in the creation of a random compressed-level graph, denoted as $G^{t=T}$.

To predict the clean compressed-level graph, $\tilde{G}^0$, the graph transformer model takes the randomly generated compressed-level graph, $\hat{G}^t$, as input. The model leverages the posterior distribution $q(G^{t-1} \mid G, G^t)$ and performs marginalization over all possible values of x and e to obtain a partially-denoised graph, $\hat{G}^{t-1}$. This partial denoising step is repeated iteratively until a complete molecule, $\hat{G}^0$, is generated. Once the compressed-level molecule, $\hat{G}^0$, is generated, we can simply decompress it using the ϕ^{-1} function to obtain the fully uncompressed graph representation.

5 EXPERIMENTS

5.1 DATASET

We evaluate our model on a much more challenging dataset made of more than a million molecules: GuacaMol (Brown et al., 2019), which contains large drug-like molecules. From the Vignac et al.

Table 2: Molecule generation on GuacaMol. Note that ..

Model	Valid	Unique	Novel
DiGress	85.2	**100**	99.9
Ours	**92.2**	**100**	**100**

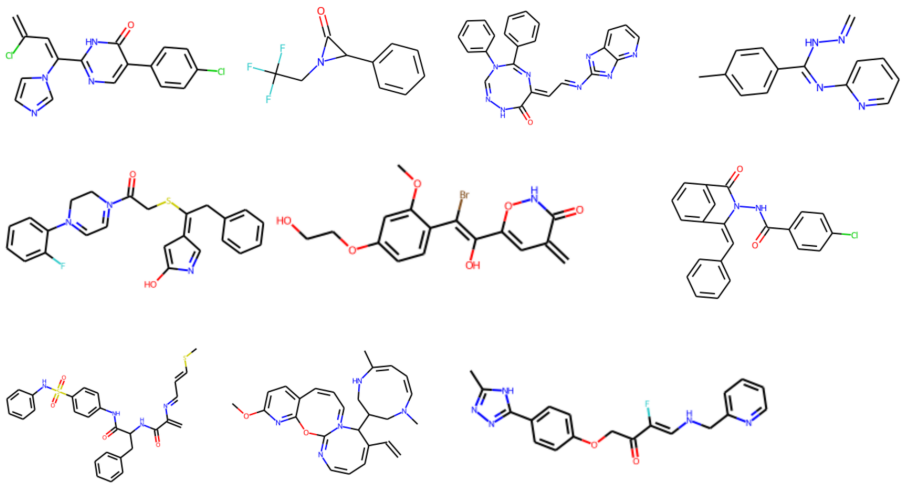

Figure 2: Samples generated by our model.

(2023), we note that GuacaMol contains complex molecules that are difficult to process, for example, because they contain formal charges or fused rings. As a result, mapping the train smiles to a graph and then back to a train SMILES does not work for around 20% of the molecules. We cut these 20% of the molecules.

5.2 METRIC

To evaluate our method, we compare qualitative metrics to baseline methods. Like Polykovskiy et al. (2020), we measure the chemical validity, uniqueness, and novelty of generated molecules in a qualitative manner. Detailed metrics are described below:

- **Validity**: percentage of chemically valid graphs
- **Uniqueness**: percentage of unique graphs
- **Novelty**: the proportion of generated molecules that are not in the training set.

5.3 MOLECULE GENERATION

The results are presented in Table 2. Our method outperforms the original model digress with atom-level graphs. To be honest, we had to figure out the trade-off between reducing the number of nodes but increasing the dimensionality of the edge features. However, it failed to directly compare GPU memory or training time, so the results could not be reported.

5.4 GENERATED MOLECULES

Finally, here are the generated samples. Please see Figure 2. While there are some invalid molecules, our model generates unique and novel molecules.

6 DISCUSSION

6.1 REDUCTION RATE

The model proposed in our study improved the ability to generate large molecules by utilizing the subgraph as a feature vector. Thus, the maximum number of nodes was reduced by 34% on average compared to the baseline model. It is worth noting that we only used feature vectors with a higher frequency than the feature vector represented by the six patterns proposed by (Kwon et al., 2020) based on the triple bonds that must be included in the feature vector. The patterns we used as feature vectors contain up to 4 atoms, and the node reduction rate would be much higher if larger patterns were used as features. However, the disadvantage is that larger patterns are less efficient in representing molecules because they occur less frequently in the dataset. The best representation method to represent molecules in this trade-off needs further research.

Furthermore, the node reduction rate of our model is noteworthy because we only used two redundant patterns and all other patterns used only one. (Kwon et al., 2020) used up to three redundant patterns to represent molecules, with patterns connecting different nodes, but this approach is less efficient than ours as I mention above. Our redundant patterns are all ring substructures, representing benzene and cyclohexane. The prevalence of hexagonal substructures in our dataset is a good indication that these structures are chemically stable.

6.2 QUALITATIVE METRICS

Our model achieved the same or better validity, uniqueness, and novelty compared to the baseline model. This indicates that the patterns we used in our model better train given dataset. In particular, the improved validity compared to the baseline model can be attributed to the fact that the pattern features we used are chemically valid substructures. In contrast, the atom-wise diffusion model has a relatively low validity because it does not learn chemistry well, but our model compensates for this limitation by using appropriate pattern features. However, it is still necessary to study how to generate chemically valid molecules, and we believe that it should be improved from both the representation point of view and the model point of view.

7 CONCLUSION

This paper proposes a generative model that creates a feature vector using subgraphs and uses it to generate large molecules using a discrete diffusion model. Compared to the existing discrete diffusion model using atom-wise features, we achieved higher validity, uniqueness, and novelty, and showed the ability to generate large molecules using subgraphs as features. Our future work is to validate the performance of the model on more diverse datasets containing large molecules.

REFERENCES

Nathan Brown, Marco Fiscato, Marwin HS Segler, and Alain C Vaucher. Guacamol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling*, 59(3): 1096–1108, 2019.

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020.

Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

Sai Krishna Gottipati, Boris Sattarov, Sufeng Niu, Yashaswi Pathak, Haoran Wei, Shengchao Liu, Simon Blackburn, Karam Thomas, Connor Coley, Jian Tang, et al. Learning to navigate the synthetically accessible chemical space using reinforcement learning. In *International Conference on Machine Learning*, pp. 3668–3679. PMLR, 2020.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, pp. 8867–8887. PMLR, 2022.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pp. 2323–2332. PMLR, 2018.

Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *International Conference on Machine Learning*, pp. 10362–10383. PMLR, 2022.

Youngchun Kwon, Dongseon Lee, Youn-Suk Choi, Kyoham Shin, and Seokho Kang. Compressed graph representation for scalable molecular graph generation. *Journal of Cheminformatics*, 12 (1):1–8, 2020.

Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander Gaunt. Constrained graph variational autoencoders for molecule design. *Advances in neural information processing systems*, 31, 2018.

Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9(1):1–14, 2017.

Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, et al. Molecular sets (moses): a benchmarking platform for molecular generation models. *Frontiers in pharmacology*, 11:565644, 2020.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I 27*, pp. 412–422. Springer, 2018.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. 2023.

Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models. In *International conference on machine learning*, pp. 5708–5717. PMLR, 2018.