FRAGFM: EFFICIENT FRAGMENT-BASED MOLECULAR GENERATION VIA DISCRETE FLOW MATCHING

Joongwon Lee*, **Seonghwan Kim***
Department of Chemistry, KAIST
{leejwon942,dmdtka00}@kaist.ac.kr

Wou Youn Kim†
Department of Chemistry, KAIST
Department of Data Science, KAIST
HITS
{wooyoun}@kaist.ac.kr

ABSTRACT

We introduce FragFM, a novel fragment-based discrete flow matching framework for molecular graph generation. FragFM generates molecules at the fragment level, leveraging a coarse-to-fine autoencoding mechanism to reconstruct atom-level details. This approach reduces computational complexity while maintaining high chemical validity, enabling more efficient and scalable molecular generation. We benchmark FragFM against state-of-the-art diffusion- and flow-based models on standard molecular generation benchmarks and natural product datasets, demonstrating superior performance in validity, property control, and sampling efficiency. Notably, FragFM achieves over 99% validity with significantly fewer sampling steps, improving scalability while preserving molecular diversity. These results highlight the potential of fragment-based generative modeling for large-scale, property-aware molecular design, paving the way for more efficient exploration of chemical space.

1 INTRODUCTION

Deep generative models, such as diffusion and flow matching, have demonstrated remarkable success across domains like images (Nichol et al., 2021; Rombach et al., 2022; Ho et al., 2020), text (Li et al., 2022), and videos (Hu & Xu, 2023; Ho et al., 2022). Recently, their application to molecular graph generation has gained attention, where they aim to generate chemically valid molecules by leveraging the structural properties of molecular graphs (Jo et al., 2022; Vignac et al., 2022; Qin et al., 2024).

However, existing atom-based generative models face scalability challenges, particularly in generating large and complex molecules. The quadratic growth of edges as graph size increases results in computational inefficiencies. At the same time, the inherent sparsity of chemical bonds makes accurate edge prediction more complex, often leading to unrealistic molecular structures or invalid connectivity constraints (Qin et al., 2023; Chen et al., 2023). Additionally, graph neural networks (GNNs) struggle to capture topological features such as rings and loops, leading to deviations from chemically valid structures. While various methods incorporate auxiliary features (e.g., spectral, ring, and valency information) to mitigate these issues, they do not fully resolve the sparsity and scalability bottlenecks (Vignac et al., 2022).

Fragment-based molecular generation has been explored as an alternative approach, inspired by its long-standing role in medicinal chemistry (Hajduk & Greer, 2007; Joseph-McCarthy et al., 2014; Kirsch et al., 2019). Instead of generating molecules atom by atom, fragment-based methods construct molecules using functional groups, ring systems, or chemically meaningful substructures, reducing complexity while preserving structural validity. This approach leverages established domain knowledge and significantly improves scalability by representing molecules as coarse-grained graphs. Within molecular generative frameworks, fragment-based methods enable more efficient exploration of the chemical space while maintaining structural coherence, offering better control over

*These authors contributed equally.
†Corresponding author

molecular properties than atom-based approaches (Jin et al., 2018; Qiang et al., 2023; Seo et al., 2023; Hetzel et al., 2023).

Recent studies (Levy & Rector-Brooks, 2023; Chen et al., 2024) utilizing diffusion models have demonstrated that fragment-level representations improve scalability and property control. However, many of these approaches rely on predefined fragment libraries or non-chemically driven fragmentation, which can constrain the accessible chemical space. Expanding the fragment library could mitigate this limitation, but it introduces computational overhead and modeling challenges in handling diverse fragment types. To address these challenges, we introduce FragFM, which is the first fragment-based discrete flow matching framework for molecular graph generation. Rather than being restricted to fixed fragment libraries, FragFM employs a sub-sampling strategy and GNN-based fragment embeddings, enabling generalization beyond predefined fragment sets while maintaining computational efficiency.

Through extensive benchmarking on MOSES and GuacaMol (Polykovskiy et al., 2020; Brown et al., 2019), we demonstrate that FragFM outperforms state-of-the-art diffusion and flow-based models in validity, property-based evaluation, and Fréchet ChemNet Distance (FCD) while requiring significantly fewer denoising steps. To further assess both scalability and the ability to capture high-level molecular semantics (e.g., biological relevance), we introduce a natural product benchmark based on the COCONUT dataset (Sorokina et al., 2021; Chandrasekhar et al., 2025), which contains large and structurally intricate molecules that present a demanding testbed for fragment-based generative modeling. Our findings suggest that integrating fragment-level representations into generative frameworks provides a scalable and interpretable solution for molecular design, paving the way for advancements in drug discovery, materials science, and functional molecular engineering.

2 FragFM Framework

We propose FragFM, a novel molecular generative framework that utilizes discrete flow matching (DFM) at the fragment level graph. In this approach, fragments and their connections are represented as nodes and edges respectively. This enables a discrete flow matching procedure on the resulting fragment-level graph. Because a single fragments arrangement can correspond to multiple molecular structures depending on how fragments junctions are permuted, we bridge fragment- and atom-level representations via a KL-regularized autoencoder that reconstructs the atom-level graph from its fragment-level graph with a latent variable. The learned latent variable contains the missing information during the fragmentation procedure, and it is generated through the flow matching model in conjunction with the fragment-level graph.

2.1 Fragment Graph Notation

We represent a molecule at the atom level as a graph $G = (V, E)$, where V is the set of atoms, and E represents chemical bonds between them. Each node $v_k \in V$ corresponds to a distinct atom, while an edge $e_{kl} \in E$ denotes a bond (including non-bond interactions) between atoms v_k and v_l. At the fragment level, we define a coarse-grained representation of the molecule as a graph $\mathcal{G} = (\mathcal{X}, \mathcal{E})$. Here, $x_i \in \mathcal{F}$ corresponds to a fragment, while each edge $\varepsilon_{ij} \in \mathcal{E}$ corresponds to the connectivity of the fragments. Each fragment is interpreted as an atom-level graph. Specifically, $\{x_i\}_i = \{(V_i, E_i)\}_i$ are disjoint sub-graphs of $G = (V, E)$, where $V_i \subseteq V$ and $E_i \subseteq E$, with $V_i \cap V_j = \varnothing$ for different fragment indices i, j. The edges in the fragment-level graph $\mathcal{E}$ are induced from E, meaning that two fragments $F_i, F_j \in \mathcal{X}$ are connected if at least one bond exists between their corresponding atoms, i.e.,

$$\varepsilon_{ij} \in \mathcal{E} \quad \text{if} \quad \exists\, e_{kl} \in E \quad \text{such that} \quad v_k \in V_i, v_l \in V_j. \tag{1}$$

2.2 Molecular Graph Compression by Coarse-to-fine Autoencoder

Recent advances in hierarchical generative models (Razavi et al., 2019; Rombach et al., 2022; Qiang et al., 2023) have demonstrated the effectiveness of learning structured latent representations through autoencoding, enabling efficient perceptual compression and reconstruction of complex data distributions. Motivated by this, we extend discrete generative modeling to molecular graphs by incorporating a coarse-to-fine autoencoding framework, where a fragment-level graph serves as a compressed representation of an atom-level graph. The fragment-level graph provides a higher-level

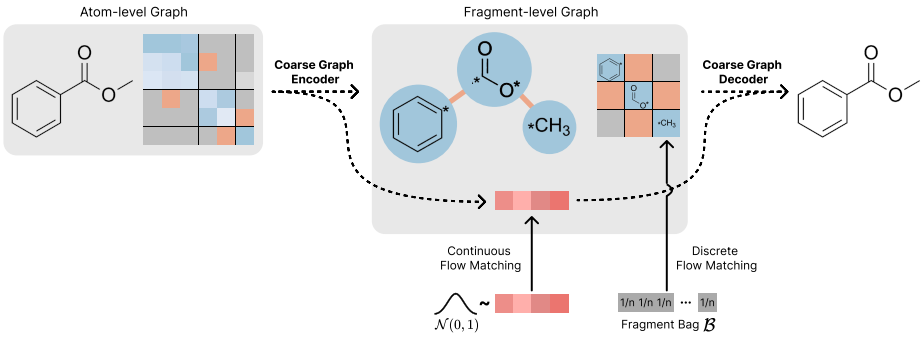

Figure 1: **Overview of the coarse-to-fine autoencoder architecture.**

abstraction of molecular structures. However, it inherently introduces ambiguity in reconstructing atomic connectivity due to multiple valid atom-level configurations corresponding to the same fragment arrangement. To resolve this, we introduce a coarse-to-fine autoencoder, where the encoder maps the atom-level graph to a fragment-level graph $\mathcal{G}$ along with a continuous latent variable z, and the decoder reconstructs the atom-level graph using both the fragment representation and the latent space. In practice, the decoder only predicts the atom-level connectivity of connected fragments, and the results are discretized by the Blossom algorithm (Edmonds, 1965). The details about coarse-to-fine conversion is described in Appendices B.1 and B.4

2.3 Discrete Flow Matching for Coarse Graph

We aim to model the joint distribution over the fragment-level graph and its latent representation $p(\mathcal{G}, z)$ with the flow-matching formulation. Because the connectivity variable $\varepsilon \in \mathcal{E}$ is binary, and the latent z is a real-valued vector with low dimension, we follow the DFM approach from Campbell et al. (2024) for ε and adopt the Lipman et al. (2022) for z. Although the fragment $x \in \mathcal{F}$ is also a discrete variable, the potentially large number of fragment types required to span the molecular space makes the transition rate matrix of continuous time Markov chain (CTMC) in the DFM approach prohibitively large. To address this, we introduce a stochastic bag selection strategy; for a fragment type variable x_1, we set a fragment type where the stochastic path $\{x_t\}_{t\in[0,1]}$ can take the values. For a given data $x_1^{1:D}$, we sample a type bag $\mathcal{B}$ that includes the types that are in present following a distribution, $\mathcal{B} \sim \mathbb{Q}(\cdot|x_1^{1:D})$, where D is the dimension of the discrete variables. With a clean data x_1 and $\mathcal{B}$, we define temporal marginal conditioned to $\mathcal{B}$, based on an linear interpolation with a prior distribution:

$$p_{t|1}(x_t|x_1, \mathcal{B}) = t\delta^{\mathcal{B}}(x_t, x_1) + (1-t)p_0(x_t|\mathcal{B}), \tag{2}$$

where $\delta^{\mathcal{B}}(\cdot,\cdot) = \mathbf{1}_{\mathcal{B}}(\cdot)\delta(\cdot,\cdot)$ represents the Kronecker delta multiplied by the indicator function restricted on $\mathcal{B}$ and $p_0(\cdot|\mathcal{B})$ is uniform over $\mathcal{B}$.

Discrete flow matching is learning a denoising process meeting the marginal distribution using a CTMC formulation. For a given fragments bag $\mathcal{B}$, sampling $x_t|\mathcal{B}$ begins with an initial distribution $p_0(\cdot|\mathcal{B})$ and propagates through a CTMC with transition rate $R_t(\cdot,\cdot|\mathcal{B})$ which only allows the transition between the states in the $\mathcal{B}$. The evolution of the process follows the Kolmogorov forward equation:

$$p_{t+dt|t}(y|x_t, \mathcal{B}) = \delta^{\mathcal{B}}(x_t, y) + R_t(x_t, y|\mathcal{B})dt. \tag{3}$$

In the sampling phase, x_1 is sampled based on eq. (3), which requires $\mathcal{B}$. Thus, the sampling phase begins with sampling $\mathcal{B}$ from $\mathbb{Q}$ and x_0 from $p_0(\cdot|\mathcal{B})$, where the unconditional bag distribution $\mathbb{Q}$ obtained by marginalizing out x_1, $\mathbb{E}_{x_1\sim p_{\text{data}}}[\mathbb{Q}(\cdot|x_1)]$. The neural network model approximates the distribution of x_1 given x_t and $\mathcal{B}$, which is utilized in computing the transition rate R_t. For the generalization of the neural network to diverse molecules, we incorporated a fragment embedding strategy, allowing the model to make predictions on novel fragment compositions. More details about coarse graph sampling and training algorithms are described in Appendix B.

3 RESULTS

3.1 MOLECULAR GRAPH GENERATION

Model	Class	Val. ↑	Unique. ↑	Novel ↑	Filters ↑	FCD ↓	SNN ↑	Scaf ↑
Training set	-	100.0	100.0	0.0	100.0	0.01	0.64	99.1
JT-VAE (Jin et al., 2018)	Fragment + AR	**100.0**	**100.0**	**99.9**	97.8	1.00	0.53	10.0
GraphINVENT (Mercado et al., 2021)	Atom + AR	96.4	99.8	-	95.0	1.22	0.54	12.7
DiGress (Vignac et al., 2022)	Atom + Diffusion	85.7	**100.0**	95.0	97.1	1.19	0.52	14.8
DisCo (Xu et al., 2024)	Atom + Diffusion	88.3	**100.0**	97.7	95.6	1.44	0.50	15.1
Cometh-PC (Siraudin et al., 2024)	Atom + Diffusion	90.5	99.9	92.6	99.1	1.27	0.54	16.0
DeFoG (# steps = 50) (Qin et al., 2024)	Atom + Flow	83.9	99.9	96.9	96.5	1.87	0.50	23.5
DeFoG (# steps = 500) (Qin et al., 2024)	Atom + Flow	92.8	99.9	92.1	98.9	1.95	0.55	14.4
FragFM (train fragments, # steps = 500)	Fragment + Flow	99.9	99.6	86.0	99.3	0.71	0.57	10.8

Table 1: **Molecule generation on MOSES dataset**. The upper part consists of auto-regressive methods, while the second part consists of iterative denoising methods, including diffusion-based and flow-based methods. The table compares their performance on several metrics.

We evaluate FragFM using the MOSES (Polykovskiy et al., 2020) and GuacaMol (Brown et al., 2019) benchmark datasets, following the dataset splits and evaluation metrics from Vignac et al. (2022). Across both MOSES (Table 1) and GuacaMol (Table 3), FragFM consistently outperforms existing diffusion- and flow-based models. While denoising-based models have traditionally lagged behind auto-regressive models, FragFM is the first to surpass them, achieving state-of-the-art Fréchet ChemNet Distance (FCD) on MOSES and KL divergence on GuacaMol. Notably, FragFM achieves over 99% validity while also demonstrating strong property-based performance (MOSES Filters, GuacaMol KL divergence), performing on par with JT-VAE and GraphINVENT. Furthermore, FragFM maintains uniqueness and novelty, highlighting its ability to assemble fragments into diverse molecular structures. We provide non-curated samples of generated molecules in Figures 6 and 7.

On the MOSES Scaf metric, FragFM tends to underperform when generation is restricted to training fragments, which can be attributed to the scaffold-split evaluation in MOSES. If a test molecule's scaffold is absent from the fragment bag, the model cannot generate it. However, when using test fragments, FragFM significantly improves its performance, demonstrating its ability to generalize to unseen fragments through the fragment embedding module.

3.2 NATURAL PRODUCT MOLECULE GENERATION

Model	Val. ↑	Unique. ↑	Novel ↑	NP Class KL Div. ↓			KL Div. ↓	FCD ↓
				Superclass	Class	Pathway		
Training set	100.0	100.0	0.0	0.0054	0.1243	0.0002	0.0042	0.13
DiGress (Vignac et al., 2022)	85.7	**99.9**	**99.4**	0.2309	1.0783	0.1842	0.1654	2.13
FragFM	**92.2**	96.3	95.8	**0.0879**	**0.4517**	**0.0149**	**0.0182**	**1.21**

Table 2: **Natural product generation benchmark.**

Understanding natural products is crucial as they serve as a rich source of bioactive compounds and provide valuable insights for drug discovery (Atanasov et al., 2021; Newman & Cragg, 2020). Additionally, the COCONUT dataset includes a hierarchical classification scheme (pathway, superclass, class) that captures structural and biosynthetic relationships, enabling a more in-depth evaluation of generative models on complex molecular categories. Details on the dataset and evaluation metrics are provided in Appendix C.2.2.

We compared FragFM with DiGress on the COCONUT benchmark, as summarized in Table 2. While both models achieve high validity, validity alone does not guarantee that generated molecules resemble natural products. FragFM outperforms DiGress, by achieving lower KL divergence across pathway, superclass, class, and NP-likeness scores, indicating a closer alignment with the training set natural products. This suggests that fragment-based modeling more effectively captures molecular

structural characteristics, leading to the generation of more biologically relevant molecules. Example molecules generated by FragFM and DiGress are shown in Figures 8 to 10.

3.3 Sampling Efficiency

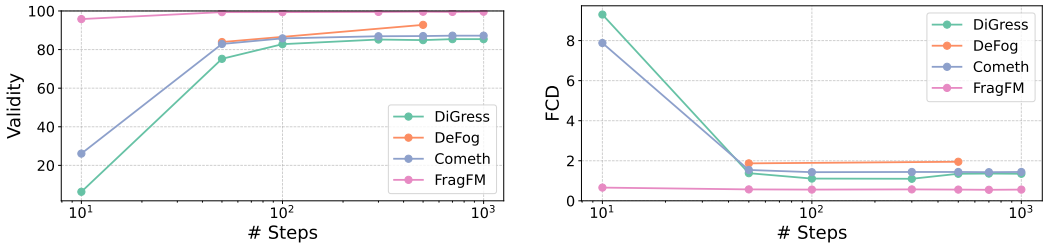

Figure 2: **Analysis of sampling steps across multiple denoising models.** FragFM consistently outperforms baseline models as the number of sampling steps decreases, exhibiting less degradation in sampling quality. Additional metrics and further results are provided in Appendix D.3.

Generative models based on the denoising process required multiple iterative steps, making sampling inefficient. Therefore, reducing the number of denoising steps while maintaining generation quality is crucial for improving efficiency and scalability in molecular generation. Figure 2 and Table 5 present the MOSES benchmark results across different denoising steps for various generative models. As expected, reducing the number of denoising steps generally leads to a decrease in generation quality. However, FragFM exhibits a significantly lower decline in quality and consistently outperforms other models, achieving over 95%validity and an FCD of 0.66 with just 10 steps, exceeding other models. This is likely due to the fragment-based discrete flow matching approach, which reduces the number of edges that need to be predicted and allows for more stable intermediate representations during generation. Further details and sampling efficiency analysis are provided in Appendices D.3 and D.4.

3.4 Conditional Generation

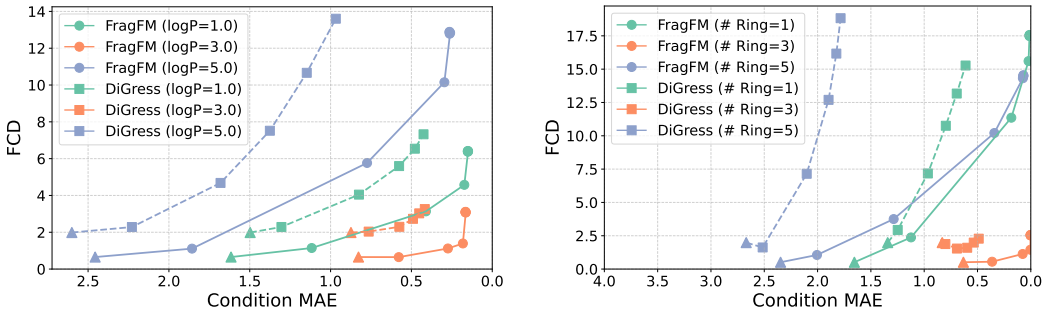

Figure 3: **Condition MAE and FCD curves for logP and number of rings.** The triangle denotes the unconditional models. Each curve is color-coded to represent a model with a different target value. Additional results for QED and TPSA are provided in Appendix D.5.

Conditional generation is crucial in molecular design, enabling precise control over molecular properties. We integrate classifier-free guidance (CFG) (Ho & Salimans, 2022; Nisonoff et al., 2024) to FragFM (details in Appendix B.6) to guide the generation process towards desired property values. We conduct conditional molecular generation on the MOSES dataset to evaluate its effectiveness, targeting logP, number of rings, QED, and TPSA.

From Figures 3 and 5, we observe that FragFM achieves a lower condition MAE while maintaining a lower FCD compared to DiGress, positioning our method on the Pareto-optimal frontier in the FCD-Condition MAE trade-off. The improvement can be attributed to the structured generative process

of FragFM, where molecules are assembled from semantically meaningful fragments, allowing for better preservation of structural patterns and improved property control. We provide further details on the CFG and experiments in Appendix D.5.

4 CONCLUSION

We introduce FragFM, a fragment-based discrete flow matching framework for molecular graph generation. By leveraging fragment-level representations with a coarse-to-fine autoencoder and a fragment bag selection approach, FragFM enables efficient and accurate molecular generation while preserving structural diversity and validity. Extensive benchmarks on MOSES, GuacaMol, and natural product datasets demonstrate that FragFM consistently outperforms existing diffusion and flow-based models across multiple metrics, including validity and Fréchet ChemNet Distance (FCD), while requiring fewer denoising steps.

Additionally, we explore efficient sampling strategies and classifier-free guidance, showcasing the effectiveness of FragFM in property-aware molecular generation. Our results highlight the potential of fragment-based modeling to improve the scalability and expressiveness of denoising-based molecular generation frameworks. Looking forward, an important direction is to enhance FragFM by incorporating fragment bag control, allowing for more precise selection and optimization of fragment compositions. By dynamically guiding fragment selection based on molecular properties and downstream design objectives, this approach could further improve the controllability of molecular generation across diverse applications, including protein-ligand binding design, OLED materials, and novel functional molecules.

ACKNOWLEDGMENTS

Basic Science Research Programs supported this work through the National Research Foundation of Korea (NRF-2023R1A2C2004376, NRF-2022M3J6A1063021, RS-2023-00257479, RS-2024-00512498).

REFERENCES

Atanas G Atanasov, Sergey B Zotchev, Verena M Dirsch, and Claudiu T Supuran. Natural products in drug discovery: advances and opportunities. *Nature reviews Drug discovery*, 20(3):200–216, 2021.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

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.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv preprint arXiv:2402.04997*, 2024.

Venkata Chandrasekhar, Kohulan Rajan, Sri Ram Sagar Kanakam, Nisha Sharma, Viktor Weißenborn, Jonas Schaub, and Christoph Steinbeck. Coconut 2.0: a comprehensive overhaul and curation of the collection of open natural products database. *Nucleic Acids Research*, 53(D1):D634–D643, 2025.

Xiaohui Chen, Jiaxing He, Xu Han, and Li-Ping Liu. Efficient and degree-guided graph generation via discrete diffusion modeling. *arXiv preprint arXiv:2305.04111*, 2023.

Zijun Chen, Yu Wang, Liuzhenghao Lv, Hao Li, Zongying Lin, Li Yuan, and Yonghong Tian. Multi-granularity score-based generative framework enables efficient inverse design of complex organics. *arXiv preprint arXiv:2409.07912*, 2024.

Jorg Degen, Christof Wegscheid-Gerlach, Andrea Zaliani, and Matthias Rarey. On the art of compiling and using'drug-like'chemical fragment spaces. *ChemMedChem*, 3(10):1503, 2008.

Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of mathematics*, 17:449–467, 1965.

Floor Eijkelboom, Grigory Bartosh, Christian Andersson Naesseth, Max Welling, and Jan-Willem van de Meent. Variational flow matching for graph generation. *arXiv preprint arXiv:2406.04843*, 2024.

Peter Ertl, Silvio Roggo, and Ansgar Schuffenhauer. Natural product-likeness score and its application for prioritization of compound libraries. *Journal of chemical information and modeling*, 48 (1):68–74, 2008.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *arXiv preprint arXiv:2407.15595*, 2024.

Philip J Hajduk and Jonathan Greer. A decade of fragment-based drug design: strategic advances and lessons learned. *Nature reviews Drug discovery*, 6(3):211–219, 2007.

Leon Hetzel, Johanna Sommer, Bastian Rieck, Fabian Theis, and Stephan Günnemann. Magnet: Motif-agnostic generation of molecules from shapes. *arXiv preprint arXiv:2305.19303*, 2023.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

Xiaoyang Hou, Tian Zhu, Milong Ren, Dongbo Bu, Xin Gao, Chunming Zhang, and Shiwei Sun. Improving molecular graph generation with flow matching and optimal transport. *arXiv preprint arXiv:2411.05676*, 2024.

Zhihao Hu and Dong Xu. Videocontrolnet: A motion-guided video-to-video translation framework by using diffusion model with controlnet. *arXiv preprint arXiv:2307.14073*, 2023.

Jan H Jensen. A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. *Chemical science*, 10(12):3567–3572, 2019.

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.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Multi-objective molecule generation using interpretable substructures. In *International conference on machine learning*, pp. 4849–4859. PMLR, 2020.

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.

Diane Joseph-McCarthy, Arthur J Campbell, Gunther Kern, and Demetri Moustakas. Fragment-based lead discovery and design. *Journal of chemical information and modeling*, 54(3):693–704, 2014.

Hyun Woo Kim, Mingxun Wang, Christopher A Leber, Louis-Félix Nothias, Raphael Reher, Kyo Bin Kang, Justin JJ Van Der Hooft, Pieter C Dorrestein, William H Gerwick, and Garrison W Cottrell. Npclassifier: a deep neural network-based structural classification tool for natural products. *Journal of Natural Products*, 84(11):2795–2807, 2021.

Jun Hyeong Kim, Seonghwan Kim, Seokhyun Moon, Hyeongwoo Kim, Jeheon Woo, and Woo Youn Kim. Discrete diffusion schr\" odinger bridge matching for graph transformation. *arXiv preprint arXiv:2410.01500*, 2024.

Philine Kirsch, Alwin M Hartman, Anna KH Hirsch, and Martin Empting. Concepts and core principles of fragment-based drug design. *Molecules*, 24(23):4309, 2019.

Xiangzhe Kong, Wenbing Huang, Zhixing Tan, and Yang Liu. Molecule generation by principal subgraph mining and assembling. *Advances in Neural Information Processing Systems*, 35:2550–2563, 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: 1–8, 2020.

Daniel Levy and Jarrid Rector-Brooks. Molecular fragment-based diffusion model for drug discovery. In *ICLR 2023-Machine Learning for Drug Discovery workshop*, 2023.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343, 2022.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Tairan Liu, Misagh Naderi, Chris Alvin, Supratik Mukhopadhyay, and Michal Brylinski. Break down in order to build up: decomposing small molecules for fragment-based drug design with e molfrag. *Journal of chemical information and modeling*, 57(4):627–631, 2017.

Krzysztof Maziarz, Henry Jackson-Flux, Pashmina Cameron, Finton Sirockin, Nadine Schneider, Nikolaus Stiefl, Marwin Segler, and Marc Brockschmidt. Learning to extend molecular scaffolds with structural motifs. *arXiv preprint arXiv:2103.03864*, 2021.

Rocío Mercado, Tobias Rastemo, Edvard Lindelöf, Günter Klambauer, Ola Engkvist, Hongming Chen, and Esben Jannik Bjerrum. Graph networks for molecular design. *Machine Learning: Science and Technology*, 2(2):025023, 2021.

David J Newman and Gordon M Cragg. Natural products as sources of new drugs over the nearly four decades from 01/1981 to 09/2019. *Journal of natural products*, 83(3):770–803, 2020.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance for discrete state-space diffusion and flow models. *arXiv preprint arXiv:2406.01572*, 2024.

Chenhao Niu, Yang Song, Jiaming Song, Shengjia Zhao, Aditya Grover, and Stefano Ermon. Permutation invariant graph generation via score-based generative modeling. In *International Conference on Artificial Intelligence and Statistics*, pp. 4474–4484. PMLR, 2020.

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.

Bo Qiang, Yuxuan Song, Minkai Xu, Jingjing Gong, Bowen Gao, Hao Zhou, Wei-Ying Ma, and Yanyan Lan. Coarse-to-fine: a hierarchical diffusion model for molecule generation in 3d. In *International Conference on Machine Learning*, pp. 28277–28299. PMLR, 2023.

Yiming Qin, Clement Vignac, and Pascal Frossard. Sparse training of discrete diffusion models for graph generation. *arXiv preprint arXiv:2311.02142*, 2023.

Yiming Qin, Manuel Madeira, Dorina Thanou, and Pascal Frossard. Defog: Discrete flow matching for graph generation. *arXiv preprint arXiv:2410.04263*, 2024.

Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

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.

Seonghwan Seo, Jaechang Lim, and Woo Youn Kim. Molecular generative model via retrosynthetically prepared chemical building block assembly. *Advanced Science*, 10(8):2206674, 2023.

Antoine Siraudin, Fragkiskos D Malliaros, and Christopher Morris. Cometh: A continuous-time discrete-state graph diffusion model. *arXiv preprint arXiv:2406.06449*, 2024.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Maria Sorokina, Peter Merseburger, Kohulan Rajan, Mehmet Aziz Yirik, and Christoph Steinbeck. Coconut online: collection of open natural products database. *Journal of Cheminformatics*, 13(1): 2, 2021.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*, 2022.

Zhe Xu, Ruizhong Qiu, Yuzhong Chen, Huiyuan Chen, Xiran Fan, Menghai Pan, Zhichen Zeng, Mahashweta Das, and Hanghang Tong. Discrete-state continuous-time diffusion for graph generation. *arXiv preprint arXiv:2405.11416*, 2024.

A Related Works

A.1 Denoising Graph Generative Models

Denoising-based generative models have become fundamental for molecular graph generation by iteratively refining noisy graphs into structured molecular representations. Diffusion methods (Ho et al., 2020; Song et al., 2020), which have been successful in a variety of domains, have been extended to graph structure data (Jo et al., 2022; Niu et al., 2020), demonstrating the advantages of applying diffusion in graph generation. This approach was further extended by incorporating discrete stochastic processes (Austin et al., 2021), addressing the inherently discrete nature of molecular graphs (Vignac et al., 2022). The discrete diffusion modeling is reformulated by the continuous time Markov chain (CTMC), which has been introduced (Xu et al., 2024; Siraudin et al., 2024; Kim et al., 2024), allowing more flexible and adaptive generative processes. More recently, flow-based models have been explored for molecular graph generation. Continuous flow matching (Lipman et al., 2022) has been applied to structured data (Eijkelboom et al., 2024), while discrete flow models (Campbell et al., 2024; Gat et al., 2024) have been extended to categorical data generation, with recent methods demonstrating their effectiveness in modeling molecular distributions (Qin et al., 2024; Hou et al., 2024).

A.2 Fragment Based Molecule Generation

Fragment-based molecular generative models construct new molecules by assembling existing molecular substructures, known as fragments. This strategy enhances chemical validity and facilitates the efficient exploration of novel molecular structures. Several works have employed fragment-based approaches within variational autoencoders (VAEs). Jin et al. (2020); Kong et al. (2022); Maziarz et al. (2021) generate molecules using VAEs by learning to assemble fragments in a chemically meaningful way. Jin et al. (2018) adopts a stepwise generation approach, first constructing a coarse fragment-level graph before refining it into an atom-level molecule through substructure

completion. Seo et al. (2023); Jin et al. (2020) construct molecules by assembling fragments sequentially, enabling better control over molecular properties during generation.

Fragment-based approaches have also been explored in diffusion-based molecular graph generation. Levy & Rector-Brooks (2023) proposed a method that utilizes a fixed set of frequently occurring fragments to generate drug-like molecules, ensuring chemical validity but limiting exploration beyond predefined structures. Since enumerating all possible fragment types is infeasible, the method operates solely within a fixed fragment vocabulary. In contrast, Chen et al. (2024) introduced an alternative fragmentization strategy depending on the dataset based on byte-pair encoding, offering a more flexible molecular representation. However, this approach cannot still incorporate chemically meaningful fragmentation methods (Degen et al., 2008; Liu et al., 2017), making it challenging to utilize domain-specific chemical priors.

B Method Details

B.1 Details of Coarse-to-fine Autoencoder

We adopted a KL-regularized autoencoder for coarse-to-fine graph conversion. The coarse-grained graph representation $\mathcal{G}$ can be interpreted as a compressed version of atom-level fine graph G. In the fragmentation procedure, atom-level graph loose the fine-grained connection information. For reconstructing the original atom-level graph, the fragment-level graph and the missing information is required, which is encoded in the latent variable z. Formally, the encoding and decoding process is defined as:

$$\begin{aligned} \mathcal{G} &= \text{Fragmentation}(G), \\ z &\sim q_\theta = \mathcal{N}(\text{Encoder}(G;\theta), \sigma), \\ \hat{E} &= \text{Decoder}(\mathcal{G}, z; \theta), \end{aligned} \tag{4}$$

where the decoder reconstructs only those atom-level edges $\hat{E}$ corresponding to the fragment connectivity in the coarse representation.

To ensure that the reconstructed graph faithfully preserves the original molecular structure, we optimize the autoencoder using a reconstruction loss. Additionally, we introduce a small KL regularization term to the training loss for latent variable to enforce a well-structured and unscaled latent space:

$$\mathcal{L}_{\text{VAE}}(\theta) = \mathbb{E}_{G\sim p_{\text{data}}}\left[\mathcal{L}_{\text{CE}}\left(E, \hat{E}(\theta)\right) + \beta D_{\text{KL}}\left(q_\theta(z|G) \parallel p(z)\right)\right]. \tag{5}$$

We set a low regularization coefficient of $\beta = 0.0001$ to maintain high-fidelity reconstruction.

We discretize the decoded edges $\hat{E}$ during the fine-graph sampling procedure. In this process, we employ the blossom algorithm detailed in Appendix B.4, which yields robust sampling performance.

B.2 Fragment Denoising Flow matching

For sampling, we now want to design a stochastic process that meets the temporal marginal distributions of eq. (2) along with the Kolmogorov forward equation eq. (3). Campbell et al. (2024) proposed such a transition rate conditioned on x_1 that aligns with the linear interpolated distributions, and we modified the equation in the $\mathcal{B}$ conditioned form, which is as:

$$R_t^*(x_t, y|x_1, \mathcal{B}) = \frac{\text{ReLU}\Big[\partial_t p_{t|1}(y|x_1, \mathcal{B}) - \partial_t p_{t|1}(x_t|x_1, \mathcal{B})\Big]}{Z_t^{>0} p_{t|1}(x_t|x_1, \mathcal{B})} \quad \text{for} \quad x_t \neq y, \tag{6}$$

while $Z_t^{>0} = |z_t : p_{t|1}(x_t|x_1, \mathcal{B}) > 0|$. The entries for $z_t = y$ are calculated by normalization. For a D dimensional case, where D is the number of dimensions we model, Campbell et al. (2024) proposed an approximation of CTMC under mild conditions, independently conducting the Euler step for each dimension. With a finite time step Δt:

$$\tilde{p}_{t+\Delta t|t}\left(x_{t+\Delta t}^{1:D}|x_t^{1:D}, \mathcal{B}\right) = \prod_{d=1}^{D}\left(\delta^{\mathcal{B}}\left(x_t^{(d)}, x_{t+\Delta t}^{(d)}\right) + \mathbb{E}_{p_{1|t}^{(d)}\left(x_1^{(d)}|x_t^{1:D}, \mathcal{B}\right)}\left[R_t^{(d)}\left(x_t^{(d)}, x_{t+\Delta t}^{(d)}|x_1^{(d)}, \mathcal{B}\right)\right]\Delta t\right). \tag{7}$$

Based on the transition kernel on eq. (7), we can sample x_1 given the fragments bag $\mathcal{B}$. To define $\mathbb{Q}(\cdot|x_1)$ for sampling fragment bags, we use a two-stage procedure: (1) select a fixed number of molecules from the data distribution and include x_1 among them, and (2) gather all fragments from these molecules to form the fragment bag $\mathcal{B}$. During training, we sample $\mathbb{Q}(\cdot|x_1)$. However, in the sampling (inference) phase, x_1 is not available a priori, so we employ the unconditional distribution $\mathbb{Q}$ given by,

$$\mathbb{Q} = \mathbb{E}_{x_1 \sim p_{\text{data}}} \left[\mathbb{Q}(\cdot|x_1) \right]. \tag{8}$$

Sampling from $\mathbb{Q}$ follows a similar two-stage procedure, except we do not force the inclusion of any particular x_1 in the initial set of molecules. In practice, we gathered 256 molecules from the training dataset to sample the fragments bag.

B.3 Neural Network Parameterization

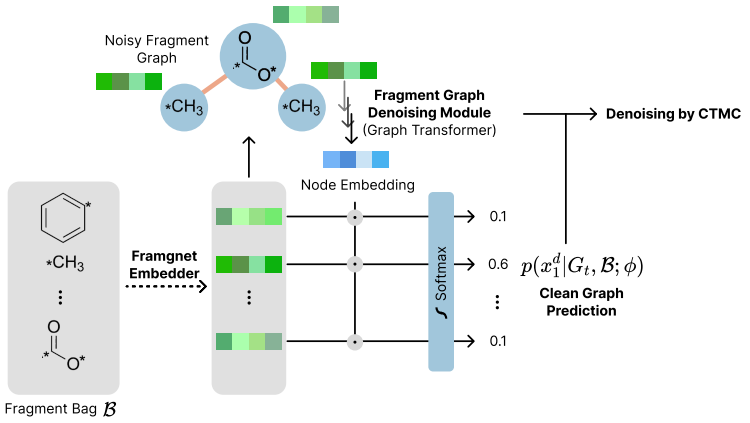

Figure 4: **Schematic of FragFM's fragment denoising module.**

First, we model the coarse-to-fine autoencoder with simple MPNN. We model $p_{1|t}(G_1|G_t;\phi)$ using a fragment embedding message passing neural network (fragment encoder) and a graph transformer (GT) to facilitate message passing in the fragment-level graph. Through fragment MPNN, each fragment type in $\mathcal{F}$ is represented as a 1-dimenstional latent vector:

$$h_i = \text{FragmentEncoder}(x_i;\phi), \quad \text{for} \quad x_i \in \mathcal{F}. \tag{9}$$

Using these embedded fragment representations, we construct a fragment-level graph where nodes correspond to individual fragment embeddings, edges representing fragment-level connectivity, and global features z, a latent variable from the coarse-to-fine autoencoder. We incorporate the graph transformer backbone from Vignac et al. (2022); Qin et al. (2024) to propagate information across the fragment graph. After l layers of graph transformer, we obtain node embeddings, edge embeddings, and the global embedding for the fragment graph, denoted as $h_i^{(l)}, e_{ij}^{(l)}, g^{(l)}$. For fragment edge types e_{ij} and the continuous latent variable z, FragFM utilizes simple linear layers. For fragment type prediction, we perform a softmax operation over the fragment bag, where the logit score is computed as the inner product between the fragment embeddings and the node embeddings:

$$\hat{p}_i = \text{Softmax}\left(\{ h_i^{(l)} \cdot h_k^{(0)} \}_{x_k \in \mathcal{B}} \right). \tag{10}$$

B.4 Atom-level Graph Reconstruction from Fragment Graphs

We utilize the Blossom algorithm (Edmonds, 1965) to determine the optimal matching in the atom-level connectivity given coarse-to-fine decoder output. The Blossom algorithm is an optimization technique used to find the maximum matching in general graphs by iteratively contracting and expanding odd-length cycles (blossoms) to identify augmenting paths efficiently. We leverage this algorithm in our framework to accurately reconstruct atom-level connectivity from fragment-level graphs, ensuring chemically valid molecular structures. The algorithm takes as input the matching

nodes V_m, edges E_m, and edge weights w_{ij}. Once the fragment-level graph and the probabilities of atom-level edges from the coarse-to-fine autoencoder are computed, we define $V_m \subseteq \hat{V}$ as the set of junction atoms in fragment graphs, which are marked as * in Figure 11, and E_m as the set of connections between junction atoms belonging to connected fragments. Formally, an edge e_{kl} exists in E_m if the corresponding atoms belong to different fragments that are connected in the fragment-level graph, expressed as:

$$e_{kl} \in E_m \quad \text{if} \quad v_k \in \hat{V}_i, \quad v_l \in \hat{V}_j, \quad \text{and} \quad \varepsilon_{ij} \in \mathcal{E}. \tag{11}$$

The edge weights w_{ij} correspond to the predicted log probability of each connection obtained from the coarse-to-fine autoencoder. The Blossom algorithm is then applied to solve the maximum weighted matching problem, formulated as

$$M^* = \text{argmax}_{M \subseteq E_m} \sum_{(i,j) \in M} w_{ij}. \tag{12}$$

Here, M^* represents the optimal set of fragment-level connections that best reconstructs atom-level connectivity; maximizing the joint probability of the autoencoder preidction. Although the algorithm has a $O(N^3)$ complexity for N fragment junctions, its computational cost remains negligible in our case, as the number of fragment junctions is relatively tiny compared to the total number of atoms in a molecule.

B.5 Sampling Techniques

B.5.1 Target Guidance

Diffusion and flow models are typically designed to predict the clean data $G_{f,1} = \{V_{f,1}, E_{f,1}\}$ from noisy input. Building on this, (Qin et al., 2024) proposed a modified sampling method for DFM by adjusting the rate matrix toward the predicted clean data. Specifically, the rate matrix is redefined as

$$R_t(x_t, y \mid x_1) = R_t^*(x_t, y \mid x_1) + R_t^\omega(x_t, y \mid x_1) \tag{13}$$

for $x_t \neq y$, where

$$R_t^\omega(x_t, y \mid x_1) = \omega \frac{\delta(y, x_1)}{Z_t^{>0} p_{t|1}(x_t \mid x_1)} \tag{14}$$

This modification introduces a slight $O(\omega)$ violation of the Kolmogorov equation. However, empirical findings suggest that a small ω improves sample quality without significantly distorting the learned distribution. For our results, we use a small value of $\omega = 0.002$.

B.5.2 Detailed Balance

The space of valid rate matrices extends beyond the original formulation of $R_t^*(z_t, z_{t+dt} \mid z_1)$, meaning that alternative formulations can still satisfy the Kolmogorov equation. Building on this, Campbell et al. (2024) explored this space and demonstrated that any rate matrix R_t^{DB} satisfying the detailed balance condition,

$$p_{t|1}(x_t \mid x_1) R_t^{DB}(x_t, y \mid x_1) = p_{t|1}(y \mid x_1) R_t^{DB}(y, x_t \mid x_1), \tag{15}$$

can be used to construct a modified rate matrix:

$$R_t^\eta = R_t^* + \eta R_t^{DB}, \quad \eta \in \mathbb{R}^+, \tag{16}$$

which remains consistent with the Kolmogorov equation. Increasing η introduces additional stochasticity into CTMC, enabling more transition pathways between states. We integrate this stochasticity into our FragFM framework, enabling variability in fragment-type transitions while maintaining valid generative pathways. We set $\eta = 0.1$ in our experiments.

B.6 Classifier-Free Guidance

Classifier-Free Guidance (CFG) allows for controllable molecular generation by interpolating between conditioned and unconditioned models, eliminating the need for explicit property classifiers (Ho & Salimans, 2022). This approach, widely used in continuous diffusion models, was recently

extended to discrete flow matching (Nisonoff et al., 2024). We adopt this technique to enhance the controllability of FragFM while maintaining sample diversity.

During training, FragFM learns both conditional rate matrix $R_t^\theta(x_t, y \mid c)$ and an unconditional rate matrix $R_t^\theta(x_t, y)$ simultaneously by conditioning on property labels c for 90% of samples and conditioning with the masked label φ for the remaining 10%. During sampling, the rate matrix is adjusted using the guidance level γ:

$$R_t^{\theta,\gamma}(x_t, y \mid c) = R_t^\theta(x_t, y \mid c)^\gamma R_t^\theta(x_t, y \mid \varphi)^{1-\gamma}. \tag{17}$$

Setting $\gamma = 0$ corresponds to purely unconditional generation while increasing γ strengthens class adherence, biasing transitions toward the desired property distribution. However, excessively high γ values can constrain exploration, generating molecules that diverge from the overall unconditional data distribution. Ho & Salimans (2022) demonstrated that this trade-off follows a characteristic pattern: as guidance strength increases, sample fidelity (e.g., lower FID) improves at the cost of class adherence (e.g., lower IS).

C Experimental Details

C.1 Baselines

Our experiments compare FragFM with several state-of-the-art graph and molecule generative models. For non-denoising methods, we include JT-VAE (Jin et al., 2018), GraphINVENT (Mercado et al., 2021), NAGVAE (Kwon et al., 2020), and MCTS (Jensen, 2019). For diffusion-based models, we evaluate DiGress (Vignac et al., 2022), DisCo (Xu et al., 2024), and Cometh (Siraudin et al., 2024). In addition, we compare with DeFog (Qin et al., 2024) for flow-based approaches.

C.2 Dataset and Metrics

C.2.1 MOSES and GuacaMol

MOSES and GuacaMol provide standardized molecular-generation benchmarking frameworks, offering predefined training, test datasets, and automated evaluation metrics. Validity refers to the percentage of generated molecules that adhere to fundamental valency constraints, ensuring chemically plausible structures. Uniqueness quantifies the proportion of generated molecules with distinct SMILES representations, indicating non-isomorphism. Novelty measures the number of generated molecules that do not appear in the training dataset, assessing the model's ability to create new structures.

The filter score evaluates the percentage of molecules that satisfy the same chemical constraints applied during test set construction. Fréchet ChemNet Distance (FCD) quantifies the similarity between training and test molecules based on learned neural network embeddings. SNN (Similarity to Nearest Neighbor) captures how closely generated molecules resemble their closest counterparts in the training set based on Tanimoto similarity. Scaffold similarity assesses how well the distribution of Bemis-Murcko scaffolds in the generated molecules aligns with that of actual molecules. Finally, KL divergence compares the distributions of various physicochemical properties.

C.2.2 Natural Product Generation Benchmark

While understanding natural products is crucial, their intrinsically complex structures and large molecular sizes pose significant challenges for molecular design. To address this, we designed a natural product generation benchmark to evaluate the ability of generative models to capture and reproduce the biochemical characteristics of natural products.

We first preprocessed the COCONUT dataset (Sorokina et al., 2021; Chandrasekhar et al., 2025) , the most extensive open-access collection of natural products by filtering out molecules, including charges, retaining only natural compounds. We also excluded metal-containing molecules and selected only those composed of the following atom types: B, N, C, O, F, Si, P, S, Cl, Br, I, Se, and As. Additionally, molecules exceeding 99 heavy atoms were removed to avoid arbitrarily large molecules. This processing resulted in a final dataset of 416,249 molecules, which was randomly split into training (85%), validation (5%), and test (10%) sets.

We sampled $10,000$ molecules for benchmarking and evaluated them based on validity, uniqueness, novelty, and Fréchet ChemNet Distance (FCD), following the MOSES benchmarking protocol. However, achieving high validity and uniqueness alone does not guarantee that the generated molecules resemble natural products. To address this, we applied the hierarchical classification scheme from NPClassifier (Kim et al., 2021), which categorizes molecules at three levels: Pathway, representing broad biosynthetic origins with seven categories; Superclass, defining structural groupings within pathways within 70 categories; and Class, providing a finer-grained classification of 672 structural categories. We compute the Kullback-Leibler (KL) divergence across these categorical distributions to assess the alignment between generated molecules and the training dataset. Additionally, since the COCONUT dataset includes NP-likeness scores (Ertl et al., 2008), we also evaluated the KL divergence of the NP-likeness score distribution, quantifying how closely the generated molecules resemble authentic natural products.

D Additional Results

D.1 Results on GuacaMol Dataset

Model	Class	Val. ↑	V.U. ↑	V.U.N. ↑	KL Div. ↑	FCD ↑
Training set	-	100.0	100.0	-	0.0	92.8
NAGVAE (Kwon et al., 2020)	VAE	92.9	88.7	88.7	38.4	0.9
MCTS (Jensen, 2019)	-	**100.0**	**100.0**	95.4	82.2	1.5
DiGress (Vignac et al., 2022)	Atom + Diffusion	85.2	85.2	85.1	92.9	68.0
DisCo (Xu et al., 2024)	Atom + Diffusion	86.6	86.6	86.5	92.6	59.7
Cometh (Siraudin et al., 2024)	Atom + Diffusion	98.9	98.9	<u>97.6</u>	96.7	72.7
DeFoG (# steps = 50) (Qin et al., 2024)	Atom + Flow	91.7	91.7	91.2	92.3	57.9
DeFoG (# steps = 500) (Qin et al., 2024)	Atom + Flow	99.0	<u>99.0</u>	**97.9**	<u>97.7</u>	<u>73.8</u>
FragFM (train fragments, # steps = 500)	Fragment + Flow	<u>99.1</u>	98.9	93.2	**99.6**	**87.2**

Table 3: **Molecule generation on GuacaMol dataset**. The upper part consists of non-denoising models, while the second part consists of iterative denoising methods, including diffusion-based and flow-based methods. The table compares their performance on several metrics.

D.2 Analysis of Coarse-to-fine Autoencoder

Dataset	Train set		Test set	
	Bond acc.	Graph acc.	Bond acc.	Graph acc.
MOSES	99.99	99.96	99.99	99.93
GuacaMol	99.99	99.43	99.98	99.42
Coconut	99.98	97.62	99.71	97.43

Table 4: **Coarse-to-fine autoencoder accuracy.**

To evaluate the performance of the coarse-to-fine autoencoder, we assess the accuracy of bond and graph reconstruction. As shown in Table 4, the model achieves nearly perfect bond and graph recovery on the MOSES and GuacaMol datasets, with bond accuracy exceeding 99%, demonstrating its reliability in preserving the information of atom-level graph. The model also achieves strong recovery performance on the more complex COCONUT dataset, demonstrating its robustness in handling diverse and structurally complicated molecular graphs.

D.3 Sampling Step

A key challenge in generative models based on stochastic processes is the need for multiple iterative refinement steps, which can significantly impact computational efficiency. While increasing the number of denoising steps generally improves the sampling of diffusion and flow-based models, it also extends the time required for sampling, making large-scale generation impractical. To evaluate

# Step	Model	Val. ↑	V.U. ↑	V.U.N. ↑	Filters ↑	FCD ↓	SNN ↑	Scaf ↑
-	Training set	100.0	100.0	0.0	100.0	0.01	0.64	99.1
10	DiGress*	6.3	6.3	6.2	66.4	9.30	0.37	10.7
	Cometh	26.1	26.1	26.0	59.9	7.88	0.36	8.9
	FragFM	**95.8**	**95.7**	**88.3**	**98.7**	**0.66**	**0.54**	**13.0**
50	DiGress*	75.2	75.2	72.1	94.1	1.38	0.51	15.1
	DeFog	83.9	83.8	81.2	96.5	1.87	**0.59**	14.4
	Cometh	82.9	82.9	80.5	94.6	1.54	0.49	**18.4**
	FragFM	**99.4**	**99.2**	**90.4**	**98.8**	**0.57**	0.55	11.9
100	DiGress*	82.8	82.8	79.4	95.1	1.11	0.51	14.7
	Cometh	85.8	85.7	82.9	96.5	1.43	0.50	**17.2**
	FragFM	**99.4**	**99.2**	**90.3**	**98.8**	**0.56**	**0.55**	11.0
300	DiGress*	85.2	85.2	80.9	96.4	1.10	0.52	14.6
	Cometh	86.9	86.9	83.8	97.1	1.44	0.51	**17.8**
	FragFM	**99.5**	**99.4**	**90.7**	**98.9**	**0.57**	**0.54**	11.8
500	DiGress*	84.9	84.9	82.0	94.5	1.35	0.50	**16.5**
	DeFog	92.8	92.7	85.4	**98.9**	1.95	**0.55**	14.4
	Cometh	87.0	86.9	83.8	97.2	1.44	0.51	15.9
	FragFM	**99.6**	**99.5**	**90.7**	**98.9**	**0.56**	0.54	14.6
700	DiGress*	85.4	85.4	82.4	95.2	1.36	0.50	**16.7**
	Cometh	87.2	87.1	83.9	97.2	1.43	0.51	15.9
	FragFM	**99.5**	**99.4**	**90.0**	**98.9**	**0.55**	**0.55**	12.6
1000	DiGress*	84.4	84.4	81.0	96.0	1.35	0.51	14.4
	Cometh	87.2	87.2	84.0	97.2	1.44	0.51	**17.3**
	FragFM	**99.6**	**99.4**	**90.1**	**98.9**	**0.56**	**0.55**	12.5

Table 5: **Performance comparison of denoising graph generative frameworks on the MOSES dataset across varying numbers of sampling steps.** To solely evaluate the effect of sampling steps, all FragFM results are obtained without utilizing target guidance and noise. For DiGress, we re-trained the model using different diffusion timesteps.

this trade-off, we analyzed how different generative models behave under varying step conditions, focusing on their ability to maintain validity, uniqueness, and structural diversity.

As shown in Table 5, reducing the number of denoising steps leads to a general decline in molecular quality across all models. However, the extent of this degradation varies considerably depending on the model architecture. DiGress, for example, suffers a catastrophic performance drop, achieving only 6.3% validity and an FCD of 9.30 at 10 steps, highlighting its heavy reliance on many iterative refinements. Continuous-time models, such as DeFoG and Cometh, exhibit better robustness but still experience a significant decline in performance when operating with a low number of steps.

In contrast, FragFM maintains high validity even with significantly fewer denoising steps. In fact, at just 10 steps, FragFM achieves 95.8% validity while preserving other key MOSES benchmark metrics. This performance surpasses other models, typically requiring at least 50 to 100 steps to reach comparable results. This improvement can be attributed to the fragment-based discrete flow matching approach, which reduces the number of nodes and edges in the molecular graph, lowering the computational complexity of edge prediction. Since molecular fragments inherently capture larger structural motifs, they provide a more structured and stable generative process, allowing for efficient sampling with fewer denoising steps while preserving overall molecular validity.

D.4 Sampling Time Analysis

		MOSES	GuacaMol	COCONUT
Property	Min. nodes	8	2	2
	Max. nodes	27	88	99
	Sampled Molecules	25000	10000	10000
Sampling Time (h)	DiGress (# steps = 500)	3.0	-	12.5
	DeFoG* (# steps = 500)	5	7	-
	FragFM (# steps = 500)	1.1	0.6	1.7
	FragFM (# steps = 50)	0.4	0.1	0.3

Table 6: **Comparison of sampling time across different datasets and methods.** All experiments were performed using a single NVIDIA GeForce RTX 3090 GPU and an Intel Xeon Gold 6234 CPU @ 3.30GHz. *Results for DeFoG are borrowed from the original paper, where experiments were conducted on an NVIDIA A100 GPU

While graph representations exhibit a quadratic increase in edge dimensions as the number of nodes grows, fragment-level graphs contain significantly fewer nodes and edges, resulting in lower computational complexity than atom-level graphs. A detailed analysis of sampling time is provided in Table 6, where FragFM demonstrates the fastest sampling time across all datasets with 500 sampling steps. Furthermore, as shown in Table 5, FragFM achieves comparable or superior performance to baseline models even with just 50 sampling steps. This suggests that its sampling time can be further optimized, enabling speeds $\times 10$ to $\times 30$ times faster than other models while maintaining the generative quality.

D.5 Conditional Generation

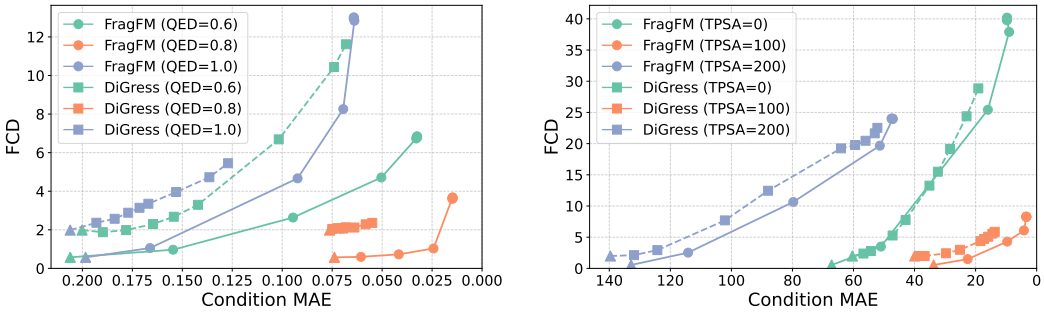

Figure 5: **Condition MAE, FCD curves of QED (left) and TPSA (right) over CFG guidance strengths for MOSES models.**

D.6 VISUALIZATION OF GENERATED MOLECULES

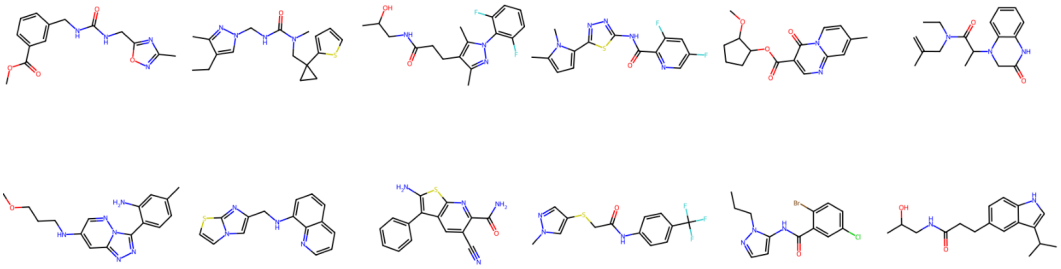

Figure 6: **Randomly selected samples generated by FragFM trained on the MOSES dataset.**

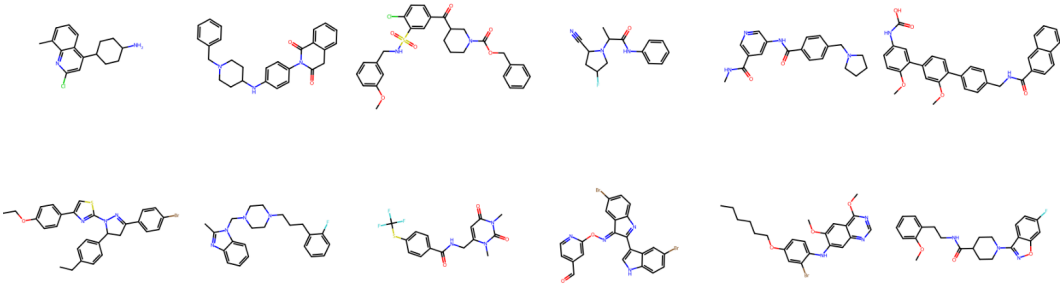

Figure 7: **Randomly selected samples generated by FragFM trained on the GuacaMol dataset.**

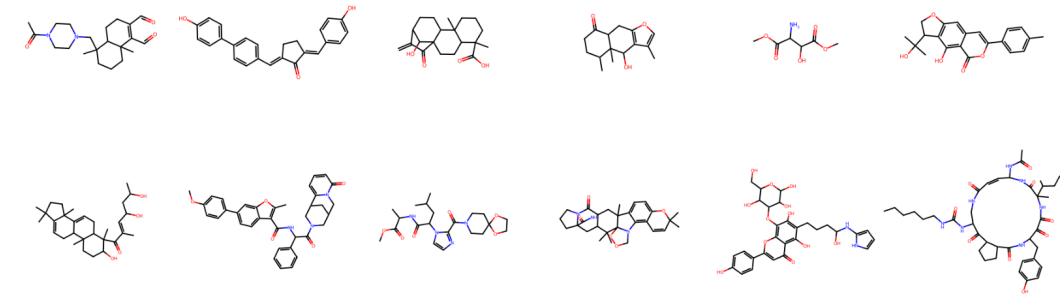

Figure 8: **Randomly selected samples generated by FragFM trained on the COCONUT dataset.** The top row displays molecules with a heavy atom count below 30, while the bottom row includes those with a count between 30 and 60. FragFM can generate complex natural products such as steroids, peptides, and saccharides.

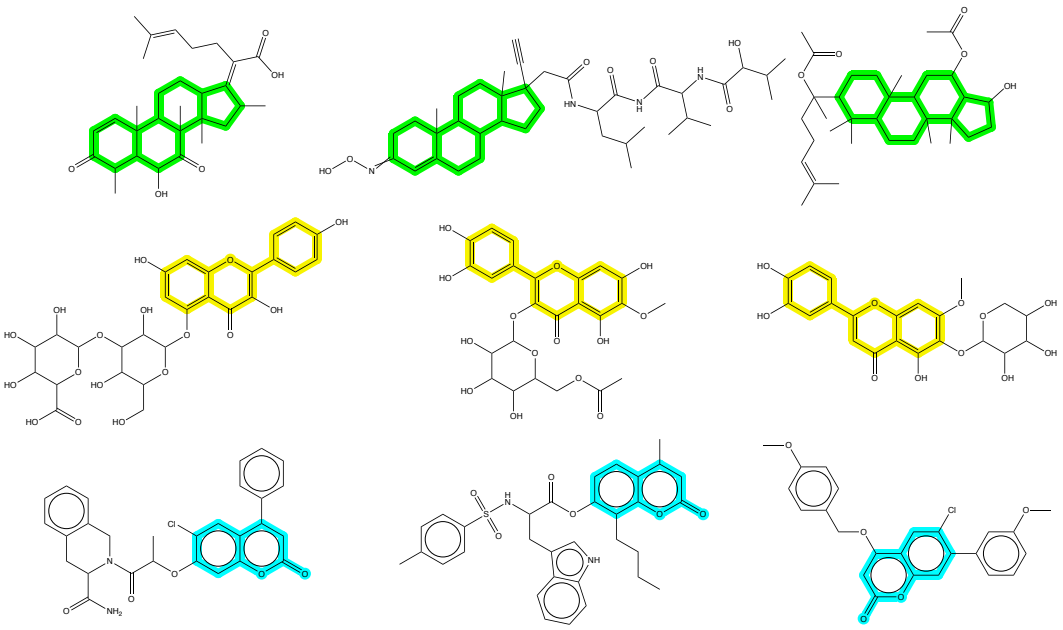

Figure 9: **Cherry-picked samples generated by FragFM trained on COCONUT dataset, with natural product substructures highlighted**. The model successfully generates molecules that are steroid, flavonoid, and coumarin derivatives, showcasing its ability to generate natural product-like molecules successfully.

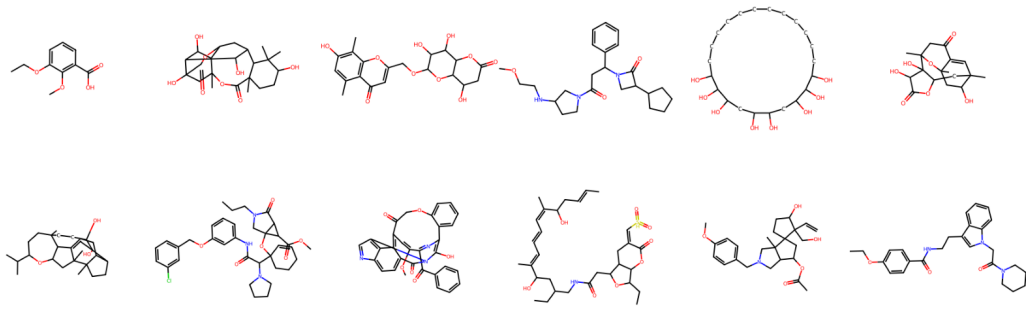

Figure 10: **Randomly selected samples generated by DiGress trained on the COCONUT dataset.** The top row presents molecules with a heavy atom count below 30, while the bottom row includes those with a count between 30 and 60. Atom-based models often struggle to generate chemically stable structures, frequently producing molecules with sterically strained and chemically implausible fused ring systems (second in the top row; first, second, and third in the bottom row).

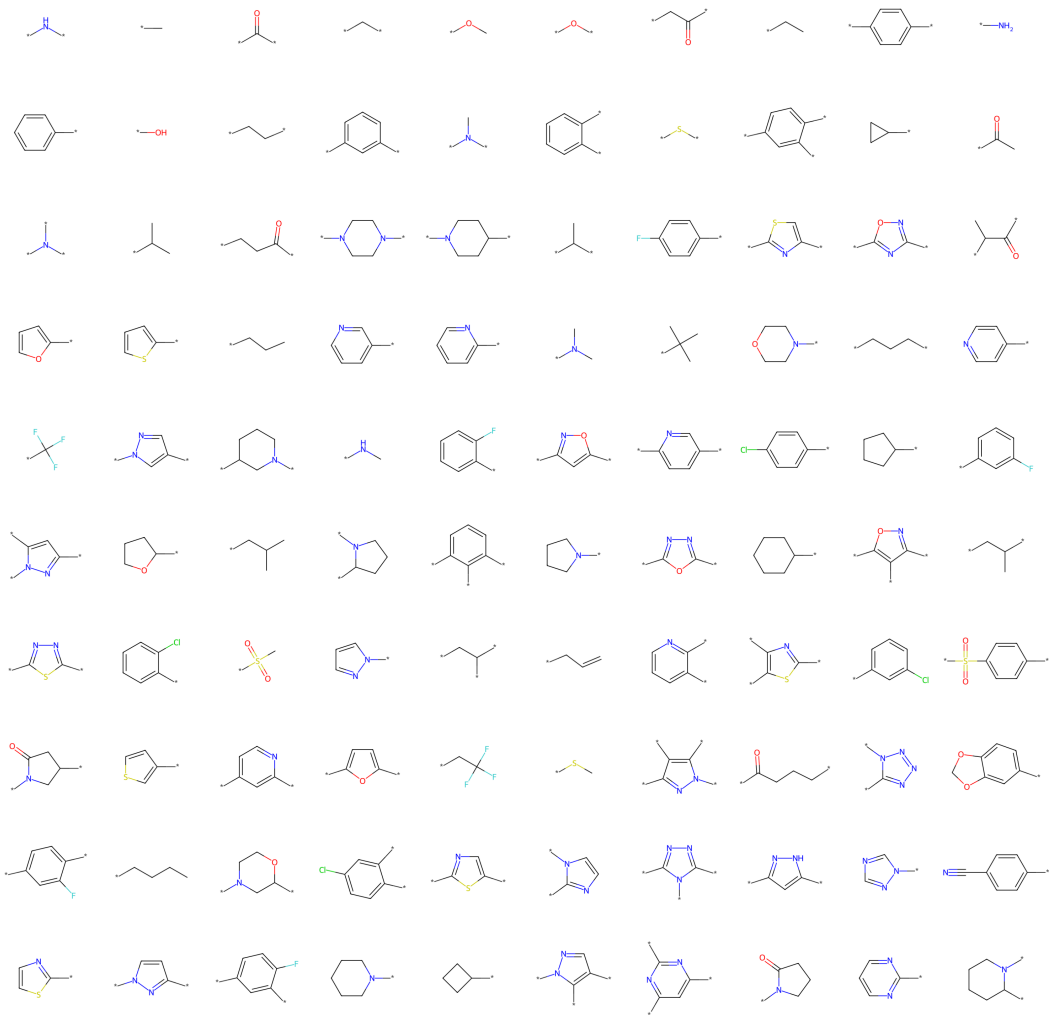

Figure 11: **Top 100 Common fragments extracted from the MOSES dataset using BRICS decomposition.** More frequently occurring fragments are positioned towards the top left.