INSTRUCTION - BASED MOLECULAR GRAPH GENERATION

1. Definition

Molecular graph generation is the task of automatically constructing chemical molecules in the form of graphs, where atoms are represented as nodes and chemical bonds as edges. Unlike traditional generative modeling on continuous data such as images or text, molecular graph generation must adhere to strict structural and chemical validity constraints, including valency rules, bond types, and molecular stability. The objective is not only to produce syntactically valid graphs, but also to generate molecules with desirable properties - such as biological activity, synthesizability, or drug-likeness - based on learned patterns from existing chemical datasets. Because of these requirements, the problem sits at the intersection of graph theory, machine learning, and computational chemistry.

This problem matters because discovering new molecules is a cornerstone challenge in drug design, materials science, and chemical engineering. Conventional molecular discovery relies heavily on human intuition and exhaustive laboratory experimentation, both of which are time-consuming and costly. Automated molecular graph generation provides a data-driven alternative that can explore vast chemical spaces far beyond human capability. By guiding the generation process towards specific biological or physicochemical properties, these models can significantly accelerate early-stage discovery pipelines and reduce development costs. Ultimately, advances in molecular graph generation hold the potential to enable faster drug discovery, design novel materials with tailored properties, and enhance our fundamental understanding of chemical structure–function relationships.

2. Challenges

Molecular graph generation faces several fundamental challenges that arise from the nature of chemical space and the practical requirements of molecular design. One of the most prominent difficulties is the sheer size of the combinatorial search space. The set of all theoretically possible molecules is estimated to be on the order of 10 to the power of 60, far exceeding what can be explored through brute-force enumeration or naive search strategies. As a result,

generative models must navigate an extremely high-dimensional and sparsely populated space, making efficient exploration and targeted generation both computationally demanding and methodologically complex.

Another core challenge lies in balancing chemical validity and structural novelty. Generated molecules must adhere to strict chemical rules, such as valence constraints and allowable bond configurations, to be physically meaningful. At the same time, models are expected to produce novel structures rather than trivial variations of the training data. Achieving both simultaneously is difficult: models that prioritize validity often become conservative and overfit to known molecules, while models that push for novelty may produce unstable or chemically impossible structures. This tension creates a persistent trade-off that current generative techniques struggle to fully overcome.

In real-world applications, molecular generation is inherently a multi-objective optimization problem. Molecules must satisfy a range of criteria - such as absorption, distribution, toxicity, binding affinity, and synthetic accessibility - that frequently conflict with one another. For instance, a candidate molecule may exhibit strong target binding but also display undesirable toxicity or be prohibitively difficult to synthesize. Improving one property can inadvertently degrade another, making it challenging for generative models to balance diverse and sometimes opposing design goals within a single framework.

Finally, evaluating the quality and utility of generated molecules remains an open problem. Widely used metrics - such as validity, uniqueness, and novelty - provide only a partial picture of molecular quality and do not necessarily correlate with practical usefulness in drug discovery or materials design. High benchmark scores may still correspond to molecules that are chemically unrealistic, hard to synthesize, or unlikely to exhibit desirable biological properties. Consequently, the development of more meaningful and application-aware evaluation criteria remains a critical barrier to progress in molecular graph generation.

3. Approaches

Approaches to molecular graph generation can be broadly categorized into three methodological families, each offering different trade-offs between efficiency, chemical validity, and structural controllability.

The first class - *all-at-once generation* - produces an entire molecular graph in a single forward pass of the model. Early examples include GraphVAE and VGAE, while more recent diffusion-based frameworks such as GDSS, DiGress, and Wave-GD have demonstrated improved generative quality through denoising or score-based formulations. This approach is highly computationally efficient and well suited for large-scale sampling, as it allows the model to generate many molecules simultaneously. However, because the entire structure is created holistically, it is difficult to enforce local chemical rules such as valence and bond-type constraints. As a result, these methods often struggle to guarantee chemical validity, especially when generating complex or unconventional molecular structures.

A second family of methods, known as *fragment-based generation*, constructs molecules from predefined chemical substructures, such as rings, scaffolds, or functional groups. Representative models include JT-VAE, MoLeR, PS-VAE, and various GFlowNet-based architectures. By operating on chemically meaningful fragments rather than individual atoms, these methods inherently incorporate structural priors that increase the likelihood of producing valid and realistic molecules. The use of motifs also facilitates the encoding of pharmacophoric or bioisosteric information, making these approaches particularly appealing for drug design tasks. Nevertheless, the reliance on predefined fragment libraries can limit the diversity of generated molecules and may bias the model toward known chemical patterns.

The third class, *node-by-node generation,*builds a molecule incrementally by adding atoms and bonds in a sequential manner. Models such as CGVAE, GraphAF, and GraphDF follow this paradigm, mirroring aspects of step-wise chemical synthesis. This sequential construction allows the model to explicitly control valence, bonding configurations, and intermediate graph validity at each generation step, often resulting in highly valid molecular structures. However, this fine-grained control comes with increased computational complexity, as the model must evaluate and update the graph repeatedly during the generation process. Consequently, node-by-node methods are typically slower and less scalable than fragment-based or all-at-once approaches, making them less suitable for high-throughput molecular exploration despite their strong validity performance.

4. Problem Formulation

The problem of molecular graph generation can be formally understood as learning a mapping from existing molecular data to a generative distribution capable of producing novel, valid, and useful chemical structures. Central to this task is the choice of input representation, as molecules can be encoded in several complementary formats. Common representations include SMILES strings, which provide compact linear text descriptions of molecular structures, and graph-based encodings in which atoms are treated as nodes and chemical bonds as edges. These graph formulations are particularly suitable for graph neural network (GNN) architectures, which can directly learn relational and topological patterns. For tasks requiring spatial fidelity, 3D molecular geometries capture conformational information critical to molecular interactions. Additionally, predefined fingerprints or descriptor vectors offer feature-rich summaries of structural and physicochemical properties. Regardless of the representation chosen, the goal is to learn a model capable of generating new molecular instances that reflect the statistical and chemical patterns present in the training data while extending beyond them in a meaningful way.

The objective function of molecular graph generation is inherently multi-faceted, as the generated molecules must satisfy several quality criteria simultaneously. One key objective is *novelty*, ensuring that the model produces structures that are not merely replicas or trivial modifications of known molecules. Equally important is *chemical validity*, requiring that generated molecules adhere to fundamental rules such as proper valence, permissible bond types, and structural connectivity. Beyond structural correctness, practical applications demand attention to *synthetic accessibility*, as the utility of a molecule depends on its potential to be synthesized using existing laboratory or computational methods. Models must also encourage *diversity*, promoting exploration across the vast chemical space rather than focusing on narrow structural families. Finally, many generative tasks require explicit *property optimization*, where models are guided to produce molecules with favorable characteristics such as low toxicity, high solubility, or desirable bioavailability. Collectively, these objectives define a challenging optimization landscape in which multiple, often competing criteria must be balanced within a unified generative framework.

5. Generation Methodology

Generation methodologies for molecular graph construction span several powerful machine learning paradigms, each offering distinct mechanisms for

learning and producing new chemical structures. *Variational Autoencoders (VAEs)* operate by encoding molecules into a continuous latent space and decoding them back into valid structures, enabling smooth interpolation and controlled exploration of chemical space. *Generative Adversarial Networks (GANs)* employ a generator–discriminator framework in which the generator proposes candidate molecules while the discriminator evaluates their plausibility, driving the model toward generating increasingly realistic structures. *Reinforcement Learning (RL)*-based methods treat molecule construction as a sequential decision-making process, allowing an agent to iteratively modify or assemble molecules while directly optimizing for specific objectives such as toxicity reduction or binding affinity enhancement. More recently, *diffusion models* have emerged as a robust alternative, generating molecular structures by gradually denoising random noise or corrupted graphs, thereby capturing complex structural distributions with high fidelity. Together, these methodologies provide the algorithmic foundation for modern molecular generation systems, each contributing unique strengths to the pursuit of valid, novel, and property-optimized molecules.

6. Evaluation

Evaluating molecular graph generation models requires a set of statistical metrics that capture both the structural quality and exploratory capability of the generated molecules. A fundamental metric is *validity*, which measures the proportion of generated molecules that satisfy chemical rules such as proper valence, permissible bond types, and structural connectivity. High validity indicates that a model is capable of producing chemically meaningful structures that reside within realistic chemical space. Complementing this is *novelty,* defined as the fraction of generated molecules that do not appear in the training dataset. Novelty assesses the model's ability to explore new regions of chemical space rather than merely replicating known molecules, which is essential for applications such as drug discovery where unseen structural motifs may lead to improved biological activity.

Another important metric is *uniqueness,* which quantifies the proportion of distinct molecules among the valid outputs. This metric ensures that the generative model is not producing redundant samples and is instead offering a broad range of potential candidates. In addition to these measures, *diversity* provides a more global assessment of how broadly the model spans chemical space. Diversity is often computed using similarity measures such as the

Tanimoto distance between molecular fingerprints and reflects the structural variation within the generated set. High diversity indicates that the model is capable of covering a wide spectrum of chemical structures rather than concentrating on closely related compounds. Together, these metrics offer a comprehensive framework for assessing the performance of molecular generation models, balancing structural correctness, creativity, and breadth of exploration.

Model Architecture

The overall molecular generation framework is implemented as a cascaded, two-stage process. The first stage, based on Momentum Contrastive Learning, learns a compact and meaningful latent representation of the molecule. The second stage, a Diffusion Transformer, operates in this learned latent space to generate novel molecular representations conditioned on text.

1. **Momentum Contrastive Learning (LDMol Autoencoder)**

The LDMol Autoencoder is designed for **self-supervised learning**, meaning it learns molecular features without explicit human-labeled targets. Its core function is to establish a chemically meaningful, compressed vector representation, known as a **latent code**, for molecular structures input as **SMILES** strings—a standardized, linear textual notation used to represent molecular structure.

Model Components and Function

The architecture is dual-purpose, serving as both a reconstructive autoencoder and a contrastive learning model. The main feature extractor is the **Primary Encoder**, implemented as a **BERT Transformer**—a powerful neural network adapted here to process the sequential nature of the SMILES strings. Its role is to take the tokenized SMILES sequence and produce a high-dimensional feature vector. This vector is then compressed into the final **Latent Space** (a continuous vector space of dimension 512) by the **Projection Head**, a simple linear layer. The **Primary Decoder** works in tandem, attempting to reconstruct the original SMILES string from the compressed latent code, ensuring the latent vector is information-rich and reconstructible.

For the **Momentum Contrastive Learning** (MoCo) task, the model uses two critical additions: the **Momentum Encoder** and **Momentum Projection Head**. These are identical to the primary networks but are designated as the **stable target**. Their weights are not updated via normal training gradients; instead, they evolve very slowly, providing a consistent reference. Finally, a large **Negative Sample Queue** is used as a substantial memory bank (size 16,384) to store past representations, significantly enriching the contrastive signal by providing a large pool of negative examples.

Overall Training and Architecture Flow

The training minimizes a **composite loss function** that judiciously combines a **Contrastive Loss** (for feature discriminability) and a **Decoding Loss** (for feature completeness), with the decoding component weighted at 0.4. The training process involves generating two augmented views of the same molecule, creating a positive pair. The **Primary Encoder** processes one view to create a *query* vector, while the **Momentum Encoder** processes the other to create a *positive key* vector. The contrastive loss then works to maximize the similarity between the query and its positive key, while simultaneously minimizing the similarity between the query and all the negative keys stored in the queue. This forces the latent space to cluster chemically similar molecules together. The parameters of the primary components are updated normally, while the weights of the momentum components are updated using an **Exponential Moving Average (EMA)**, which dictates their slow update rate, maintaining the stability required for the target network.

Hyperparameters and Optimization

Optimization for this stage is performed using the **AdamW** algorithm, which is an advanced variant of stochastic gradient descent well-suited for Transformer models due to its effective mechanism for weight decay regularization. The **Learning Rate Scheduler** employs a **cosine schedule with a warmup phase**. The warmup linearly increases the learning rate during the initial phase of training to prevent instability, after which the rate smoothly decays following a cosine curve for stable final convergence. The **Momentum Coefficient** is a critical parameter that governs how quickly the weights of the stable target network (Momentum Encoder) track those of the primary network; a high value indicates a very slow, stabilizing update. The **Contrastive Weight** determines the relative importance of the reconstructive decoding loss compared

to the primary contrastive loss in the final objective function. The **Negative Sample Queue Size** explicitly controls the historical memory size, determining the number of dissimilar molecular representations available for the contrastive task.

2. Diffusion Transformer (DiT) Model

The DiT model is the generative component, performing iterative noise prediction within the **Latent Space** defined by Stage 1. This framework is known as a **Latent Diffusion Model**, which performs the computationally intensive diffusion process in a compressed space rather than the original high-dimensional data space.

Model Components and Function

The core of this stage is the **Diffusion Transformer (DiT)**, which acts as the **noise prediction network**. This large-scale Transformer's sole function is to predict the noise component added to a noisy latent code, allowing for iterative denoising.

The input (a noisy latent vector) is prepared by an **Input Projection** layer, matching its size to the Transformer's internal hidden size. The temporal condition is handled by the **Timestep Embedder**, which processes the scalar diffusion time step (which indicates the current level of noise) into a rich vector representation. The text condition is provided by a separate, frozen **Text Encoder**, which extracts a stable feature vector from the molecular description.

The DiT's capacity for conditional generation stems from its **Conditional DiT Blocks**, which implement **Adaptive Layer Normalization (AdaLN)**. AdaLN is a sophisticated mechanism that strategically uses the external conditions (time step and text embedding) to dynamically modulate the internal activation statistics (scaling and shifting) within the Transformer layers, ensuring the noise prediction is precisely guided by the input text and the current position in the denoising sequence.

Overall Training and Architecture Flow

During Stage 2 training, the **Stage 1 LDMol Autoencoder is frozen** and used solely for converting molecules to latent codes. The training involves corrupting a clean latent code with noise over a total of 1,000 steps according to

a **linear schedule**. The DiT model is trained to minimize the difference between its predicted noise and the true noise added to the latent representation.

To enhance the fidelity of text-to-molecule generation, the technique of **Classifier-Free Guidance (CFG)** is employed. This is implemented by randomly setting the text condition to null (meaning *unconditional* generation) for a fraction of the training samples. This forces the model to learn both conditional (text-guided) and unconditional distributions simultaneously, which can be leveraged during inference to achieve a strong balance between diversity and adherence to the text prompt.