Link to the paper: https://arxiv.org/abs/2401.13858

Link on the Kaggle notebook: (The experiment is done on the QM9 dataset)
https://www.kaggle.com/code/phoenix301123/graph-diffusion-transformer-for-multi-conditional

Problem Definition

Inverse Molecular Design: automatically generate novel molecules that satisfy a specific set of desired chemical or physical properties.

This notebook focuses on a **Multi-Conditional** approach, specifically using a single molecular property (U0, the atomization energy at 0K) from the **QM9 dataset** as the conditioning variable.

- **Input**: A desired molecular property value (the condition, c) and a random noise tensor (the initial graph state, xT).
- **Output**: A valid molecular graph structure (nodes representing atoms, edges representing bonds, x0) whose property is close to the target condition c.
- **Method**: A **Denoising Diffusion Probabilistic Model (DDPM)** adapted for graph data, where a **Transformer** architecture (GraphDiT) is used as the **denoising model**.

Diffusion Model on Graph Data:
- **Forward Process (Noise Injection):** Gradually adds Gaussian noise to the initial graph features x0 over T timesteps, resulting in a noisy graph state (x t). The node features are treated as continuous vectors during this process.
- **Reverse Process (Denoising/Generation)**: A neural network is trained to predict the noise added at any time step t, conditioned on t and the target property c. By iteratively subtracting the predicted noise, the model can sample a new data point x0 from pure noise xT.
- **Graph Adaptation:** The GraphDiT uses a standard Transformer architecture, which is inherently designed for sequential/set data,

to process the node features of the graph. The sparse graph structure is converted into a dense batch format with attention masking to handle variable-sized molecules.

Model Architecture:

- Node Embedding (node_embed): A simple linear layer that maps the 11-dimensional noisy input node features (x_noisy) into the internal hidden dimension (HIDDEN_DIM = 128)
- Time Embedding (time_embed): The Sinusoidal Position Embeddings layer encodes the discrete diffusion time step t into a high-dimensional vector, which is crucial for the model to know how much noise is currently present in the input.
- Condition Encoder (cond_encoder): The Condition Encoder is an MLP that transforms the single, normalized property (e.g., U0 energy) into a Hidden Dimensional vector. This is the mechanism for conditional generation.

Graph Transformer Core:

The model uses a stack of **four** GraphDiTBlock instances, which adapt the standard Transformer for graph data:

- **Graph-to-Dense Conversion:** Before processing, the sparse graph node features (x) are converted into a dense batch tensor (x-dense) using the torch_geometric.utils.to_dense_batch function. This conversion involves padding the graphs up to a maximum atom count of nine and generating a corresponding batch mask.

- **Self-Attention:** The block utilizes a standard Multihead Attention mechanism. To handle the variable-sized graphs correctly, the batch mask is critically applied as the key-padding-mask. This prevents the padded, non-existent nodes from participating in the attention calculation.

- **Feature-wise Linear Modulation (FiLM) Conditioning:** This is the core mechanism for incorporating the time and condition information. The model first computes the joint condition embedding (c-joint) by summing the time embedding (t-emb) and the condition embedding (c-emb). The GraphDiTBlock then uses two linear layers (film-gamma and film-beta) on this joint embedding to compute the scaling parameter (gamma) and the shifting parameter (beta). These parameters modulate the node features before both the self-attention and the feed-forward steps. The mathematical effect is to scale and shift the features:

$$\mathbf{x}_{\text{modulated}} = \mathbf{x} \odot (1 + \gamma) + \beta$$

- **Output:** After the stack of blocks, the final features are projected back to the original dimension (11) via final_norm and final_projection, yielding the predicted noise