molecule-cell-gen

Installation:

- Follow the instructions in TopoModelX

GDSS Overview:

Graph Diffusion Process

A graph G with N nodes is defined by its node features $X \in \mathbb{R}^{N \times F}$ and the weighted adjacency matrix $A \in \mathbb{R}^{N \times N}$ as $G = (X_0, A_0) \in \mathbb{R}^{N \times F} \times \mathbb{R}^{N \times N} := \mathcal{G}$, where F is the dimension of the node features. The diffusion process $\{\boldsymbol{G}_t = (\boldsymbol{X}_t, \boldsymbol{A}_t)\}_{t \in [0,T]}$ transforms the node features and the adjacency matrices to a simple noise distribution. We can model the diffusion process by the following Ito SDE

$$\mathrm{d}\boldsymbol{G}_t = \mathbf{f}_t(\boldsymbol{G}_t)\mathrm{d}t + \mathbf{g}_t(\boldsymbol{G}_t)\mathrm{d}\mathbf{w}, \quad \boldsymbol{G}_0 \sim p_{\text{data}}\,, \tag{1}$$

where $\mathbf{f}_t(\cdot) : \mathcal{G} \to \mathcal{G}^1$ is the linear drift coefficient, $\mathbf{g}_t(\cdot) : \mathcal{G} \to \mathcal{G} \times \mathcal{G}$ is the diffusion coefficient, and $\mathbf{w}$ is the standard Wiener process. The reverse of the diffusion process in time is also a diffusion process described by the SDE:

$$\mathrm{d}\boldsymbol{G}_t = \left[\mathbf{f}_t(\boldsymbol{G}_t) - g_t^2 \nabla_{\boldsymbol{G}_t} \log p_t(\boldsymbol{G}_t)\right]\mathrm{d}\bar{t} + g_t \, \mathrm{d}\overline{\mathbf{w}} \tag{2}$$

where p_t denotes the marginal distribution under the forward diffusion process at time t, $\bar{\mathbf{w}}$ is a reverse-time standard Wiener process, and $\mathrm{d}\bar{t}$ is an infinitesimal negative time step.

Solving **Eq. (2)** requires the estimation of $\nabla_{\boldsymbol{G}_t} \log p_t(\boldsymbol{G}_t) \in \mathbb{R}^{N \times F} \times \mathbb{R}^{N \times N}$ which is expensive to compute. So, the paper proposes a new reverse-time diffusion process.

$$\begin{cases} \mathrm{d}\boldsymbol{X}_t = \left[\mathbf{f}_{1,t}(\boldsymbol{X}_t) - g_{1,t}^2 \nabla_{\boldsymbol{X}_t} \log p_t(\boldsymbol{X}_t, \boldsymbol{A}_t)\right]\mathrm{d}\bar{t} + g_{1,t} \, \mathrm{d}\overline{\mathbf{w}}_1 \\ \mathrm{d}\boldsymbol{A}_t = \left[\mathbf{f}_{2,t}(\boldsymbol{A}_t) - g_{2,t}^2 \nabla_{\boldsymbol{A}_t} \log p_t(\boldsymbol{X}_t, \boldsymbol{A}_t)\right]\mathrm{d}\bar{t} + g_{2,t} \, \mathrm{d}\overline{\mathbf{w}}_2 \end{cases} \tag{3}$$

where $\mathbf{f}_{1,t}$ and $\mathbf{f}_{2,t}$ are linear drift coefficients satisfying $\mathbf{f}_t(\boldsymbol{X}, \boldsymbol{A}) = (\mathbf{f}_{1,t}(\boldsymbol{X}), \mathbf{f}_{2,t}(\boldsymbol{A}))$, $g_{1,t}$ and $g_{2,t}$ are scalar diffusion coefficients, and $\overline{\mathbf{w}}_1, \overline{\mathbf{w}}_2$ are reverse-time standard Wiener Process.

Estimating the Partial Score Functions

Training Objectives

Note that the diffusion processes of X_0, A_0 in the system are dependent on each other, related by the gradients of the joint log-density $\nabla_{\boldsymbol{X}_t} \log p_t(\boldsymbol{X}_t, \boldsymbol{A}_t)$ and $\nabla_{\boldsymbol{A}_t} \log p_t(\boldsymbol{X}_t, \boldsymbol{A}_t)$, which are the **partial score functions**. The partial score functions can be estimated by training the time-dependent score-based models $\boldsymbol{s}_{\theta,t}$ and $\boldsymbol{s}_{\phi,t}$, so that $\boldsymbol{s}_{\theta,t}(\boldsymbol{G}_t) \approx \nabla_{\boldsymbol{X}_t} \log p_t(\boldsymbol{G}_t)$ and $\boldsymbol{s}_{\phi,t}$, so that $\boldsymbol{s}_{\phi,t}(\boldsymbol{G}_t) \approx \nabla_{\boldsymbol{A}_t} \log p_t(\boldsymbol{G}_t)$.

The paper proposes a novel training objective

$$\begin{aligned}\min_{\theta} \mathbb{E}_t\Big\{\lambda_1(t)\mathbb{E}_{\boldsymbol{G}_0}\mathbb{E}_{\boldsymbol{G}_t|\boldsymbol{G}_0}\|\boldsymbol{s}_{\theta,t}(\boldsymbol{G}_t) - \nabla_{\boldsymbol{X}_t}\log p_{0t}(\boldsymbol{X}_t \mid \boldsymbol{X}_0)\|_2^2\Big\}\\ \min_{\phi} \mathbb{E}_t\Big\{\lambda_2(t)\mathbb{E}_{\boldsymbol{G}_0}\mathbb{E}_{\boldsymbol{G}_t|\boldsymbol{G}_0}\|\boldsymbol{s}_{\phi,t}(\boldsymbol{G}_t) - \nabla_{\boldsymbol{A}_t}\log p_{0t}(\boldsymbol{A}_t \mid \boldsymbol{A}_0)\|_2^2\Big\}\end{aligned} \tag{4}$$

wherewhere $\lambda_1(t)$ and $\lambda_2(t)$ are positive weighting functions and t is uniformly sampled from $[0, T]$. The expectations in Eq. (4) can be efficiently computed using the Monte Carlo estimate with the samples $(t, \boldsymbol{G}_0, \boldsymbol{G}_t)$.

Then, the paper proposes new architectures for the time-dependent score-based models that can capture the dependencies of $\boldsymbol{X}_t$ and $\boldsymbol{A}_t$ through time, based on GNNs. The score-based model $\boldsymbol{s}_{\phi,t}$ used to estimate $\nabla_{\boldsymbol{A}_t}\log p_t(\boldsymbol{X}_t, \boldsymbol{A}_t)$ is defined by

Permutation-Equivariant Score-based Model

$$\boldsymbol{s}_{\phi,t}(\boldsymbol{G}_t) = \mathrm{MLP}\Big(\Big[\{\mathrm{GMH}(\boldsymbol{H}_i, \boldsymbol{A}_t^p)\}_{i=0,p=1}^{K,P}\Big]\Big), \tag{5}$$

where $\boldsymbol{A}_t^p$ are the higher-order adjacency matrices, $\boldsymbol{H}_{i+1} = \mathrm{GNN}(\boldsymbol{H}_i, \boldsymbol{A}_t)$ with $\boldsymbol{H}_0 = \boldsymbol{X}_t$ given, $[.]$ denotes the concatenation operation, GMH denotes the graph multi-head attention block, and K denotes the number of GMH layers. And the score-based model $\boldsymbol{s}_{\theta,t}$ to estimate $\nabla_{\boldsymbol{X}_t}\log p_t(\boldsymbol{X}_t, \boldsymbol{A}_t)$ is defined as

$$\boldsymbol{s}_{\theta,t}(\boldsymbol{G}_t) = \mathrm{MLP}\Big(\Big[\{\boldsymbol{H}_i\}_{i=0}^{L}\Big]\Big), \tag{6}$$

where $\boldsymbol{H}_{i+1} = \mathrm{GNN}(\boldsymbol{H}_i, \boldsymbol{A}_t)$ with $\boldsymbol{H}_0 = \boldsymbol{X}_t$ given and L denotes the number of GNN layers. Since the message-passing operations of GNNs and the attention function used in GMH are permutation-equivariant, the score-based models are also equivariant, and the log-likelihood implicitly defined by the models is guaranteed to be permutation-invariant.

Solving the system of Reverse-time SDEs

Then, we simulate the system of reverse o solve the system of reverse-time SDEs

Main tasks (order by priority)

Intitution: Improve the GDSS model by considering molecules as cellular complexes, adding more atomic features and ring features. We want to know if topological deep learning can improve traditional graph representation learning.

- Update code to lift molecular graph $G : (X_0, A_0)$ to cellular complexes implemented by TopomodelX
  - Find the right place to conduct cellular lifting of the *GDSS* code repo.
  - Once we have a NetworkX graph, we can easily convert it to a CC object
  - Extract rings from the graph and add it to the CC object
    - Current Solution: use the graph-tool package to extract rings and add them to CC objects
    - Difficulty: the graph-tool package is not compatible with TMX now; this is a potential problem during training
- Design two new time-dependent score-based models based on cell attention layers to estimate $\nabla_{X_t}\log p_{0t}(G_t)$ and $\nabla_{A_t}\log p_{0t}(G_t)$
  - The cell attention network takes $X_0, X_1, A_0, A_{\uparrow,1}$ and $A_{\downarrow,1}$, which can be extracted from the CC object converted from the molecular graph
  - The *CAN* implemented in TopoModelX is designed for classification; we need to slightly update the model structure, including adding an MLP, to estimate a matrix with the same shapes X_0 and A_0

    - Validate our new cell attention layer with Prof. Derr
    - Implement the new score functions in *ScoreNetwork_A.py* and *ScoreNetwrok_X.py*
- Adding more atomic features to $\boldsymbol{X}_0$ before doing the diffusion process
    - Currently, the atom feature is just an array including the atomic numbers, e.g C:6
    - Discuss with Lance to select the feature candidates derived from SMILEs
    - Embed the feature extraction method in *smile_to_graph.py*
- Since rank-2 cells (rings) are added, we should consider assigning attributes to them with the help of biological knowledge. (Optional)
    - With this new feature, we have to update the score function again. Thus, we should consider this as a later task