Improving Discrete Diffusion on Graph Generation

Yiming Qin Manuel Madeira Clément Vignac Dorina Thanou Pascal Frossard

LTS4, EPFL, Switzerland

Graph diffusion models - DiGress [1]

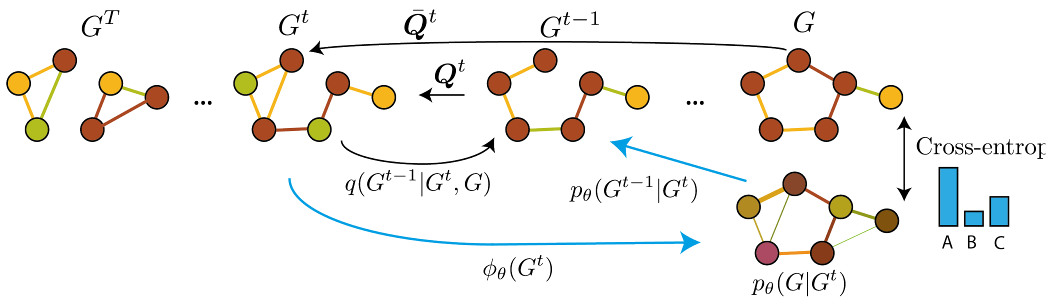
Motivation

Build a **discrete diffusion** model to leverage the inherent **discrete nature of graph** structures.

Approach

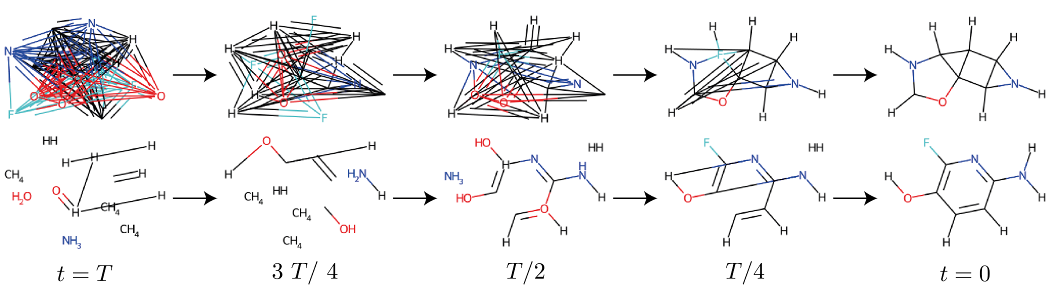
Discrete diffusion

- Transition probability is defined through the transition matrix $\mathbf{Q}_t$ and $\bar{\mathbf{Q}}_t = \mathbf{Q}_1 \cdots \mathbf{Q}_t$
- Adding noise corresponds to sample from a categorical distribution $q(z_t|z_{t-1}) = z_{t-1}\mathbf{Q}_t$
- Posterior distribution: $q(z_{t-1}|z_t, x) = z_t \mathbf{Q}_t^\top \odot x\bar{\mathbf{Q}}_{t-1}$



Discrete diffusion for graphs - DiGress

- Attributed graph with a and b node/edge classes:
$$G = (X \in \mathbb{R}^{n\times a}, E \in \mathbb{R}^{n\times n\times(b+1)})$$
- Add graph noise: $q(G^t|G^{t-1}) = (\mathbf{X}^{t-1}\mathbf{Q}_X^t, \mathbf{E}^{t-1}\mathbf{Q}_E^t)$
- Equivariant model: equivariant architecture + invariant loss
$$l(\hat{p}^G, G) = \sum_{1\le i\le n} \mathrm{CE}(x_i, \hat{p}_i^X) + \lambda \sum_{1\le i,j\le n} \mathrm{CE}(e_{ij}, \hat{p}_{ij}^E)$$
- Add extra features to overcome GNN expressive limit
- Promote sparsity with *marginal* noise model

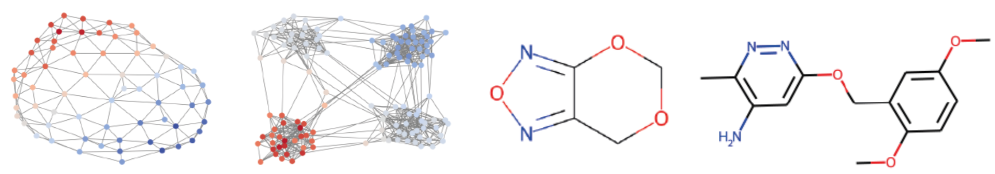


Denoising chain with uniform (top) /marginal (bottom) noise

Denoising process

- Sample the number of nodes train distribution
- Iterate over T diffusion steps to predict a clean graph

Results

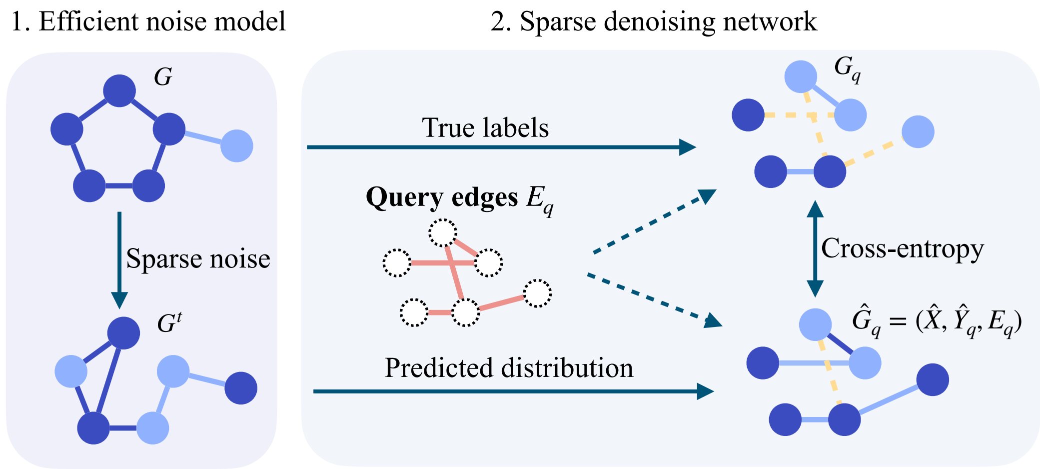


Generated graphs with DiGress

Large graph generation - SparseDiff [2]

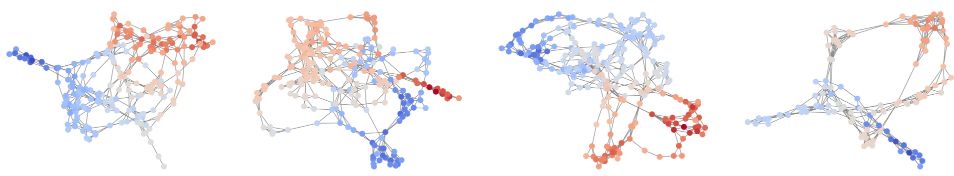
Motivation

Overcome high space complexity caused by edge encodings and enable **large graph generation**.



SparseDiff uses off-the-shell edge list representation
$$G = (\mathbf{E} \in \mathbb{N}^{2\times m}, \mathbf{X} \in \{0,1\}^{n\times a}, \mathbf{Y} \in \{0,1\}^{m\times b})$$
Its *efficient* training consists of 2 components:

- Noise model that preserves the sparsity of noisy graphs
- Sparse denoising network trained on a random subset of node pairs



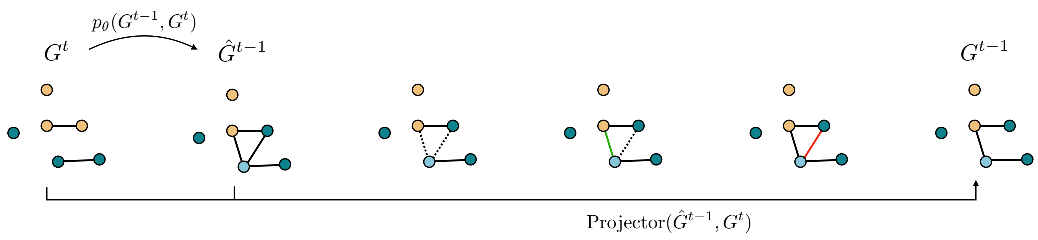
Constrained graph generation - ConStruct [3]

Motivation

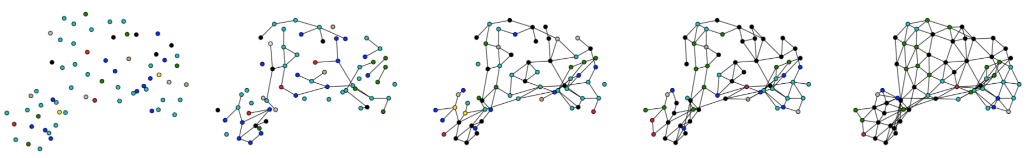
Hard-constrain graph discrete diffusion models using graph **structural properties**.

ConStruct preserves the forward and reverse process in the constrained domain:

- **Forward** - not learnable, so designed to preserve *edge-deletion invariant* properties
$$\mathbf{Q}_X^t = \alpha^t \mathbf{I} + (1-\alpha^t)\mathbf{1}_b \mathbf{m}'_X$$
$$\mathbf{Q}_E^t = \alpha^t_{\mathrm{ABS}} \mathbf{I} + (1-\alpha^t_{\mathrm{ABS}})\mathbf{1}_c \mathbf{e}'_E$$
- **Reverse** - edge insertion process due to forward, but requires *projector* to refuse constraint violating edges



- Improved 80% in validity in digital pathology setting



References

[1] Vignac et al., DiGress: Discrete Denoising diffusion for graph generation, 2023

[2] Qin et al., Sparse Training of Discrete Diffusion Models for Graph Generation, 2023

[3] Madeira et al., Generative Modelling of Structurally Constrained Graphs, 2024