A Survey on Deep Graph Generation: Methods and Applications

Yanqiao Zhu[1,*] Yuanqi Du[2,*] Yinkai Wang[3,*] Yichen Xu[4] Jieyu Zhang[5] Qiang Liu[6] Shu Wu[6]

[1]UCLA [2]Cornell [3]Tufts [4]EPFL [5]UW [6]CASIA *Equal contribution

TL;DR

This paper presents a comprehensive review of deep graph generation models, discussing algorithm taxonomy, applications, challenges, and future research directions.

Overview

Background

Graph generation techniques have been widely used in various fields, such as drug discovery and chemical science. Traditional methods for graph generation compute hand-crafted statistical features of existing graphs and generate new graphs with similar features. However, these methods oversimplify the underlying distributions of graphs and are not capable of capturing complex graph distributions.

Related Problems in Graph Learning

Link prediction aims to predict missing links between nodes.

Graph structure learning simultaneously learns an optimized graph structure and representations for downstream tasks.

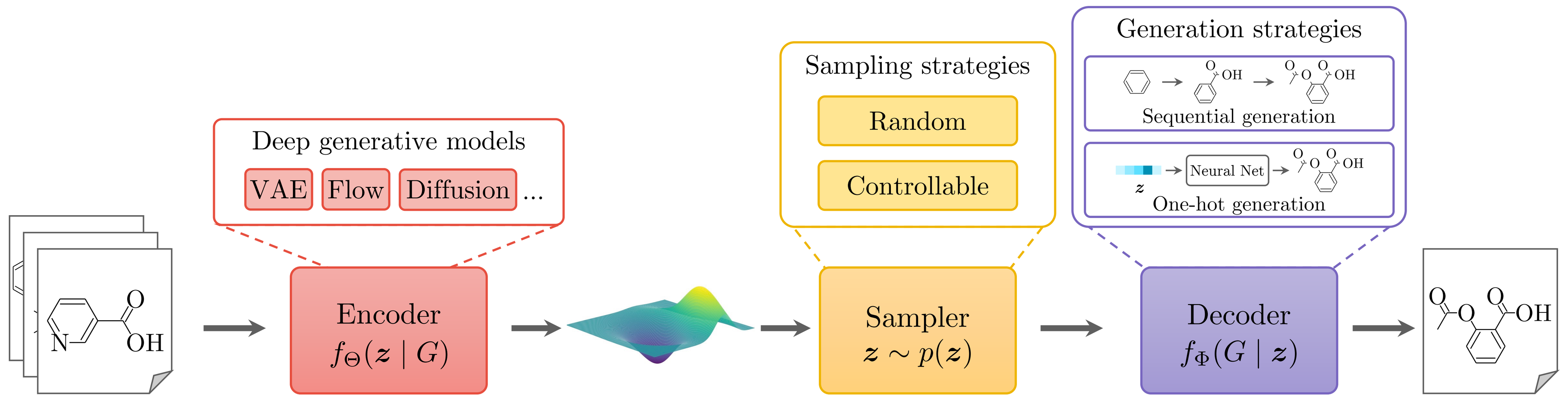
Generative sampling learns to generate subsets of nodes and edges from a large graph.

Set generation is similar to graph generation in that it seeks to generate set objects, such as point clouds or 3D molecules.

Problem Definition

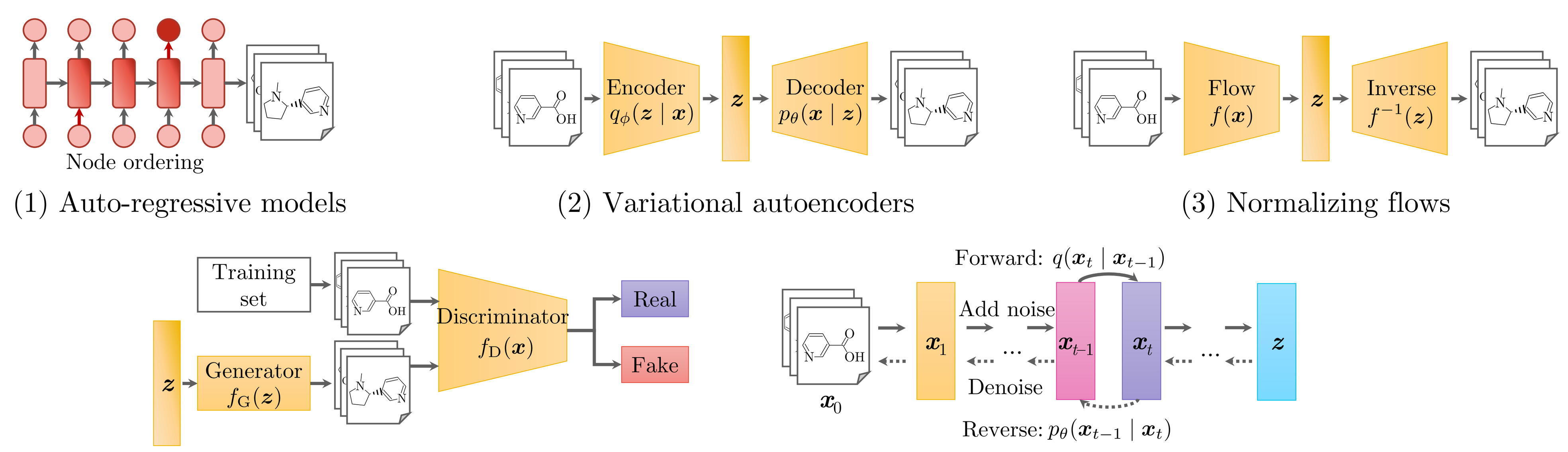
Given a set of M observed graphs $\mathcal{G} = \{G_i\}_{i=1}^{M}$, graph generation learns the distribution of these graphs $p(\mathcal{G})$, from which new graphs can be sampled $G_{\text{new}} \sim p(\mathcal{G})$.

Algorithm Taxonomy



We present an encoder–sampler–decoder pipeline to characterize existing graph generative models. The encoder maps observed graphs into continuous vectors and outputs the parameters of a stochastic distribution, which is used to sample latent representations and restore them to graph structures through a decoder.

Deep Generative Models



(1) Auto-regressive models (2) Variational autoencoders (3) Normalizing flows (4) Generative adversarial networks (5) Diffusion models

1. Auto-regressive models (AR) factorize a joint distribution over several random variables via the chain rule of probability.
2. Variational autoencoders (VAE) estimate the distributions of graphs by maximizing the evidence lower bound, which is a lower bound on the log likelihood of the data.
3. Normalizing flows (NF) estimates the density of graphs with an invertible and deterministic mapping between latent variables and graphs.
4. Generative adversarial networks (GAN) consist of a generator and a discriminator, where the generator generates realistic graphs and the discriminator distinguishes between synthetic and real graphs.
5. Diffusion models contain two processes. The forward diffusion process constantly adds noise to the data sample, while the reverse diffusion process recreates the true data sample from a Gaussian noise input.

Sampling Strategies

Random sampling simply draws latent samples from the prior distribution, whereas controllable sampling generates new graphs with desired properties.

- Disentangled sampling factorizes the latent vector into dimensions that each focus on a specific property.
- Conditional sampling introduces a conditional code that explicitly controls the property of the generated graphs.
- Traverse-based sampling searches over the latent space to obtain a latent code with specific properties, or uses heuristics to control the property of the generated graphs.

Generation Strategies

One-shot generation usually generates an adjacency matrix with optional node and edge features in one single step.

Sequential generation generates a graph consecutively in a few steps. As there is no ordering naturally defined for graphs, it has to follow a certain ordering of nodes for the generation.

Applications

- Molecule/protein design: generates syntactically valid molecules/protein graphs with certain properties.
- Program synthesis: generates programs represented in graphs with desired output.

Challenges and Opportunities

Evaluation pipeline

Graph properties/rules design

Diverse graph types

Scalability

Interpretability