#GNN Application(GCF)

——[KDD20] MoFlow: An Invertible Flow Model for Generating Molecular Graphs

Chengxi Zang, Fei Wang

Department of Population Health Sciences, Weill Cornell Medicine

Outline

- Introduction
- Related work
  - Molecular Generation
  - Flow-based Models
- Preliminary
  - The flow framework
  - Invertible affine coupling layers
  - Splitting Dimensions
  - Numerical stability by actnorm
- Proposed MoFlow Model
- Experiments

Introduction

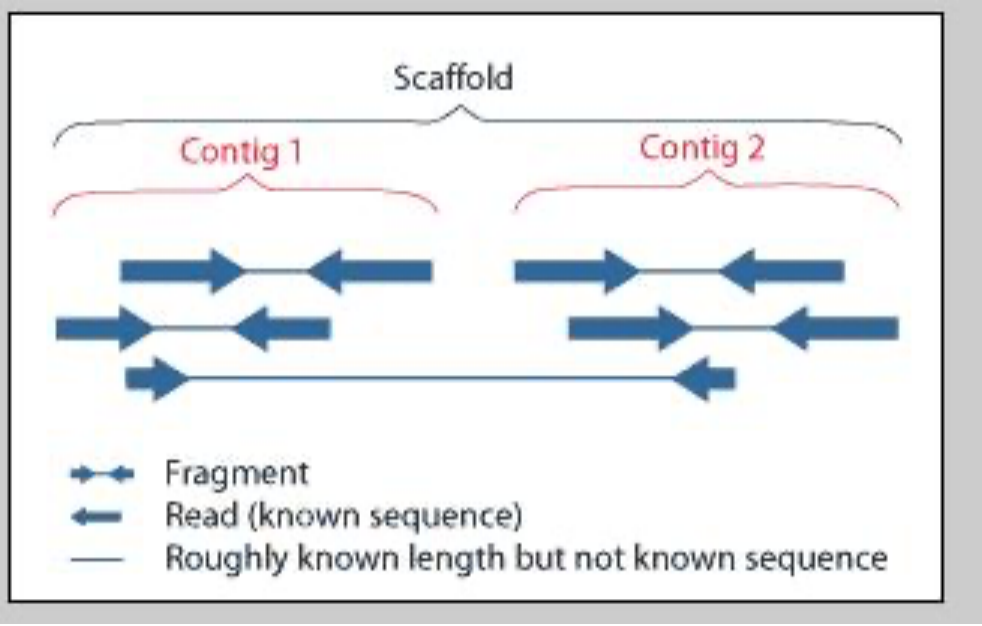
Drug discovery aims at finding candidate molecules with desired chemical properties for clinical trials, which is a long (10-20 years) and costly ($0.5-$2.6 billion) process with a high failure rate [1, 24].

a) the scale of the chemical space of drug-like compounds is 10^{60} [21] but the scale of possibly generated molecular graphs by existing methods are much smaller, and b) generating molecular graphs that have both multi-type nodes and edges and follow bond-valence constraints is a hard combinatorial task.

Related work

Molecular Generation

Different deep generative frameworks are proposed for generating molecular **SMILES** or molecular graphs. Among the variational autoencoder (VAE)-based models, the **JT-VAE** generates valid tree-structured molecules by first generating a tree-structured **scaffold** of chemical substructures and then assembling substructures according to the generated **scaffold**. The **MolGAN** is a generative adversarial networks (GAN)-based model but shows very limited performance in generating valid and unique molecules.



Related work

Flow-based Models

The (normalizing) flow-based models try to learn mappings between **complex distributions** and **simple prior distributions** through **invertible neural networks** and such a framework has good merits of exact and tractable likelihood estimation for training, efficient one-pass inference and sampling, invertible mapping and thus reconstructing all the training data etc.

Examples include **NICE**, **RealNVP**, **Glow** and **GNF** which show **promising results** in generating images or even graphs.

Preliminary

The flow framework

The flow-based models aim to learn a sequence of invertible transformations

$$f_\Theta = f_L \circ \ldots \circ f_1$$

between **complex high-dimensional data** $X \sim P_{\mathcal{X}}(X)$ and $Z \sim P_{\mathcal{Z}}(Z)$ in a **latent space** with the same number of dimensions where the latent distribution $P_{\mathcal{Z}}(Z)$ is **easy** to model (e.g., strong independence assumptions hold in such a latent space).

Preliminary

The flow framework

The potentially complex data in the **original space** can be modelled by the change of **variable formula** where $Z = f_\Theta(X)$ and:

$$P_{\mathcal{X}}(X) = P_{\mathcal{Z}}(Z) \mid \det(\frac{\partial Z}{\partial X}) \mid$$

To sample $\tilde{X} \sim P_{\mathcal{X}}(X)$ is achieved by sampling $\tilde{Z} \sim P_{\mathcal{Z}}(Z)$ and then to transform $\tilde{X} = f_\Theta^{-1}(\tilde{Z})$ by the reverse mapping of f_Θ.

Preliminary

The flow framework

The potentially complex data in the **original space** can be modelled by the change of **variable formula** where $Z = f_\Theta(X)$ and:

$$P_{\mathcal{X}}(X) = P_{\mathcal{Z}}(Z) \mid \det(\frac{\partial Z}{\partial X}) \mid$$

Let $Z = f_\Theta(X) = f_L \circ \ldots \circ f_1(X)$, $H_l = f_l(H_{l-1})$ where f_l $(l = 1, \ldots L \in \mathbb{N}^+)$ are invertible mappings, $H_0 = X$, $H_L = Z$ and $P_{\mathcal{Z}}(Z)$ follows a standard isotropic Gaussian with independent dimensions. Then we get the log-likelihood of X by the change of variable formula as follows:

$$\log P_{\mathcal{X}}(X) = \log P_{\mathcal{Z}}(Z) + \log \mid \det(\frac{\partial Z}{\partial X}) \mid$$

Preliminary

The flow framework

$$\begin{aligned}\log P_{\mathcal{X}}(X) &= \log P_{\mathcal{Z}}(Z) + \log \mid \det(\frac{\partial Z}{\partial X}) \mid \\ &= \sum_i \log P_{\mathcal{Z}_i}(Z_i) + \sum_{l=1}^{L} \log \mid \det(\frac{\partial f_l}{\partial H_{l-1}}) \mid \end{aligned} \tag{2}$$

where $P_{\mathcal{Z}_i}(Z_i)$ is the probability of the i^{th} dimension of Z and $f_\Theta = f_L \circ \ldots \circ f_1$ is an invertible deep neural network to be learnt. Thus, the exact-likelihood-based training is tractable.

Preliminary

Invertible affine coupling layers

How to design an a)**invertible function** f_Θ with b)**expressive structures** and efficient computation of the c)**Jacobian determinant** are nontrivial.

The **NICE** and **RealNVP** design an **affine coupling transformation**

$$Z = f_\Theta(X) : \mathbb{R}^n \mapsto \mathbb{R}^n:$$

$$Z_{1:d} = X_{1:d}$$

$$Z_{d+1:n} = X_{d+1:n} \odot e^{S_\Theta(X_{1:d})} + T_\Theta(X_{1:d}),$$

by **splitting** X into two partitions $X = (X_{1:d}, X_{d+1:n})$. Thus, the invertibility is guaranteed by:

$$X_{1:d} = Z_{1:d}$$

$$X_{d+1:n} = (Z_{d+1:n} - T_\Theta(Z_{1:d}))/e^{S_\Theta(Z_{1:d})},$$

Preliminary

Invertible affine coupling layers

How to design an a)**invertible function** f_Θ with b)**expressive structures** and efficient computation of the c)**Jacobian determinant** are nontrivial.

b.) the expressive power depends on arbitrary neural structures of the **Scale function** $S_\Theta : \mathbb{R}^d \mapsto \mathbb{R}^{n-d}$ and the **Transformation function** $T_\Theta : \mathbb{R}^d \mapsto \mathbb{R}^{n-d}$ in the affine transformation of $X_{d+1:n}$,

c.) the Jacobian determiant can be computed efficiently by $\det(\frac{\partial Z}{\partial X}) = \exp\left(\sum_j S_\Theta(X_{1:d})_j\right)$.

Preliminary

Splitting Dimensions

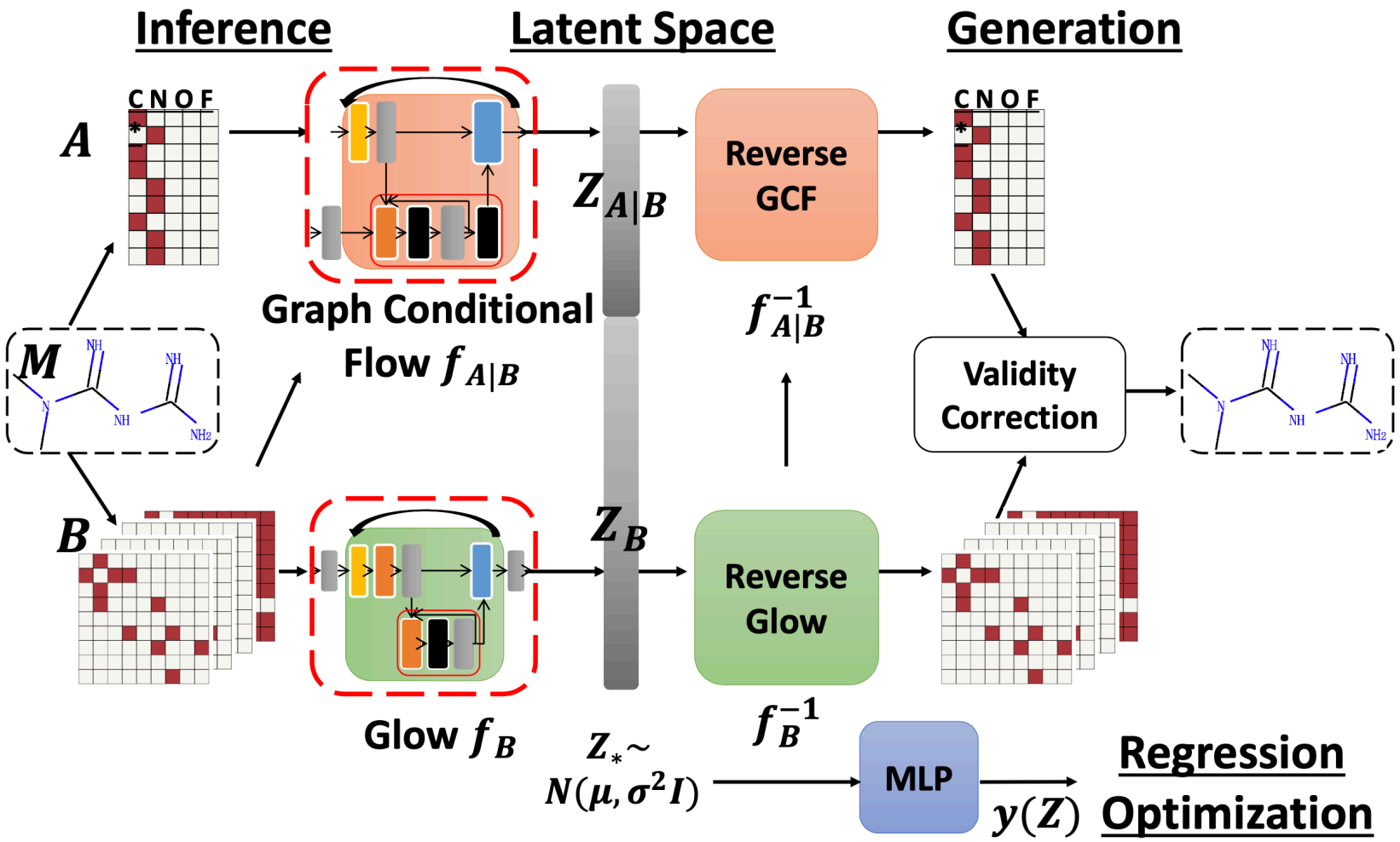
The flow-based models, e.g., **RealNVP** and **Glow**, adopt **squeeze operation** which compresses the spatial dimension $X^{c\times n\times n}$ into $X^{(ch^2)\times \frac{n}{h}\times \frac{n}{h}}$ to make more channels and then **split** channels into two halves for the **coupling layer**.

Preliminary

Numerical stability by actnorm

In order to ensure the numerical stability of the flow-based models, **actnorm layer** is introduced in **Glow** which normalizes dimensions in each channel over a batch by an **affine transformation** with learnable scale and bias. The scale and the bias are initialized as the mean and the inverse of the standard variation of the dimensions in each channel over the batch.

Proposed MoFlow Model



Proposed MoFlow Model

Problem Definition: Learning a Probability Model of Molecular Graphs

$$\mathcal{M} = \mathcal{A} \times \mathcal{B} \subset \mathbb{R}^{n \times k} \times \mathbb{R}^{c \times n \times n} \qquad n \in \mathbb{N}^+ \text{ atoms}$$

$$\mathcal{A}\text{tom} \qquad k \in \mathbb{N}^+ \text{ atom types}$$

$$\mathcal{B}\text{ond} \qquad c \in \mathbb{N}^+ \text{ bond types}$$

A molecule $M = (A, B) \in \mathcal{A} \times \mathcal{B}$

an atom matrix $A \in \mathbb{R}^{n \times k}$

a bond tensor $B \in \mathbb{R}^{c \times n \times n}$

Proposed MoFlow Model

Problem Definition: Learning a Probability Model of Molecular Graphs

A molecule $M = (A, B) \in \mathcal{A} \times \mathcal{B}$

an atom matrix $A \in \mathbb{R}^{n \times k}$

a bond tensor $B \in \mathbb{R}^{c \times n \times n}$

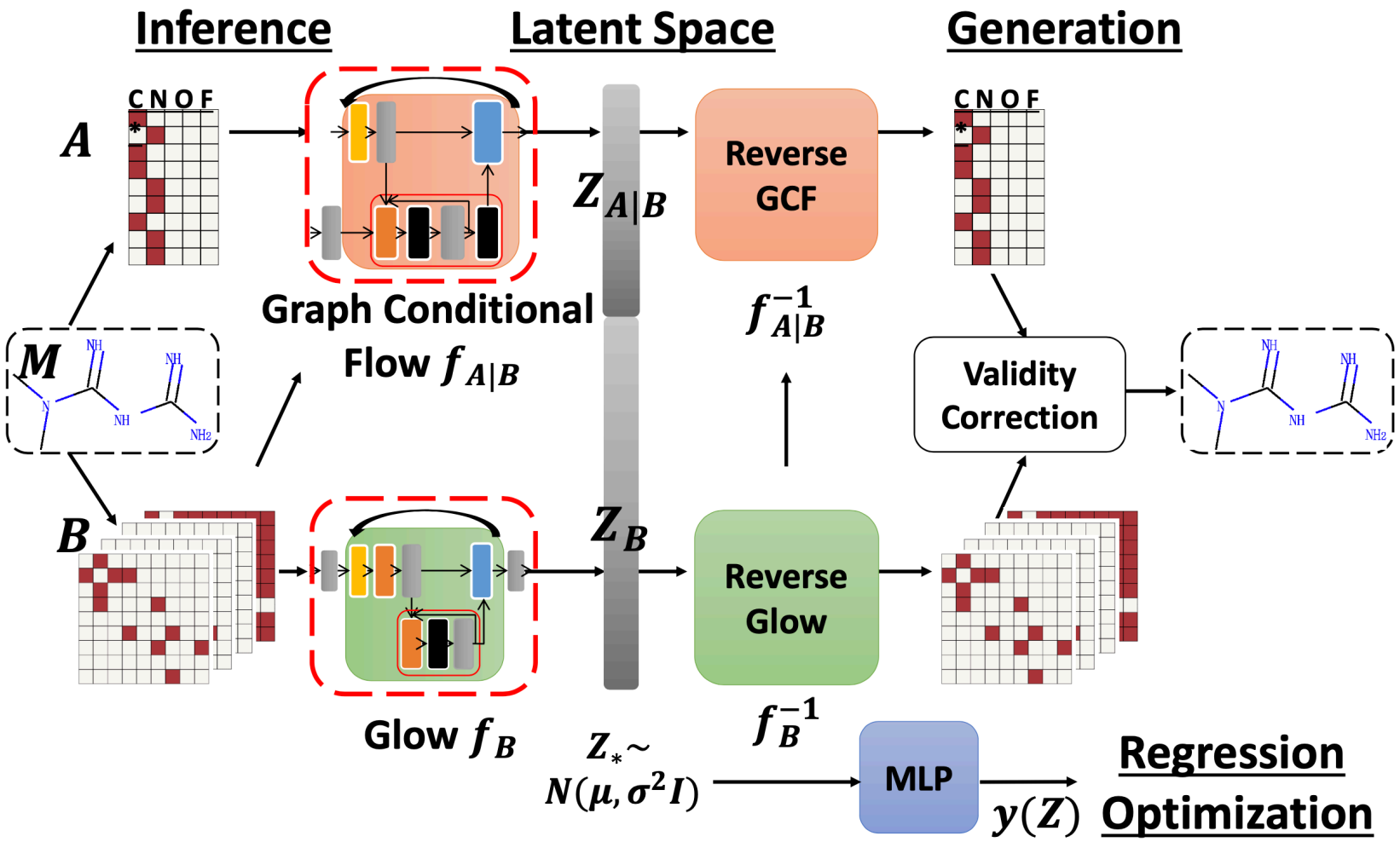
$$A(i, k) = 1$$

$$B(c, j, i) = 1$$

$$P_{\mathcal{M}}(M) = P_{\mathcal{M}}((A, B)) \approx P_{\mathcal{A}|\mathcal{B}}(A|B; \theta_{\mathcal{A}|\mathcal{B}}) P_{\mathcal{B}}(B; \theta_{\mathcal{B}})$$

$$\arg\max_{\theta_{\mathcal{B}}, \theta_{\mathcal{A}|\mathcal{B}}} \mathbb{E}_{M=(A,B) \sim p_{\mathcal{M}-data}} [\log P_{\mathcal{A}|\mathcal{B}}(A|B; \theta_{\mathcal{A}|\mathcal{B}}) + \log P_{\mathcal{B}}(B; \theta_{\mathcal{B}})]$$

Proposed MoFlow Model



Proposed MoFlow Model

Graph Conditional Flow for Atoms

Given a bond tensor $B \in \mathcal{B} \subset \mathbb{R}^{c \times n \times n}$

Our goal of the atom flow is to generate the right atom-type matrix $A \in \mathcal{A} \subset \mathbb{R}^{n \times k}$

to assemble valid molecules $M = (A, B) \in \mathcal{M} \subset \mathbb{R}^{n \times k + c \times n \times n}$

Proposed MoFlow Model

B-Conditional Flow and Graph Conditional Flow

Definition 4.1. $\mathcal{B}$**-conditional flow**: A $\mathcal{B}$**-conditional flow** $Z_{A|B}|B = f_{\mathcal{A}|\mathcal{B}}(A|B)$ is an invertible and dimension-kept mapping and there exists reverse transformation $f^{-1}_{\mathcal{A}|\mathcal{B}}(Z_{A|B}|B) = A|B$ where $f_{\mathcal{A}|\mathcal{B}}$ and $f^{-1}_{\mathcal{A}|\mathcal{B}} : \mathcal{A} \times \mathcal{B} \mapsto \mathcal{A} \times \mathcal{B}$.

The condition $B \in \mathcal{B}$ keeps fixed during the transformation. Under the independent assumption of $\mathcal{A}$ and $\mathcal{B}$, the Jacobian of $f_{\mathcal{A}|\mathcal{B}}$ is:

$$\frac{\partial f_{\mathcal{A}|\mathcal{B}}}{\partial(A, B)} = \begin{bmatrix} \frac{\partial f_{\mathcal{A}|\mathcal{B}}}{\partial A} & \frac{\partial f_{\mathcal{A}|\mathcal{B}}}{\partial B} \\ 0 & \mathbb{1}_B \end{bmatrix}, \tag{7}$$

the determiant of this Jacobian is $\det \frac{\partial f_{\mathcal{A}|\mathcal{B}}}{\partial(A,B)} = \det \frac{\partial f_{\mathcal{A}|\mathcal{B}}}{\partial A}$, and thus the *conditional version of the change of variable formula* in the form of log-likelihood is:

$$\log P_{\mathcal{A}|\mathcal{B}}(A|B) = \log P_{\mathcal{Z}_{\mathcal{A}|\mathcal{B}}}(Z_{A|B}) + \log \left| \det \frac{\partial f_{\mathcal{A}|\mathcal{B}}}{\partial A} \right| . \tag{8}$$

Proposed MoFlow Model

B-Conditional Flow and Graph Conditional Flow

Definition 4.2. Graph conditional flow: A graph conditional flow is a $\mathcal{B}$-conditional flow $Z_{A|B}|B = f_{\mathcal{A}|\mathcal{B}}(A|B)$ where $B \in \mathcal{B} \subset \mathbb{R}^{c \times n \times n}$ is the adjacency tenor for edges with c types and $A \in \mathcal{A} \subset \mathbb{R}^{n \times k}$ is the feature matrix of the corresponding n nodes.

Proposed MoFlow Model

Graph coupling layer

We construct aforementioned invertible mapping $f_{\mathcal{A}|\mathcal{B}}$ and $f^{-1}_{\mathcal{A}|\mathcal{B}}$ by the scheme of the affine coupling layer. The coupling transformation relies on **graph convolution** and thus we name such a coupling transformation as a graph coupling layer.

For each graph coupling layer, we **split** input $A \in \mathbb{R}^{n\times k}$ into two parts $A = (A_1, A_2)$ along the n row dimension, and we get the output $Z_{A|B} = (Z_{A_1|B}, Z_{A_2|B}) = f_{\mathcal{A}|\mathcal{B}}(A|B)$ as follows:

$$Z_{A_1|B} = A_1$$

$$Z_{A_2|B} = A_2 \odot \text{Sigmoid}(S_\Theta(A_1|B)) + T_\Theta(A_1|B)$$

Proposed MoFlow Model

Graph coupling layer

We design the scale function S_Θ and the transformation function T_Θ in each graph coupling layer by incorporating **graph convolution** structures.

The bond tensor $B \in \mathbb{R}^{c \times n \times n}$ keeps a fixed value during transforming the **atom matrix A.**

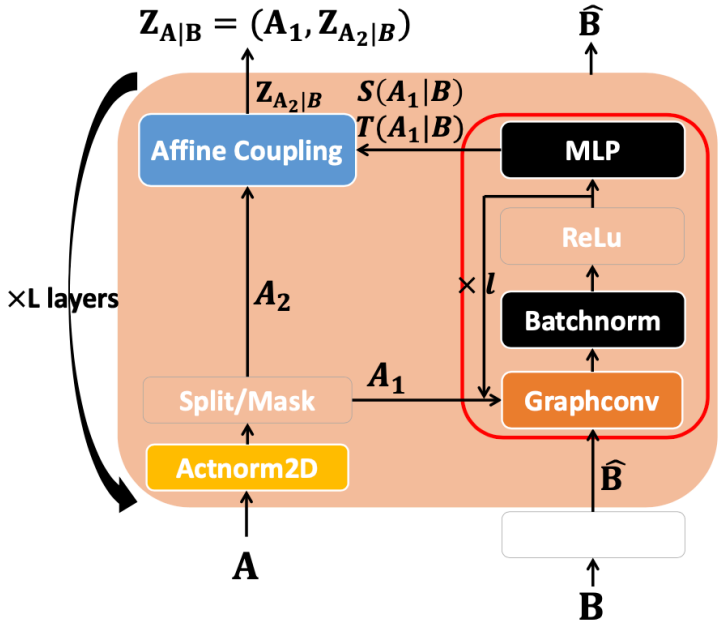


Figure 2: Graph conditional flow $f_{\mathcal{A}|\mathcal{B}}$ for the atom matrix.

Proposed MoFlow Model

Graph coupling layer

The reverse mapping of the graph coupling layer $f^{-1}_{\mathcal{A}|\mathcal{B}}$ is:

$$A_1 = Z_{A_1|B}$$

$$A_2 = (Z_{A_2|B} - T_\Theta(Z_{A_1|B}|B))/\text{Sigmoid}(S_\Theta(Z_{A_1|B}|B)).$$

The logarithm of the Jacobian determinant of each graph coupling layer can be efficiently computed by:

$$\log|\det(\frac{\partial f_{\mathcal{A}|\mathcal{B}}}{\partial A})| = \sum_j \log \text{Sigmoid}(S_\Theta(A_1|B))_j$$

Proposed MoFlow Model

Glow for Bonds

The bond flow aims to learn an invertible mapping

$$f_{\mathcal{B}} : \mathcal{B} \subset \mathbb{R}^{c\times n\times n} \mapsto \mathcal{B} \subset \mathbb{R}^{c\times n\times n}$$

where the transformed latent variable $Z_B = f_{\mathcal{B}}(B)$ follows isotropic Gaussian.

We also follow the scheme of **affine coupling layer** to build invertible mappings. For each affine coupling layer, We split input $B \in \mathbb{R}^{c\times n\times n}$ into two parts $B = (B_1, B_2)$ along the channel c dimension, and we get the output $Z_B = (Z_{B_1}, Z_{B_2})$ as follows:

$$Z_{B_1} = B_1$$

$$Z_{B_2} = B_2 \odot \text{Sigmoid}(S_\Theta(B_1)) + T_\Theta(B_1)$$

And thus the reverse mapping $f_{\mathcal{B}}^{-1}$ is:

$$B_1 = Z_{B_1}$$

$$B_2 = (Z_{B_2} - T_\Theta(Z_{B_1}))/\text{Sigmoid}(S_\Theta(Z_{B_1}))$$

Proposed MoFlow Model

Glow for Bonds

The scale function S_Θ and the transformation function T_Θ in each ane coupling layer can have arbitrary structures.

We use multiple 3x3 conv2d->BatchNorm2d->ReLu layers to build them.

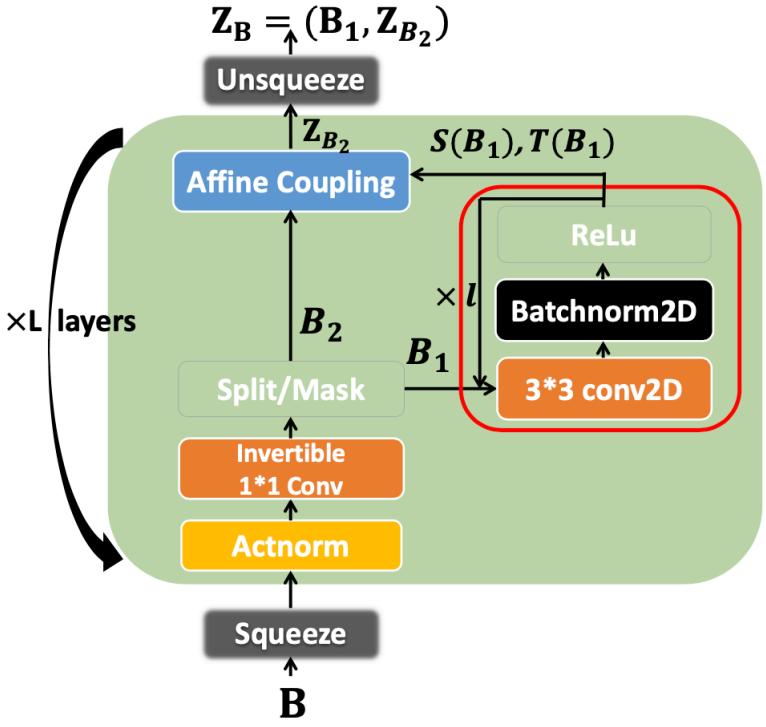


Figure 3: A variant of Glow $f_{\mathcal{B}}$ for bonds' adjacency tensors

Proposed MoFlow Model

Glow for Bonds

The logarithm of the Jacobian determiant of each affine coupling is:

$$\log \left| \det\left(\frac{\partial Z_B}{\partial B}\right) \right| = \sum_j \log \text{Sigmoid}(S_\Theta(B_1))_j$$

In order to learn optimal partition and ensure model's stability and learning rate, we also use the invertible 1x1 convolution layer and **actnorm layer** adopted in the **Glow**.

we **squeeze** the spatial size of B from $\mathbb{R}^{c \times n \times n}$ to $\mathbb{R}^{(c*h*h) \times \frac{n}{h} \times \frac{n}{h}}$ by a factor h and apply the affine coupling transformation to the squeezed data. The reverse **unsqueeze** operation is adopted to the output.

Proposed MoFlow Model

Validity Correction

Molecules must follow the **valency constraints** for each atom, but assembling a molecule from generated bond tensor and atom matrix may lead to chemically invalid ones. Here we define the valency constraint for the i^{th} atom as:

$$\sum_{c,j} c \times B(c, i, j) \leq \text{Valency}(\text{Atom}_i) + Ch$$

where $B \in \{0, 1\}^{c \times n \times n}$ is the one-hot bond tensor over $c \in \{1, 2, 3\}$ order of chemical bonds (single, double, triple) and $Ch \in \mathbb{N}$ represents the formal charge.

Experiments

- Molecular graph generation and reconstruction

Can **MoFlow** memorize and reconstruct all the training molecule datasets? Can **MoFlow** generalize to generate novel, unique and valid molecules as many as possible?

- Visualizing continuous latent space

MoFlow embed molecular graphs into continuous latent space with reasonable chemical similarity?

- Property optimization

Can **MoFlow** generate novel molecular graphs with optimized properties?

- Constrained property optimization

Can **MoFlow** generate novel molecular graphs with the optimized properties and at the same time keep the chemical similarity as much as possible?

Experiments

Datasets & Setup

For the **ZINC250K**, we adopt **10** coupling layers and **38** graph coupling layers for the bonds' Glow and the atoms' graph conditional flow respectively.
We use two **3x3** convolution layers with **512, 512** hidden dimensions in each coupling layer.

For the QM9, we adopt **10** coupling layers and **27** graph coupling layers for the bonds' Glow and the atoms' graph conditional flow respectively. There are two **3x3** convolution layers with **128, 128** hidden dimensions in each coupling layer

Table 1: Statistics of the datasets.

	#Mol. Graphs	Max. #Nodes	#Node Types	#Edge Types
QM9	133,885	9	4+1	3+1
ZINC250K	249,455	38	9+1	3+1

Experiments

Molecular graph generation and reconstruction

Table 2 and Table 3 show that our **MoFlow** outperfoms the **state-of-the-art** models on all the six metrics for both QM9 and ZINC250k datasets.

Table 2: Generation and reconstruction performance on QM9 dataset.

	% Validity	% Validity w/o check	% Uniqueness	% Novelty	% N.U.V.	% Reconstruct
GraphNVP [20]	83.1 ± 0.5	n/a	99.2 ± 0.3	58.2 ± 1.9	47.97	100
GRF [10]	84.5 ± 0.70	n/a	66.0 ± 1.15	58.6 ± 0.82	32.68	100
GraphAF [29]	100	67	94.51	88.83	83.95	100
MoFlow	**100.00 ± 0.00**	**96.17 ± 0.18**	**99.20 ± 0.12**	**98.03 ± 0.14**	**97.24 ± 0.21**	**100.00 ± 0.00**

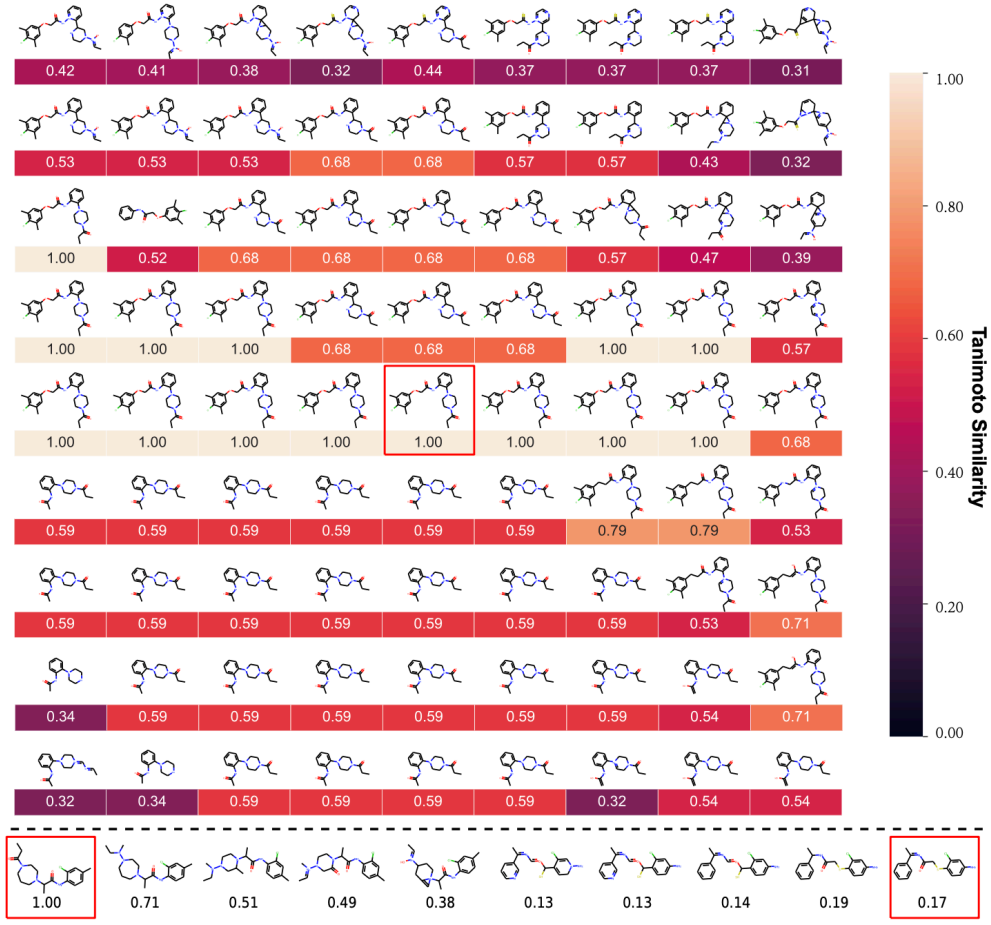
Table 3: Generation and reconstruction performance on ZINC250K dataset.

	% Validity	% Validity w/o check	% Uniqueness	% Novelty	% N.U.V.	% Reconstruct
JT-VAE [12]	100	n/a	100	100	100	76.7
GCPN [33]	100	20	99.97	100	99.97	n/a
MRNN [25]	100	65	99.89	100	99.89	n/a
GraphNVP [20]	42.6 ± 1.6	n/a	94.8 ± 0.6	100	40.38	100
GRF [10]	73.4 ± 0.62	n/a	53.7 ± 2.13	100	39.42	100
GraphAF [29]	100	68	99.10	100	99.10	100
MoFlow	**100.00 ± 0.00**	**81.76 ± 0.21**	**99.99 ± 0.01**	**100.00 ± 0.00**	**99.99 ± 0.01**	**100.00 ± 0.00**

Experiments

Visualizing Continuous Latent Space

We examine the learned latent space of our **MoFlow** by visualizing the decoded molecular graphs from a neighborhood of a latent vector in the latent space.

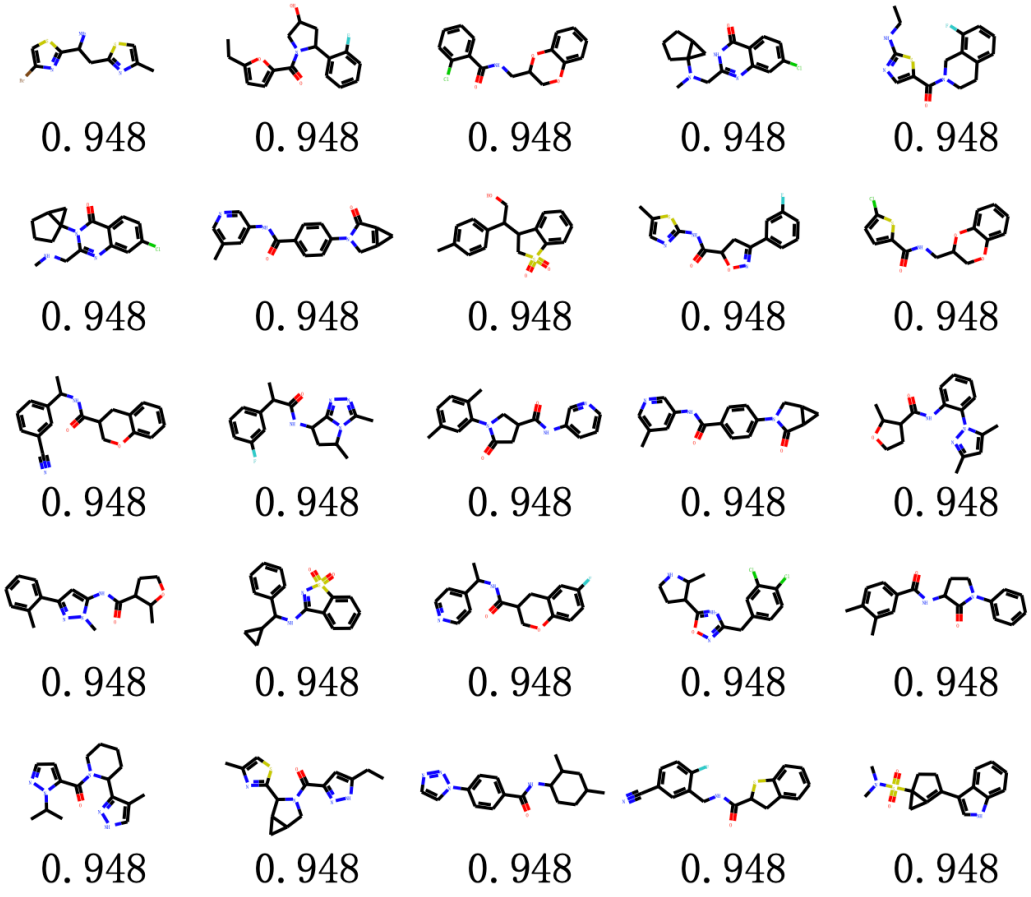


Experiments

Property optimization

The property optimization task aims at generating novel molecules with the best Quantitative Estimate of Druglikeness (QED) scores which measures the drug-likeness of generated molecules.

Method	1st	2nd	3rd	4th
ZINC (Dataset)	0.948	0.948	0.948	0.948
JT-VAE	0.925	0.911	0.910	-
GCPN	0.948	0.947	0.946	-
MRNN	0.948	0.948	0.947	-
GraphAF	0.948	0.948	0.947	0.946
MoFlow	**0.948**	**0.948**	**0.948**	**0.948**

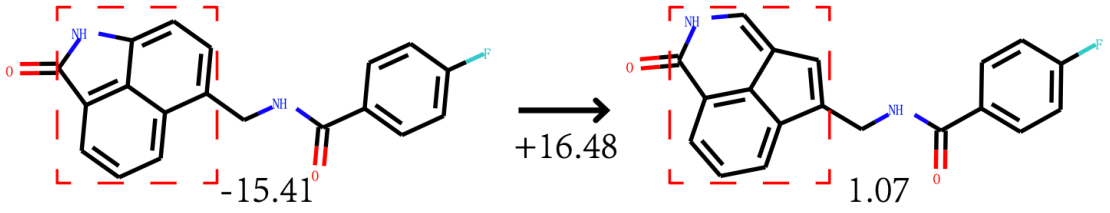


Experiments

Constrained Property Optimization

The constrained property optimization aims at finding a new molecule **M'** with the largest similarity score **sim(M,M')** and the largest improvement of a targeted property value y(M')-y(M) given a molecule M.

	JT-VAE			GCPN		
δ	**Improvement**	**Similarity**	**Success**	**Improvement**	**Similarity**	**Success**
0.0	1.91 ± 2.04	0.28 ± 0.15	97.5%	4.20 ± 1.28	**0.32 ± 0.12**	100%
0.2	1.68 ± 1.85	0.33 ± 0.13	97.1%	4.12 ± 1.19	0.34 ± 0.11	100%
0.4	0.84 ± 1.45	0.51 ± 0.10	83.6%	2.49 ± 1.30	0.48 ± 0.08	100%
0.6	0.21 ± 0.71	0.69 ± 0.06	46.4%	0.79 ± 0.63	0.68 ± 0.08	100%
	GraphAF			**MoFlow**		
δ	**Improvement**	**Similarity**	**Success**	**Improvement**	**Similarity**	**Success**
0.0	13.13 ± 6.89	0.29 ± 0.15	100%	8.61 ± 5.44	0.30 ± 0.20	98.88%
0.2	11.90 ± 6.86	0.33 ± 0.12	100%	7.06 ± 5.04	**0.43 ± 0.20**	96.75%
0.4	8.21 ± 6.51	0.49 ± 0.09	99.88%	4.71 ± 4.55	**0.61 ± 0.18**	85.75%
0.6	4.98 ± 6.49	0.66 ± 0.05	96.88%	2.10 ± 2.86	**0.79 ± 0.14**	58.25%



The End