Generating 3D Molecules for Target Protein Binding

Meng Liu, Youzhi Luo, Kanji Uchino, Koji Maruhashi, Shuiwang Ji

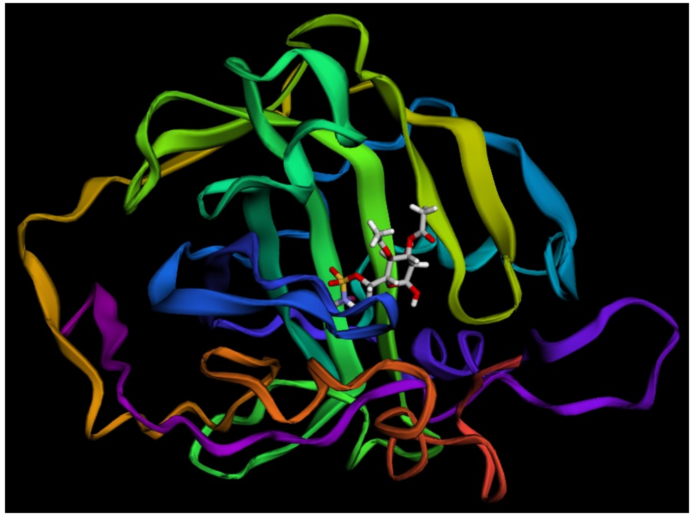
ICML 2022

TEXAS A&M UNIVERSITY
Engineering

FUJITSU

Structure-Based Drug Design

- Design molecules (ligands) that can bind to a specific target protein



- Deep learning methods become promising since there are large-scale datasets of protein-ligand complex structures
  - PDBbind (Liu et al., 2017) and CrossDocked2020 (Francoeur et al., 2020)

Figure from PDBbind

Challenges

- Complicated conditional information
  - 3D geometric structure
  - Chemical interaction

Challenges

- Complicated conditional information
  - 3D geometric structure
  - Chemical interaction

- Challenging search space
  - Enormous chemical space
  - Continuous 3D space

Challenges

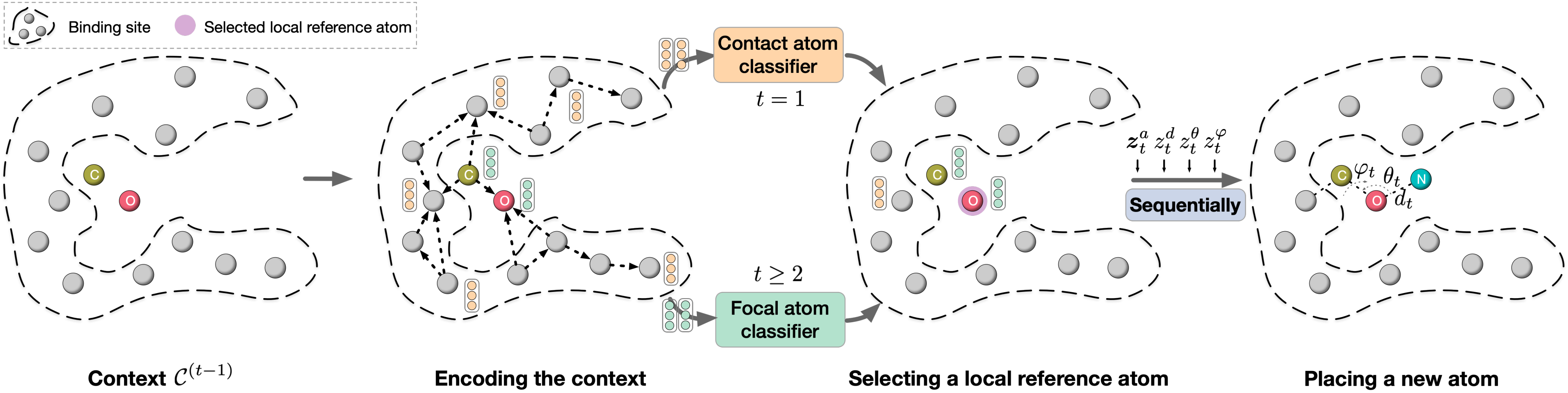
- Complicated conditional information
  - 3D geometric structure
  - Chemical interaction

- Challenging search space
  - Enormous chemical space
  - Continuous 3D space

- Equivariance property

The Proposed GraphBP: Overview

- Generate molecules that bind to given proteins, with considering the above challenges
  - Sequentially generate one atom per step based on the intermediate context



Binding site
Selected local reference atom
C
O
Contact atom classifier
$t = 1$
$t \geq 2$
Focal atom classifier
z_t^a z_t^d z_t^θ z_t^φ
Sequentially
φ_t θ_t d_t
N
Context $\mathcal{C}^{(t-1)}$
Encoding the context
Selecting a local reference atom
Placing a new atom

Notations

- 3D geometry of a molecule $\mathcal{M} = \{(\boldsymbol{a}_i, \boldsymbol{r}_i)\}_{i=1}^{n}$
  - a_i is a one-hot vector indicating the atom type
  - $r_i \in \mathbb{R}^3$ denotes a Cartesian coordinate
  - n is the number of atoms

- Similarly, the corresponding binding site of a protein is $\mathcal{P} = \{(\boldsymbol{b}_j, \boldsymbol{s}_j)\}_{j=1}^{m}$

- Our generative model aims to capture the conditional distribution $p(\mathcal{M}|\mathcal{P})$

Sequential Generation

- Place atoms in the given binding site one by one
  - Context at the step t = the binding site + atoms placed in the previous $t-1$ steps

$$\mathcal{C}^{(t-1)} = \mathcal{P} \cup \{(\boldsymbol{a}_i, \boldsymbol{r}_i)\}_{i=1}^{t-1}$$

  - Generate the atom type and the coordinate based on the context

$$\begin{aligned} \boldsymbol{a}_t &= g^a\left(\mathcal{C}^{(t-1)}; \boldsymbol{z}_t^a\right), \\ \boldsymbol{r}_t &= g^r\left(\mathcal{C}^{(t-1)}, \boldsymbol{a}_t; \boldsymbol{z}_t^r\right), \\ \mathcal{C}^{(t)} &\leftarrow \mathcal{C}^{(t-1)} \cup \{(\boldsymbol{a}_t, \boldsymbol{r}_t)\}. \end{aligned}$$

Update the context

g^a, g^r: parameterized autoregressive functions

z_t^a, z_t^r: latent variables in the flow model (introduced later)

Encoding the Context

- Construct a graph $\mathcal{G}^{(t-1)}$ for the context $\mathcal{C}^{(t-1)}$ by considering certain cutoff distance

- Employ a 3D GNN over the 3D graph to obtain node representations

$$\{\boldsymbol{h}_1^{(t)}, \cdots, \boldsymbol{h}_{m+t-1}^{(t)}\} = \text{3DGNN}\left(\mathcal{G}^{(t-1)}\right)$$

- The first embedding layer: different learnable embeddings to differentiate ligand atoms from protein atoms
- Aggregation of each 3D GNN layer

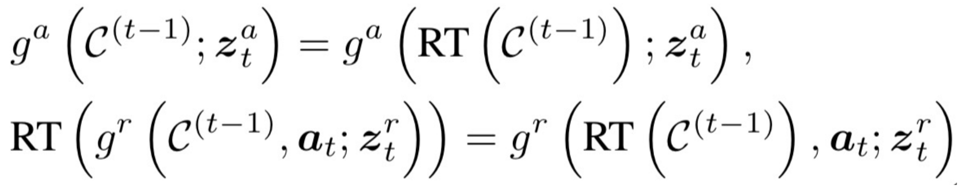
$$\boldsymbol{h}_k^{(t,\ell)} = \boldsymbol{h}_k^{(t,\ell-1)} + \sum_{u \in \mathcal{N}(k)} \boldsymbol{h}_u^{(t,\ell-1)} \odot \text{MLP}^\ell \left(\boldsymbol{e}_{\text{RBF}}\left(d_{uk}\right)\right)$$

Radial Basis Functions

The obtained representations are invariant to the rotation and translation of the context

Selecting A Local Reference Atom

- Generate coordinates that are equivariant to any rigid transformation (RT) of the binding site



$$g^a\left(\mathcal{C}^{(t-1)};\boldsymbol{z}_t^a\right)=g^a\left(\mathrm{RT}\left(\mathcal{C}^{(t-1)}\right);\boldsymbol{z}_t^a\right),$$

$$\mathrm{RT}\left(g^r\left(\mathcal{C}^{(t-1)},\boldsymbol{a}_t;\boldsymbol{z}_t^r\right)\right)=g^r\left(\mathrm{RT}\left(\mathcal{C}^{(t-1)}\right),\boldsymbol{a}_t;\boldsymbol{z}_t^r\right)$$

Selecting A Local Reference Atom

- Generate coordinates that are equivariant to any rigid transformation (RT) of the binding site

$$
\begin{aligned}
& g^a\left(\mathcal{C}^{(t-1)} ; \boldsymbol{z}_t^a\right)=g^a\left(\mathrm{RT}\left(\mathcal{C}^{(t-1)}\right) ; \boldsymbol{z}_t^a\right), \\
& \mathrm{RT}\left(g^r\left(\mathcal{C}^{(t-1)}, \boldsymbol{a}_t ; \boldsymbol{z}_t^r\right)\right)=g^r\left(\mathrm{RT}\left(\mathcal{C}^{(t-1)}\right), \boldsymbol{a}_t ; \boldsymbol{z}_t^r\right)
\end{aligned}
$$

- It is straightforward to generate invariant atom type with the obtained representations. How to generate coordinates equivariantly?

Selecting A Local Reference Atom

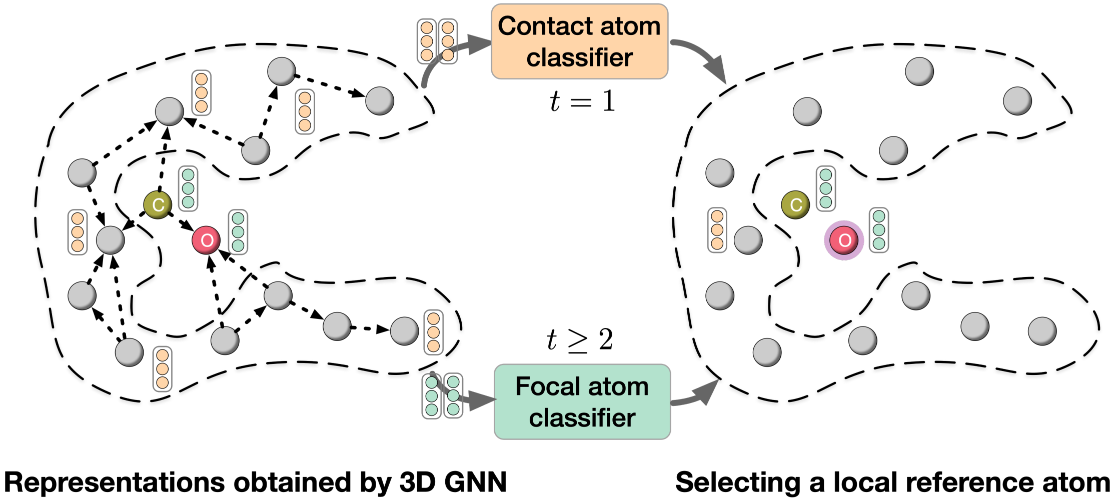
- Generate coordinates that are equivariant to any rigid transformation (RT) of the binding site

$$
\begin{aligned}
& g^a\left(\mathcal{C}^{(t-1)} ; \boldsymbol{z}_t^a\right)=g^a\left(\mathrm{RT}\left(\mathcal{C}^{(t-1)}\right) ; \boldsymbol{z}_t^a\right), \\
& \mathrm{RT}\left(g^r\left(\mathcal{C}^{(t-1)}, \boldsymbol{a}_t ; \boldsymbol{z}_t^r\right)\right)=g^r\left(\mathrm{RT}\left(\mathcal{C}^{(t-1)}\right), \boldsymbol{a}_t ; \boldsymbol{z}_t^r\right).
\end{aligned}
$$

- It is straightforward to generate invariant atom type with the obtained representations. How to generate coordinates equivariantly?
    - Construct a local spherical coordinate system (SCS) that is equivariant to the context
    - Generate the invariant 3-tuple $(d_t, \theta_t, \varphi_t)$ *w.r.t.* the constructed SCS
        - G-SchNet (Gabauer et al., 2019), MolGym (Simm et al., 2020), G-SphereNet (Luo & Ji, 2022)

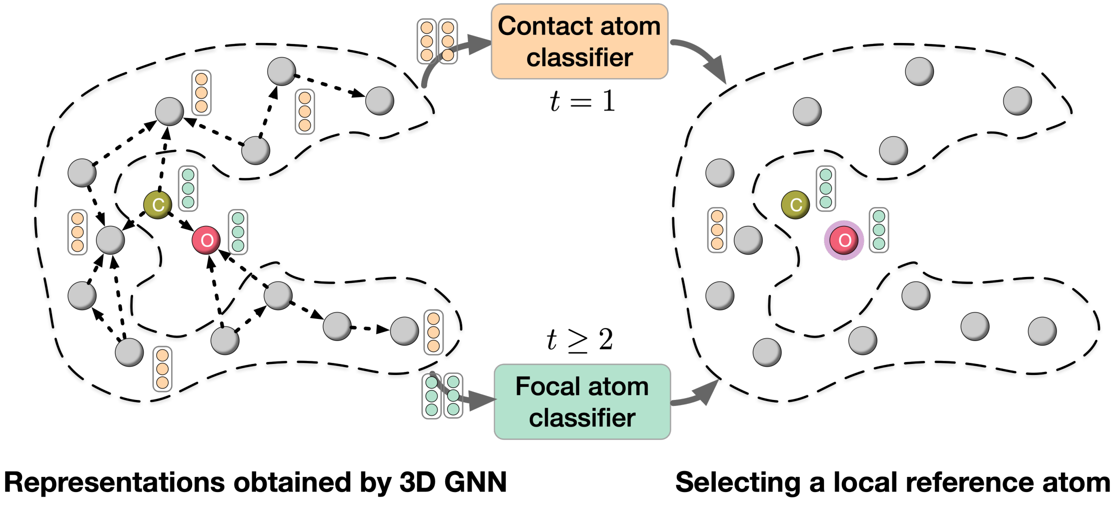
Selecting A Local Reference Atom

- Contact atom classifier ($t = 1$) over protein atoms
- Focal atom classifier ($t \geq 2$) over previously generated ligand atoms


Contact atom classifier
$t = 1$
$t \geq 2$
Focal atom classifier
C
O
C
O
Representations obtained by 3D GNN
Selecting a local reference atom

Selecting A Local Reference Atom

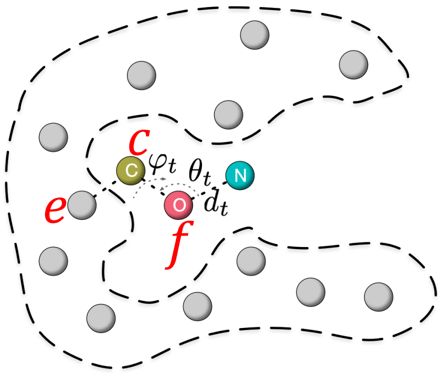
- Contact atom classifier ($t = 1$) over protein atoms
- Focal atom classifier ($t \geq 2$) over previously generated ligand atoms



Representations obtained by 3D GNN

Selecting a local reference atom

- Three points in the 3D space to defined a SCS
  - Consider the two atoms in the context that are closest and second closest to the selected local reference atom
  - This SCS is equivariant to the context naturally
  - Generate the invariant 3-tuple $(d_t, \theta_t, \varphi_t)$ *w.r.t.* the constructed SCS to place the new atom



Placing A New Atom

- Generate the invariant 3-tuple $(d_t, \theta_t, \varphi_t)$ with the context-encoded representations $(\boldsymbol{h}_f^{(t)}, \boldsymbol{h}_c^{(t)}, \boldsymbol{h}_e^{(t)})$
  - The representations are also invariant
  - Generate variables sequentially as $\boldsymbol{a}_t \rightarrow d_t \rightarrow \theta_t \rightarrow \varphi_t$ to capture the underlying dependencies

$$
\begin{aligned}
\boldsymbol{a}_t &= g^a\left(\mathcal{C}^{(t-1)}; \boldsymbol{z}_t^a\right), \\
d_t &= g^d\left(\mathcal{C}^{(t-1)}, \boldsymbol{a}_t; z_t^d\right), \\
\theta_t &= g^\theta\left(\mathcal{C}^{(t-1)}, \boldsymbol{a}_t, d_t; z_t^\theta\right), \\
\varphi_t &= g^\varphi\left(\mathcal{C}^{(t-1)}, \boldsymbol{a}_t, d_t, \theta_t; z_t^\varphi\right),
\end{aligned}
$$

  - Flow model: a parameterized invertible transformation function from the latent variable to the variable of interest
    - Training: map observed variables to latent variables, and maximize their likelihood
    - Generation: sample latent variables from known prior Gaussian distributions, and then map them to variables of interest

Training

- Decompose a 3D molecule in a ligand-protein pair to a trajectory of atom placement steps
  - We expect the new atom is placed in the local region of the reference atom during generation (Luo & Ji, 2022)
  - Select the atom in the binding site that is closest to the ligand as the first local reference atom (contact atom)
  - Apply Prim's algorithm on the 3D molecular geometry to obtain the placement order of atoms in the ligand, as well as their corresponding local reference atoms.

Training

- Decompose a 3D molecule in a ligand-protein pair to a trajectory of atom placement steps
  - We expect the new atom is placed in the local region of the reference atom during generation (Luo & Ji, 2022)
  - Select the atom in the binding site that is closest to the ligand as the first local reference atom (contact atom)
  - Apply Prim's algorithm on the 3D molecular geometry to obtain the placement order of atoms in the ligand, as well as their corresponding local reference atoms.

- Loss functions
  - **Atom placement loss**
    - We can compute the log-likelihood of training data exactly thanks to the property of the flow model
  - **Contact atom classifier loss**
    - Positive (negative) sample: Atom in the binding site that is closest (furthest) to the ligand
  - **Focal atom classifier loss**
    - The ground truth for an atom is negative if all of its bonded atoms have been generated, otherwise positive.

Experimental Setup

- 500k protein-ligand complexes from CrossDocked2020 for training
- 10 target proteins for test evaluation
  - These 10 proteins have 90 protein-ligand pairs in total. We use the corresponding ligand for reference.
  - Generate 100 molecules for each reference binding site.
  - Evaluation metric
    - **Validity**: The percentage of chemically valid molecules among all generated molecules.
    - **ΔBinding**: The percentage of generated molecules that have higher predicted binding affinity than their corresponding reference molecules.
- Baseline
  - LiGAN is a 3D CNN based generative model for structure-based drug design. LiGAN-posterior additionally encodes the whole reference protein-ligand complex as conditional information.

Experimental Results

- Better predicted binding affinity

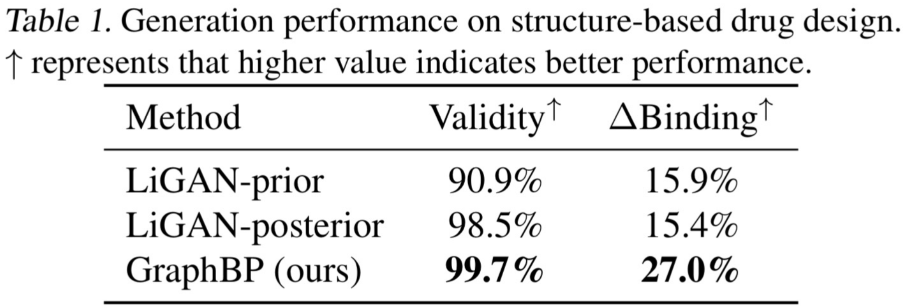
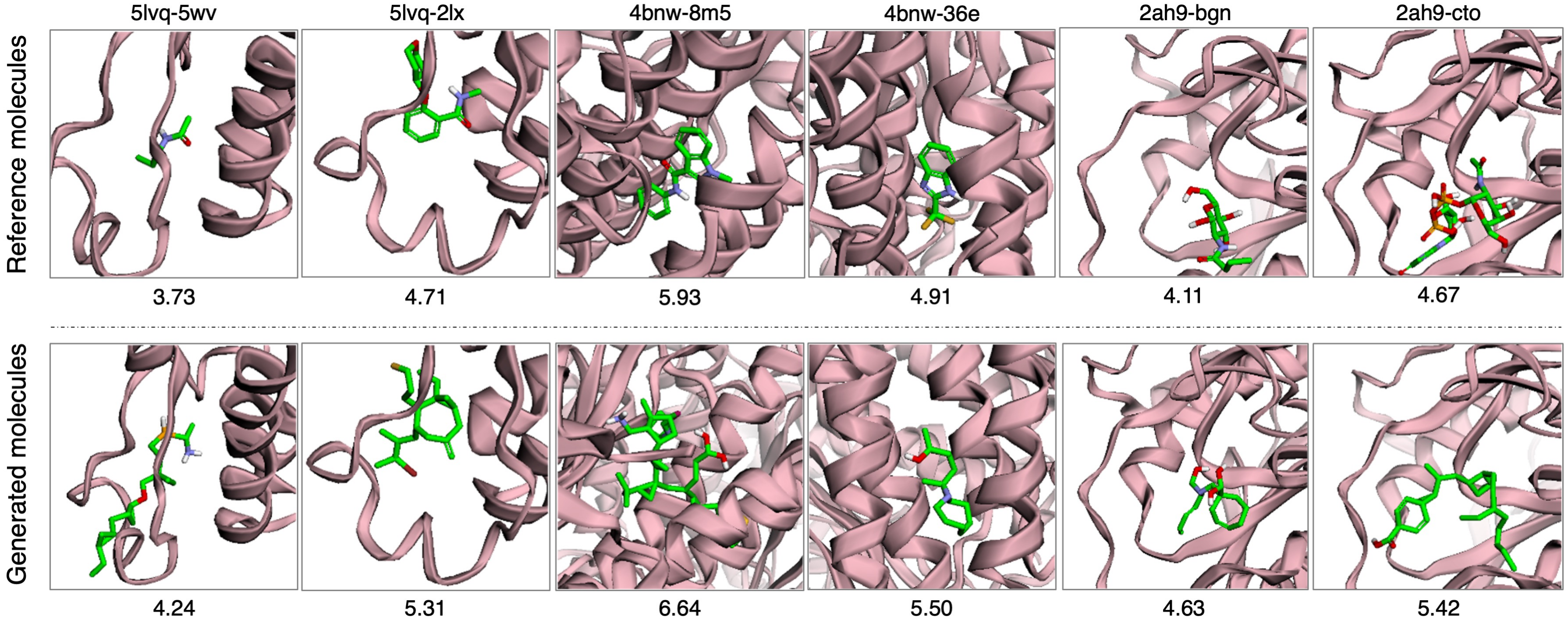


Table 1. Generation performance on structure-based drug design. ↑ represents that higher value indicates better performance.

Method	Validity↑	ΔBinding↑
LiGAN-prior	90.9%	15.9%
LiGAN-posterior	98.5%	15.4%
GraphBP (ours)	**99.7%**	**27.0%**

- Not simply memorizing or modifying known molecules



Ablation Study

- Sequentially generate the variables is effective to capture their underlying dependencies

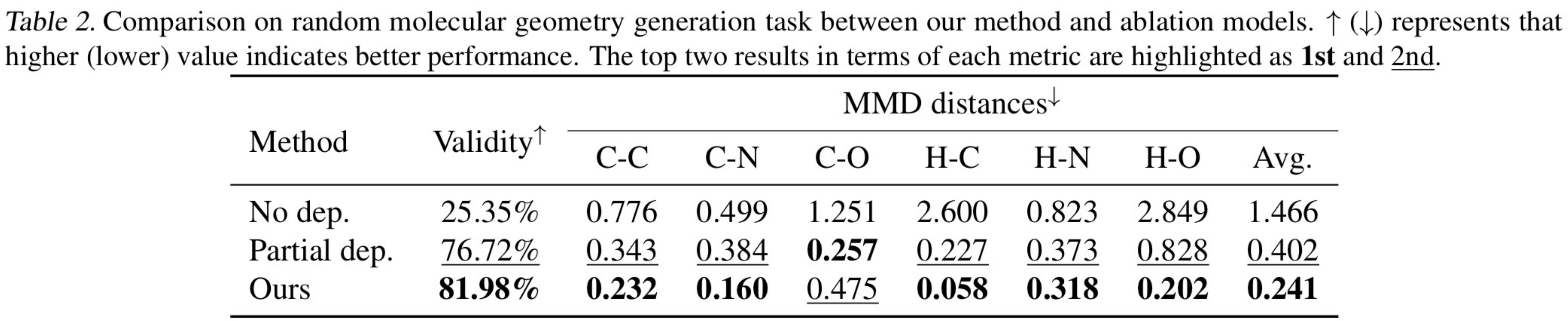
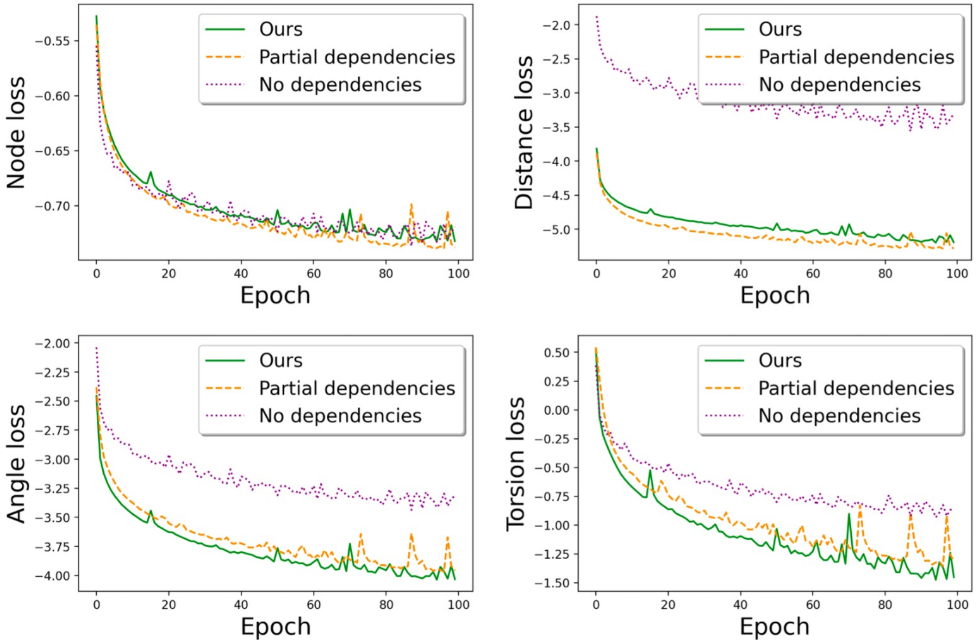
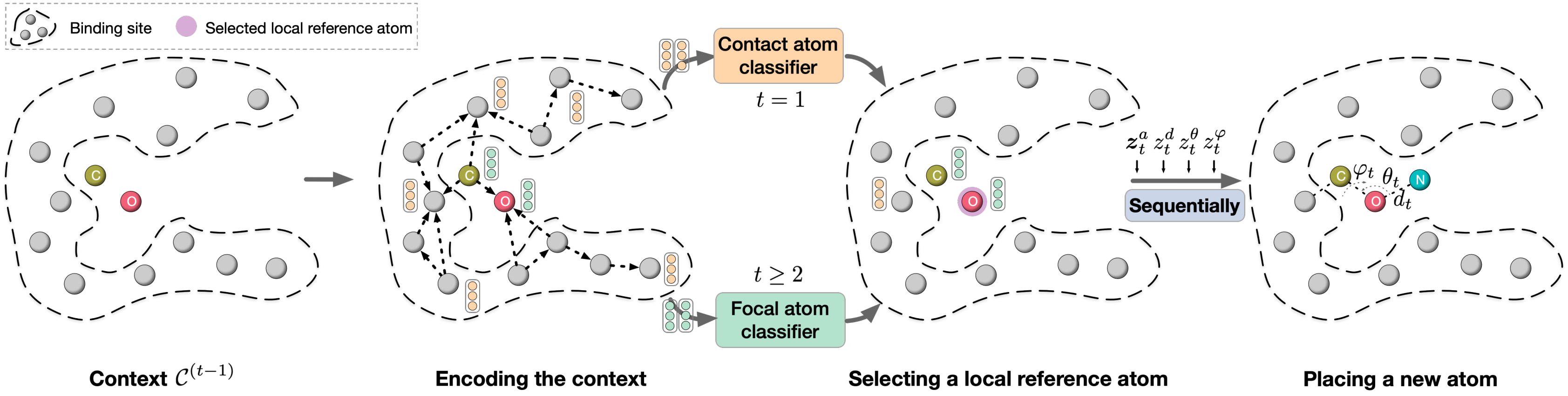


Table 2. Comparison on random molecular geometry generation task between our method and ablation models. ↑ (↓) represents that higher (lower) value indicates better performance. The top two results in terms of each metric are highlighted as **1st** and 2nd.

Method	Validity↑	MMD distances↓						
		C-C	C-N	C-O	H-C	H-N	H-O	Avg.
No dep.	25.35%	0.776	0.499	1.251	2.600	0.823	2.849	1.466
Partial dep.	76.72%	0.343	0.384	**0.257**	0.227	0.373	0.828	0.402
Ours	**81.98%**	**0.232**	**0.160**	0.475	**0.058**	**0.318**	**0.202**	**0.241**

$$\boldsymbol{a}_t \rightarrow d_t \rightarrow \theta_t \rightarrow \varphi_t$$




Binding site
Selected local reference atom
C
O
Contact atom classifier
$t = 1$
$t \geq 2$
Focal atom classifier
z_t^a z_t^d z_t^θ z_t^φ
Sequentially
φ_t θ_t d_t
N
Context $\mathcal{C}^{(t-1)}$
Encoding the context
Selecting a local reference atom
Placing a new atom

Paper

Code

@mengliu_1998

Poster: Hall E #338

Thank You!