Unsupervised Protein-Ligand Binding Energy Prediction via Neural Euler's Rotation Equation

arXiv

Wengong Jin, Siranush Sarkizova et al.

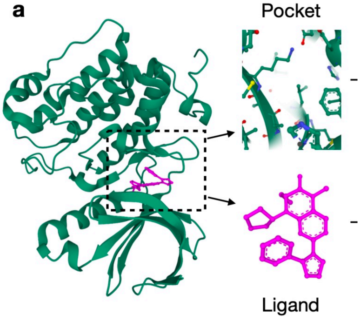
Reporter: Minjie Cheng

September 15, 2023

Overview

- Background: Binding affinity prediction task
- Main Challenges: Limited labelled data
- Unsupervised Method: Neural Euler's Rotation Equation
- Experiments: Protein-ligand and antibody-antigen affinity prediction
- Conclusion

Background: Binding affinity prediction task



Why the binding affinity is important?

- It is used to design drugs that bind well to their target.
- The drug must have a high affinity to the selected target.

Main Challenges: Limited labelled data

- Previous work focused on supervised learning methods for small molecules where binding affinity data is abundant.
- It is hard to transfer learning form small molecules to antibodies as their biochemical structures are very different.
- Labelled data is limited.
- Binding affinity is measured by experimentally, these are a variety methods for that, such as ELISE, Ggel Shift Assays, Pull Down Assays and others.
- They are expensive in terms of time, costs and human efforts.
- For antiboby-antigen, these are only 500 complexes with labelled affinity value.

Unsupervised Method: Neural Euler's Rotation Equation

- Reformulate binding affinity prediction as a generative modeling task.
- Train an energy-based model on a set of unlabelled protein-ligand complexes using SE(3) denoising score matching and interpret its log-likelihood as binding affinity.
- Outperform all unsupervised baselines and matches supervised learning methods in the antibody case.

Notions

- A protein-ligand complex is denoted as a tuple $(\boldsymbol{A}, \boldsymbol{X})$
    - $\boldsymbol{a}_i$: atom features, $\quad \boldsymbol{A} = [\boldsymbol{a}_i, ..., \boldsymbol{a}_n]$
    - $\boldsymbol{x}_i$: atom 3D coordinates, $\quad \boldsymbol{X} = [\boldsymbol{x}_i, ..., \boldsymbol{x}_n]$

Euler’s Rotation Equation

- Euler’s Rotation Equation is a first-order ordinary differential equation that describes the rotation of a rigid body.
- Suppose its current **angular velocity** is ω and $\boldsymbol{\tau}$ is a **torque** applied to the ligand. The Euler’s rotation equation is defined as:

$$\boldsymbol{I}_N \frac{d\boldsymbol{w}}{dt} + \boldsymbol{w} \times (\boldsymbol{I}_N \boldsymbol{w}) = \boldsymbol{\tau} \tag{1}$$

  - $\boldsymbol{w}$ is the angular velocity(角速度), which represents the rate of change of angles.
  - $\boldsymbol{w} = (\boldsymbol{w_x}, \boldsymbol{w_y}, \boldsymbol{w_z})$, $\boldsymbol{w_x} = \frac{d\psi}{dt}$; $\quad \boldsymbol{w_y} = \frac{d\theta}{dt}$; $\quad \boldsymbol{w_y} = \frac{d\varphi}{dt}$; $\quad \psi, \theta, \varphi$ represent the angles at which the rigid body rotates around the x, y, and z axes respectively (Euler angles).
  - $\boldsymbol{I}_N \in R^{3\times3}$ is the inertia matrix.(惯性矩阵)

Euler's Rotation Equation

- $\boldsymbol{\tau}$ and $\boldsymbol{I}_N$ depends on the rotation center $\boldsymbol{\mu}$

$$\boldsymbol{\tau} = \sum_{i \in \text{ligand}} (\boldsymbol{x}_i - \boldsymbol{\mu}) \times \boldsymbol{f}_i \tag{2}$$

$$\boldsymbol{I}_N = \sum_{i \in \text{ligand}} \|\boldsymbol{x}_i - \boldsymbol{\mu}\|^2 \boldsymbol{I} - (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T \tag{3}$$

- $\boldsymbol{\mu}$ is the center of a ligand.
- $\boldsymbol{f}_i$ is the force for atom i.
- When the ligand is static($\boldsymbol{w} = 0$), Eular's equation has a much simpler form $\boldsymbol{I}_N \frac{d\boldsymbol{w}}{dt} = \boldsymbol{\tau}$

Euler’s Rotation Equation

- When the ligand is static($\boldsymbol{w} = 0$), Eular’s equation has a much simpler form $\boldsymbol{I}_N \frac{d\boldsymbol{w}}{dt} = \boldsymbol{\tau}$
- The solution is $\boldsymbol{w} = C\boldsymbol{I}_N^{-1}\boldsymbol{\tau}$, replace $\boldsymbol{\tau}$ with $\boldsymbol{\tau} = \sum_{i\in\text{ligand}} (\boldsymbol{x}_i - \boldsymbol{\mu}) \times \boldsymbol{f}_i$

$$\boldsymbol{w} = C\boldsymbol{I}_N^{-1}\boldsymbol{\tau} = C\boldsymbol{I}_N^{-1} \sum_{i\in\text{ligand}} (\boldsymbol{x}_i - \boldsymbol{\mu}) \times \boldsymbol{f}_i) \tag{4}$$

- Given the predicted angular velocity $\boldsymbol{w}$, its corresponding rotation matrix is defined by a matrix exponential map

$$\boldsymbol{R}_{\boldsymbol{w}} = \exp\left(\boldsymbol{W}_{\boldsymbol{w}}\right), \quad \boldsymbol{W}_{\boldsymbol{w}} = \begin{bmatrix} 0 & -\boldsymbol{w}_z & \boldsymbol{w}_y \\ \boldsymbol{w}_z & 0 & -\boldsymbol{w}_x \\ -\boldsymbol{w}_y & \boldsymbol{w}_x & 0 \end{bmatrix} \tag{5}$$

Euler's Rotation Equation

$$\boldsymbol{R}_{\boldsymbol{w}} = \exp\left(\boldsymbol{W}_{\boldsymbol{w}}\right), \quad \boldsymbol{W}_{\boldsymbol{w}} = \begin{bmatrix} 0 & -\boldsymbol{w}_z & \boldsymbol{w}_y \\ \boldsymbol{w}_z & 0 & -\boldsymbol{w}_x \\ -\boldsymbol{w}_y & \boldsymbol{w}_x & 0 \end{bmatrix} \tag{6}$$

- The matrix exponential has the following closed form

$$\boldsymbol{R}_{\boldsymbol{w}} = \exp(\boldsymbol{W}_{\boldsymbol{w}}) = \boldsymbol{I} + c_1 \boldsymbol{W}_{\boldsymbol{w}} + c_2 \boldsymbol{W}_{\boldsymbol{w}}^2 \tag{7}$$

$$c_1 = \frac{sin\|\boldsymbol{w}\|}{\|\boldsymbol{w}\|}, \quad c_2 = \frac{1 - cos\|\boldsymbol{w}\|}{\|\boldsymbol{w}\|^2} \tag{8}$$

Euler's Rotation Equation

- The matrix exponential has the following closed form

$$\boldsymbol{R}_{\boldsymbol{w}} = \exp(\boldsymbol{W}_{\boldsymbol{w}}) = \boldsymbol{I} + c_1 \boldsymbol{W}_{\boldsymbol{w}} + c_2 \boldsymbol{W}_{\boldsymbol{w}}^2 \tag{9}$$

$$c_1 = \frac{sin\|\boldsymbol{w}\|}{\|\boldsymbol{w}\|}, \quad c_2 = \frac{1 - cos\|\boldsymbol{w}\|}{\|\boldsymbol{w}\|^2} \tag{10}$$

- Because $\boldsymbol{w} \times \boldsymbol{r} = \boldsymbol{W}_{\boldsymbol{w}}\boldsymbol{r}$, the new coordinate of $\boldsymbol{x}_i$ after rotation can be defined as :

$$\boldsymbol{x}_i^{\text{new}} = \boldsymbol{R}_{\boldsymbol{w}}\boldsymbol{x}_i = \boldsymbol{x}_i + c_1 \boldsymbol{w} \times \boldsymbol{x}_i + c_2 \boldsymbol{w} \times (\boldsymbol{w} \times \boldsymbol{x}_i) \tag{11}$$

- The advantage is that it avoids exponential calculations.
- It satisfy SE(3) equivariance.

The defination of force $\boldsymbol{f}_i$ and energy function

- We model the force term as the gradient of function $\frac{\partial(\mathrm{E}(\boldsymbol{A},\boldsymbol{X}))}{\partial \boldsymbol{x}_i}$
- The energy function must be differentiable with respect to $\boldsymbol{X}$ and SE(3)-invariant.
- The energy function is parameterized as follows:

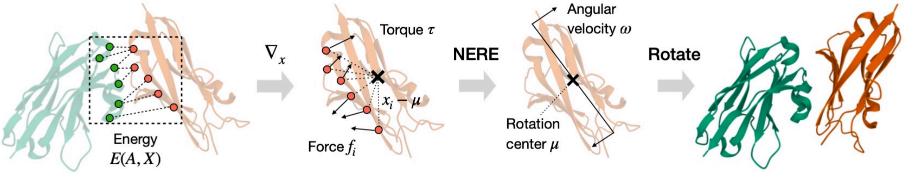
$$\boldsymbol{H} = \frac{1}{|G|} \sum_{g_k \in G} \phi_h(\boldsymbol{A}, g_k(\boldsymbol{X})) \tag{12}$$

$$E(\boldsymbol{A}, \boldsymbol{X}) = \sum_{i,j} \phi_o(\boldsymbol{h}_i, \boldsymbol{h}_j)(D_{i,j} < d) \tag{13}$$

$$\boldsymbol{f}_i = (\frac{\partial E(\boldsymbol{A}, \boldsymbol{X})}{\partial \boldsymbol{x}_i})^T \tag{14}$$

- $g_k(\boldsymbol{X})$ projects the coordinates X onto a set of frames defined in Puny et.al.
- Average the frames to maintain SE(3)-invariance.
- ϕ_h is the self-attention neural networks; The pairwise potential $\phi_0(\boldsymbol{h}_i, \boldsymbol{h}_j)$ is a scalar.

In all, NERE predicts a rotation in three steps



- First, it calculates the force based on the gradient $\boldsymbol{f}_i = (\partial E / \partial \mathbf{x}_i)^T$ by Equation 13.
- Then, predicts the corresponding **torque** (Equation 3) and **angular velocity** (Equation 4) by solving Euler's rotation equation.
- Lastly, it converts the angular vector to a **rotation** by Equation 11.

Unsupervised Binding Energy Prediction-DSM with Gaussian Noise

- Following the standard denoising score matching(DSM), we create a complex by adding Gaussian noise to ligand atom coordinates, i.e., $\tilde{\boldsymbol{X}} = \boldsymbol{X} + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$.
- The objective tries to match the score of our model $\partial E / \partial \tilde{\mathbf{X}}$ and the score of the noise distribution $\nabla_\epsilon \log(p(\epsilon)) = -\epsilon / \sigma^2$

$$l_{dsm} = \mathbb{E}[\| \frac{\partial E(\boldsymbol{A}, \tilde{\boldsymbol{X}})}{\partial \tilde{\mathbf{X}}} - \nabla_\epsilon \log(p(\epsilon)) \|^2] \tag{15}$$

- Drawback：Adding Gaussian noise is not ideal for protein-ligand binding because it may create nonsensical conformations that violate physical constraints (e.g., an aromatic ring (芳香环) must be planar)

DSM with rigid transformation Noise

- Gaussian noise $\Rightarrow$ Rigid transformation noise.
- The solution is to create a perturbed complex$(\boldsymbol{A}, \tilde{\boldsymbol{X}})$ via random ligand rotation and translation.

DSM with rigid transformation Noise

- To construct a random rotation, we sample an angular velocity $\boldsymbol{w}$ from $\mathcal{N}_{SO(3)}$, an isotropic Gaussoan distribution over SO(3) rotation group with variance σ^2.
- Each $\boldsymbol{w} \sim \mathcal{N}_{SO(3)}$ has the form $\boldsymbol{w} = \theta\tilde{\boldsymbol{w}}$, where $\tilde{\boldsymbol{w}}$ is a vector sampled uniformly from a unit sphere and $\theta \in [0, \pi]$ is a rotation angle with density

$$f(\theta) = \frac{1 - \cos(\theta)}{\pi} \sum_{l=0}^{\infty} (2l+1) e^{-l(l+1)\sigma^2} \frac{sin((l+1/2)\theta)}{\sin(\theta/2)} \tag{16}$$

- So we can get the score of rotation noise$\nabla_{\boldsymbol{w}} \log p(\boldsymbol{w}) = \nabla_{\boldsymbol{\theta}} \log f(\boldsymbol{\theta})\tilde{\boldsymbol{w}}$

DSM with rigid transformation Noise

- Likewise, we sample a random translation vector $\boldsymbol{t}$ for a normal distribution $\boldsymbol{t} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$, the score is $\nabla_{\boldsymbol{t}} \log(p(\boldsymbol{t})) = -\boldsymbol{t}/\sigma^2$

DSM with rigid transformation Noise

- With rotation noise $\boldsymbol{w} \sim \mathcal{N}_{SO(3)}$ and translation noise $\boldsymbol{t} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$, we can compute the perturbed coordinates $\tilde{\boldsymbol{X}} = \boldsymbol{R}_{\boldsymbol{w}} \boldsymbol{X} + \boldsymbol{t}$
- The score function $\partial E / \partial \tilde{\boldsymbol{X}}$ can not directly comparable with $\nabla_{\boldsymbol{w}} \log(p(\boldsymbol{w}))$ and $\nabla_{\boldsymbol{t}} \log(p(\boldsymbol{t}))$ because different dimensions.
- To address this issue, we propose to project $\partial E / \partial \tilde{\boldsymbol{X}}$ to angular velocity $\tilde{\boldsymbol{w}}$ via NERE.

DSM with rigid transformation Noise

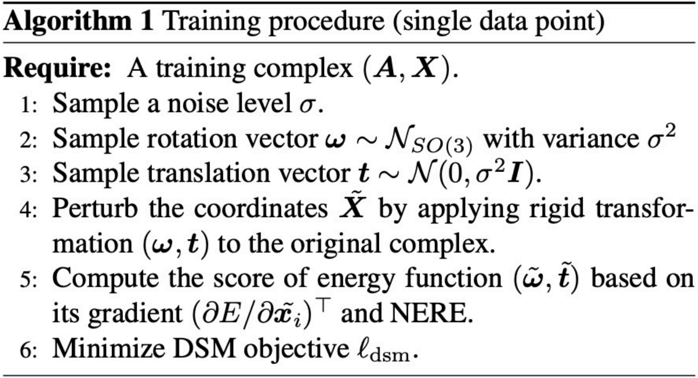
- The objective function can be defined as

$$l_{dsm} = \mathbb{E}[\|\tilde{\boldsymbol{w}} - \nabla_{\boldsymbol{w}} \log(p(\boldsymbol{w}))\|^2 + \|\tilde{\boldsymbol{t}} - \nabla_{\boldsymbol{t}} \log(p(\boldsymbol{t}))\|^2] \tag{17}$$

$$\tilde{\boldsymbol{w}} = C\boldsymbol{I}_N^{-1}\tau; \quad \tilde{\boldsymbol{t}} = \frac{1}{n}\sum_i \boldsymbol{f}_i \tag{18}$$

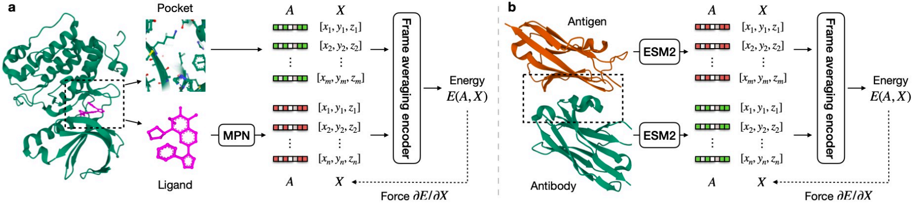
- In the experiments, the projected score matching objective is effective.
- No theoretical analysis in the paper.

Algorithm



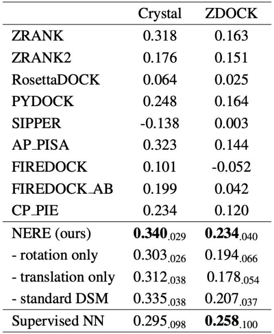
```
Algorithm 1 Training procedure (single data point)
Require: A training complex $(\boldsymbol{A}, \boldsymbol{X})$.
1: Sample a noise level $\sigma$.
2: Sample rotation vector $\boldsymbol{\omega} \sim \mathcal{N}_{SO(3)}$ with variance $\sigma^2$
3: Sample translation vector $\boldsymbol{t} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$.
4: Perturb the coordinates $\tilde{\boldsymbol{X}}$ by applying rigid transformation $(\boldsymbol{\omega}, \boldsymbol{t})$ to the original complex.
5: Compute the score of energy function $(\tilde{\boldsymbol{\omega}}, \tilde{\boldsymbol{t}})$ based on its gradient $(\partial E / \partial \tilde{\boldsymbol{x}}_i)^\top$ and NERE.
6: Minimize DSM objective $\ell_{\mathrm{dsm}}$.
```

Experiments



- For small molecules, we jointly train the MPN and energy function using the DSM objective.
- For antibodies, we do not fine-tune ESM-2 during training.

Experiments-Antibody-Antigen binding



	Crystal	ZDOCK
ZRANK	0.318	0.163
ZRANK2	0.176	0.151
RosettaDOCK	0.064	0.025
PYDOCK	0.248	0.164
SIPPER	-0.138	0.003
AP_PISA	0.323	0.144
FIREDOCK	0.101	-0.052
FIREDOCK_AB	0.199	0.042
CP_PIE	0.234	0.120
NERE (ours)	$\mathbf{0.340}_{.029}$	$\mathbf{0.234}_{.040}$
- rotation only	$0.303_{.026}$	$0.194_{.066}$
- translation only	$0.312_{.038}$	$0.178_{.054}$
- standard DSM	$0.335_{.038}$	$0.207_{.037}$
Supervised NN	$0.295_{.098}$	$\mathbf{0.258}_{.100}$

Table 2. Pearson correlation on SAbDab test set (both crystal and docked structures). Standard deviation is only shown for NERE because baseline models are deterministic.

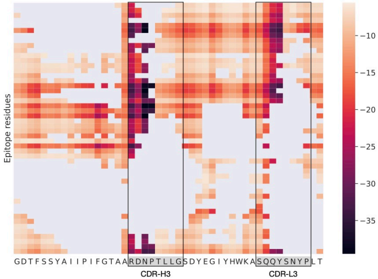


Figure 4. The heat map of our learned energy function. Each entry represent the binding energy between residues (i, j) (the darker the stronger). An entry is left blank (grey) if their distance $\boldsymbol{D}_{ij} > d$. Our model correctly puts more attention to CDR-H3/L3 residues.

- Each entry in Fig.4 is the energy between two residues.
- This agrees with the domain knowledge that CDR-H3 and CDR-L3 residues is the major component for binding.

Experiments-energy function

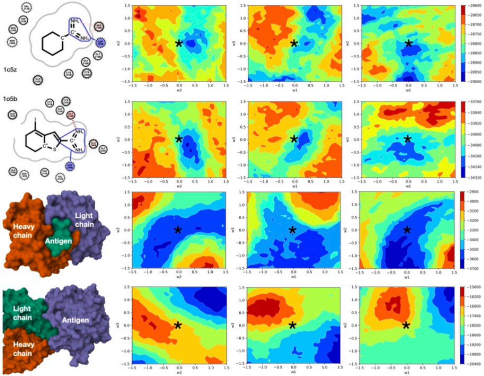


Figure 5. Visualizing learned energy landscape for small molecules (row 1-2) and antibodies (row 3-4). We perform a grid search of ligand rotation angles $\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]$ and plot the predicted energy as 2D contour plots with one of the axis ω_i fixed. The crystal structure is at the origin (marked with *) and supposed to be the local minima of the energy landscape.

- The crystal structure is supposed to be the local minima of the energy landscape
- For row(1-3), the crystal structures are located relatively near the local minima
- Row 4 fails to recognize crystal as local minima.

Conclusion

- It proposes an energy based model based on score marching model.
- It uses equivariant rotation noise based on Euler's Rotation Equation.
- Infers energy function with binding affinity for small molecules and antibodies.