UNIVERSITÄT
HEIDELBERG

UNIVERSITY OF
CAMBRIDGE

Drug Design – Present and Future

L10, Structural Bioinformatics

WiSe 2023/24, Heidelberg University

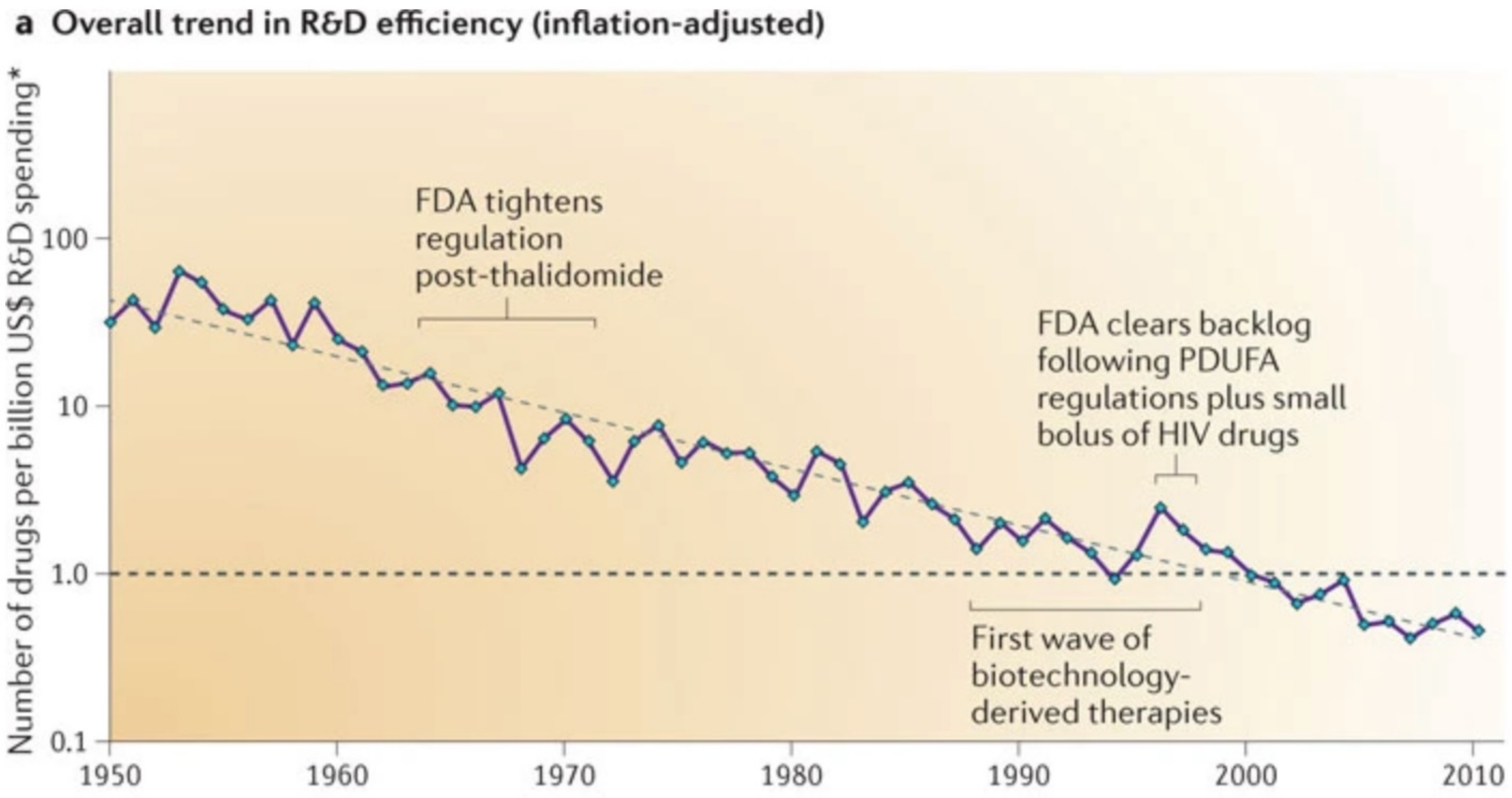
Overview for this lecture

1. Background: Drug Discovery Pipeline
2. Traditional approaches to early-stage design
3. Deep learning-based docking methods
4. Generative modelling for drug design

1. Background: Drug Discovery

Eroom’s Law

Drug Discovery is hard

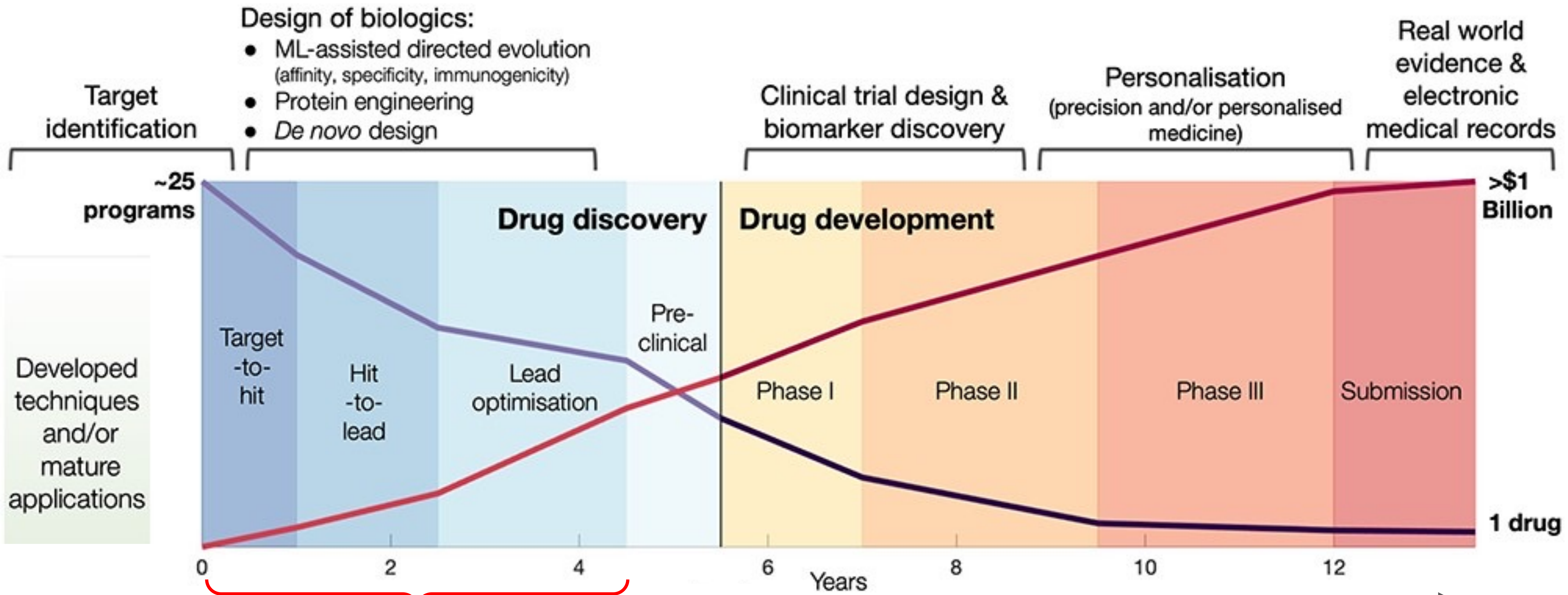


Recommended Reading:

- Derek Lowe – ‘In the pipeline’

Image source – Scannel et al, 2012

The Drug Discovery Pipeline

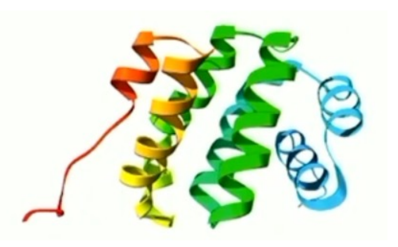


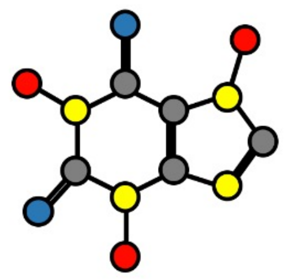
Really complicated

My work

This talk

Really complicated



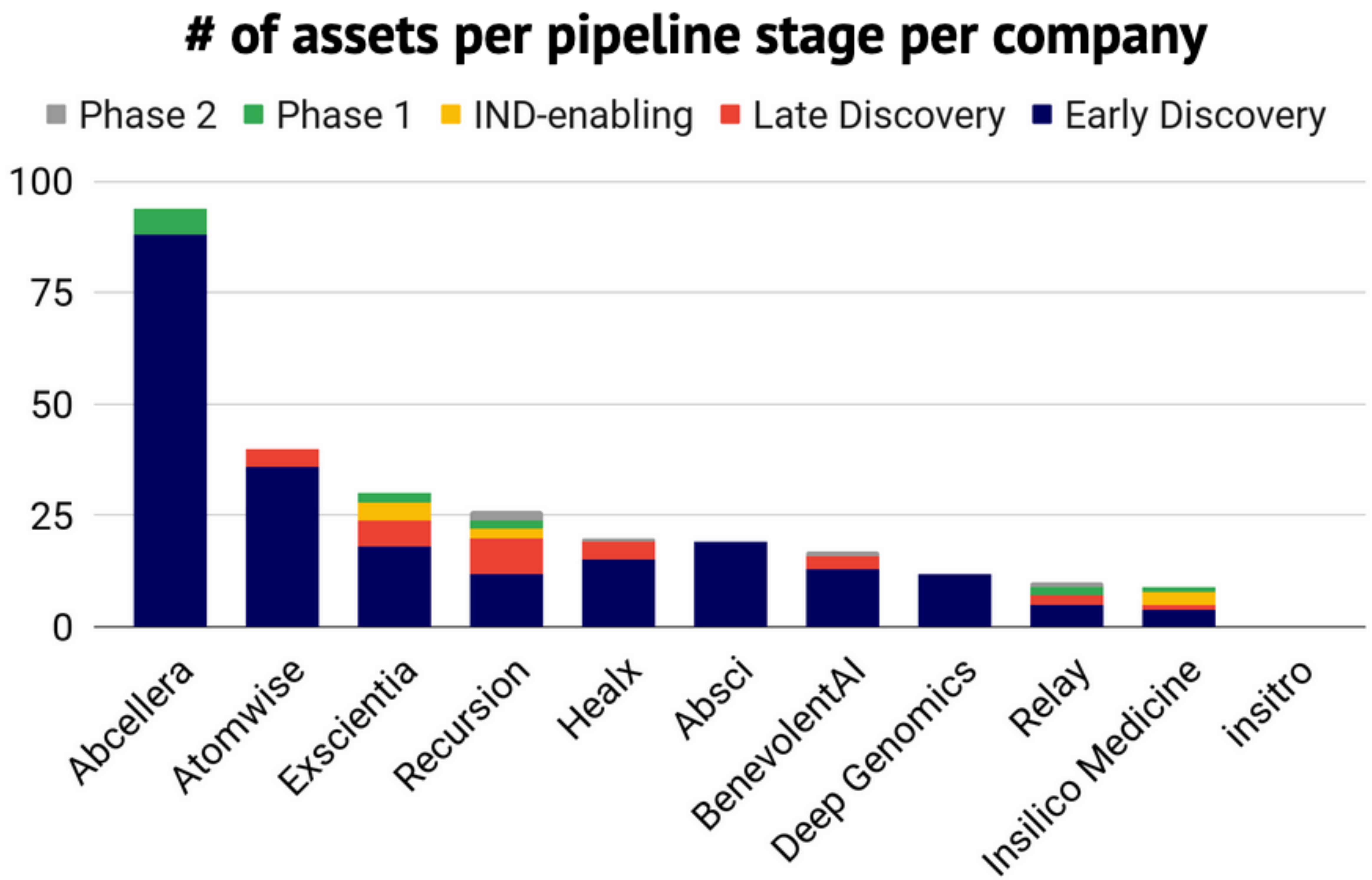


$X \rightarrow Y$

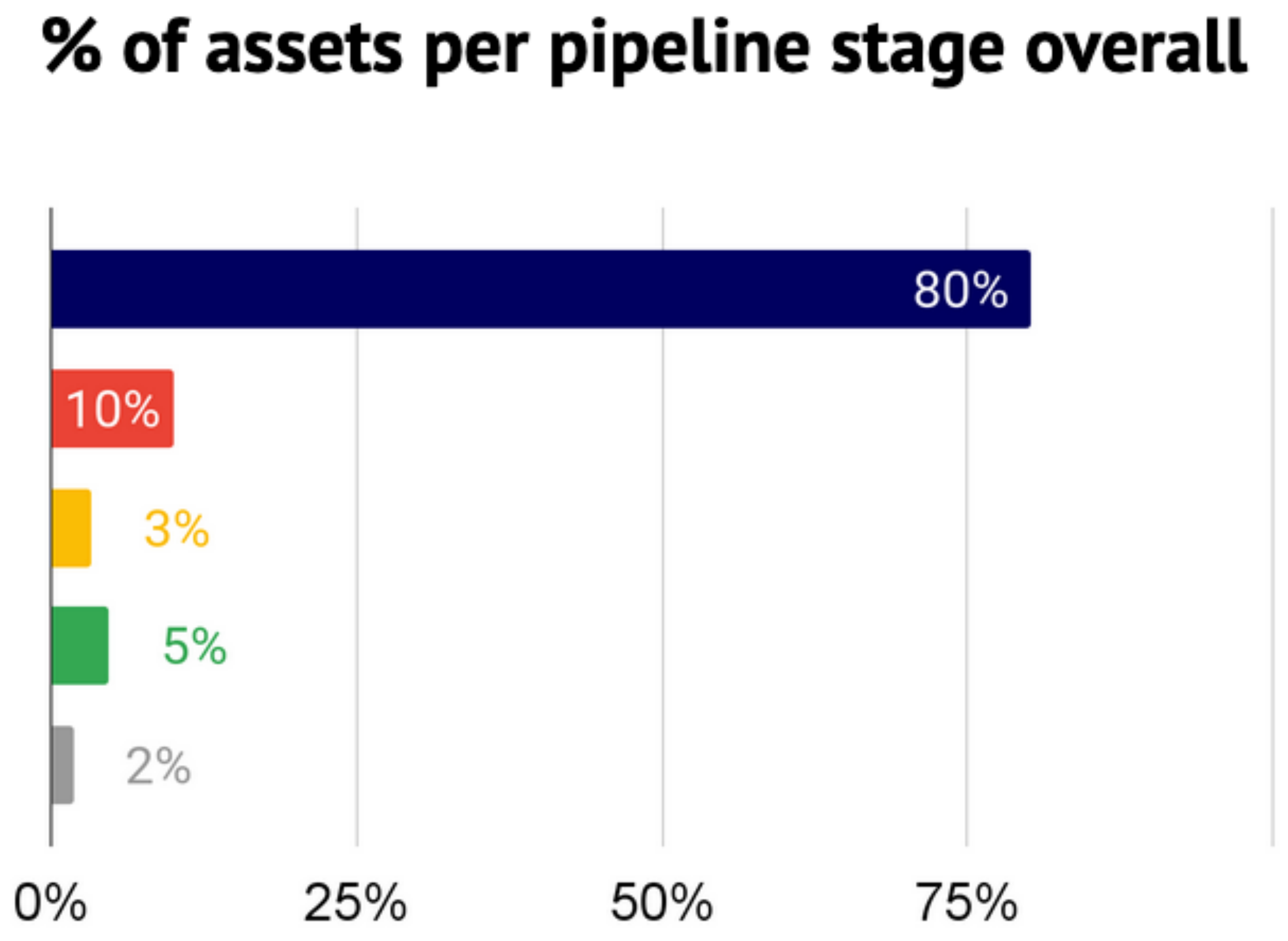
(Gaudelet et al, 2021)

AI-first Drug Design – reducing the cost?

>18 assets from AI DD companies now in trails



# of assets per pipeline stage per company



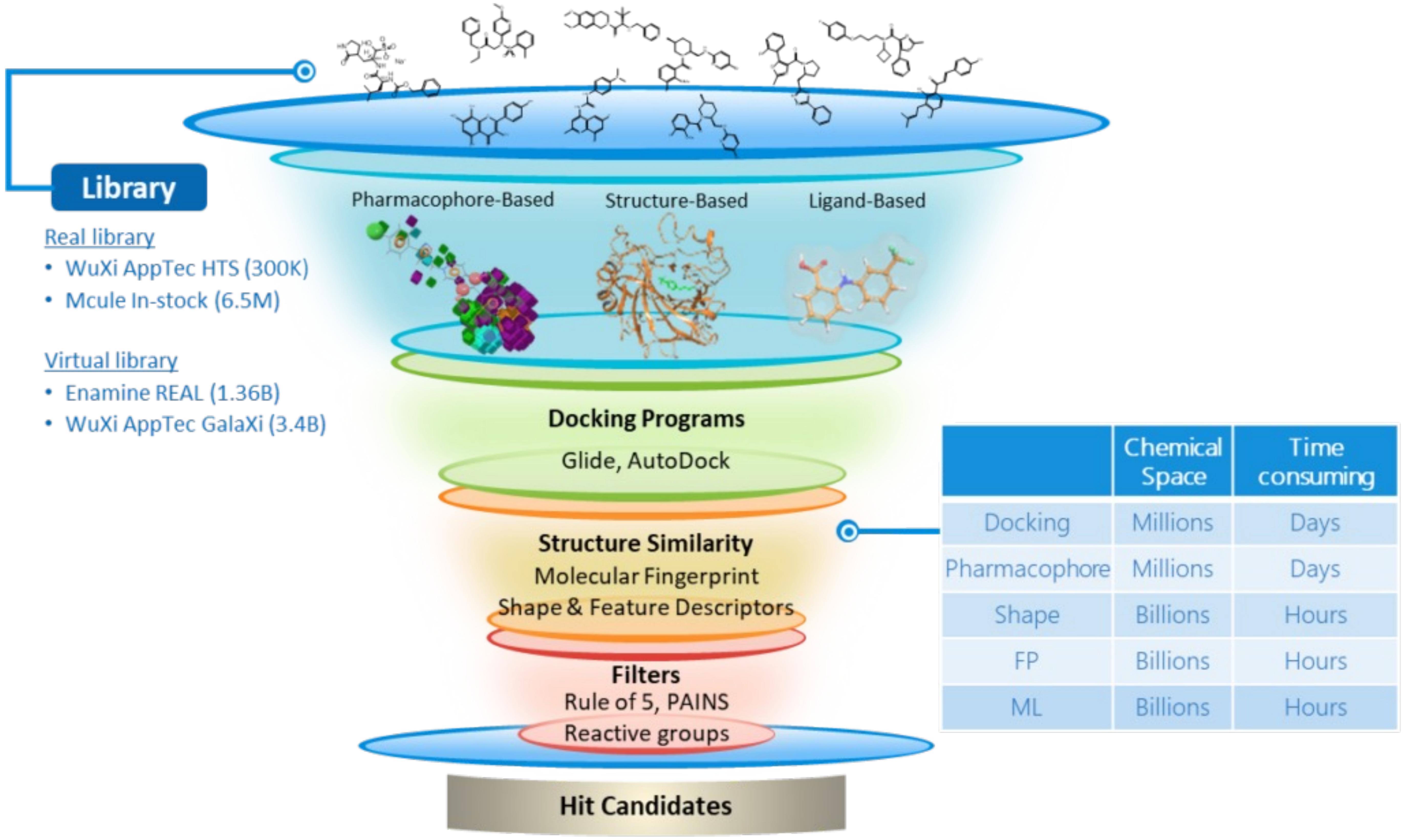
% of assets per pipeline stage overall

Image source – State of AI Report

1. Drug Design: Current paradigm

Virtual Screening

Efficiently searching large chemical space to find hits



Library

Real library

- WuXi AppTec HTS (300K)
- Mcule In-stock (6.5M)

Virtual library

- Enamine REAL (1.36B)
- WuXi AppTec GalaXi (3.4B)

Docking Programs

Glide, AutoDock

Structure Similarity

Molecular Fingerprint

Shape & Feature Descriptors

Filters

Rule of 5, PAINS

Reactive groups

Hit Candidates

	Chemical Space	Time consuming
Docking	Millions	Days
Pharmacophore	Millions	Days
Shape	Billions	Hours
FP	Billions	Hours
ML	Billions	Hours

Image source – WuXi AppTec

💡Central idea:

Search the vast drug-like space using
virtual screening
to identify good starting points

Chemical libraries

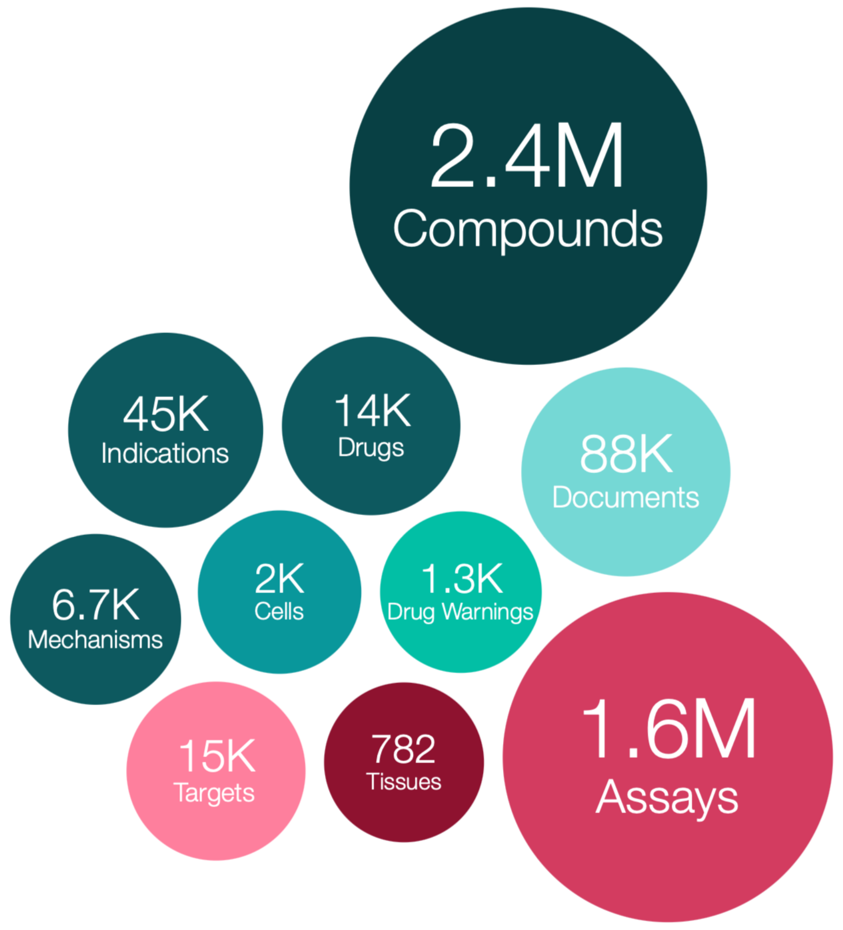
New chemical libraries are vast, diverse and commercial available

ZINC

Most common virtual library. ~1.3 billion purchasable molecules by pooling together 310 commercial catalogs

ChEMBL

2.6 million manually curated database of bioactive molecules with drug-like properties



Enamine REAL

(Relatively) small library of ‘building-blocks can be combined combinatorially to make a vast chemical space

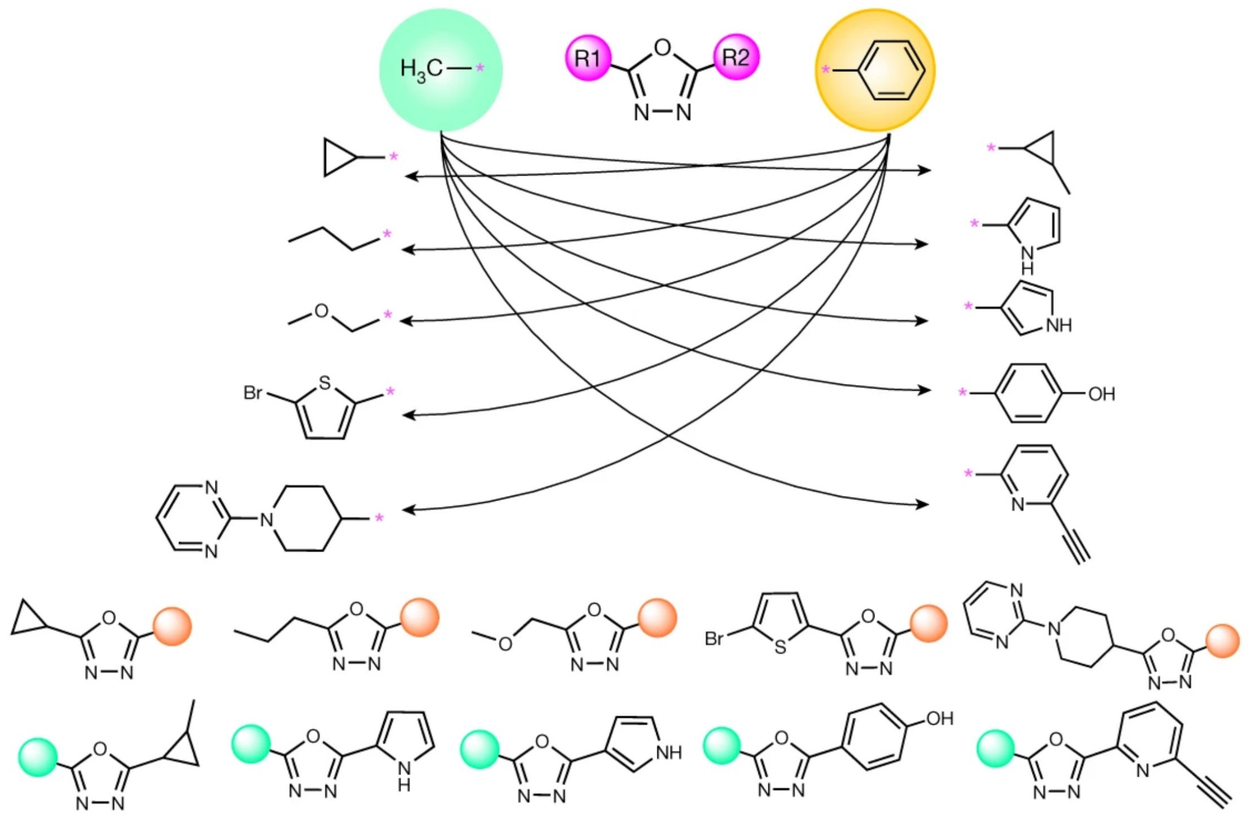


Image sources – Irwin et al, 2020, EBI, Sadybekov et al, 2021

Protein-ligand docking

Approximates protein-ligand complementarity and affinity

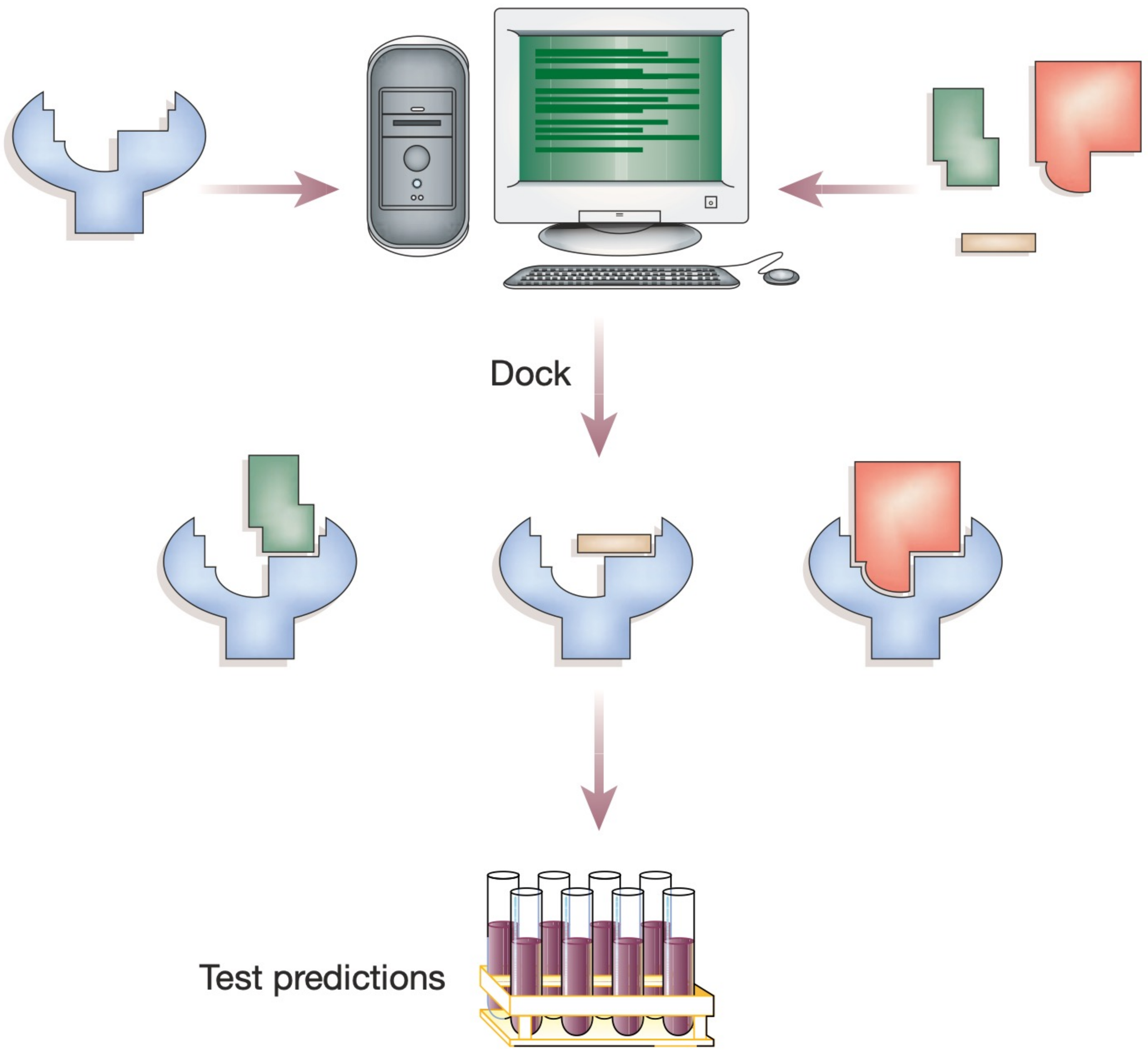
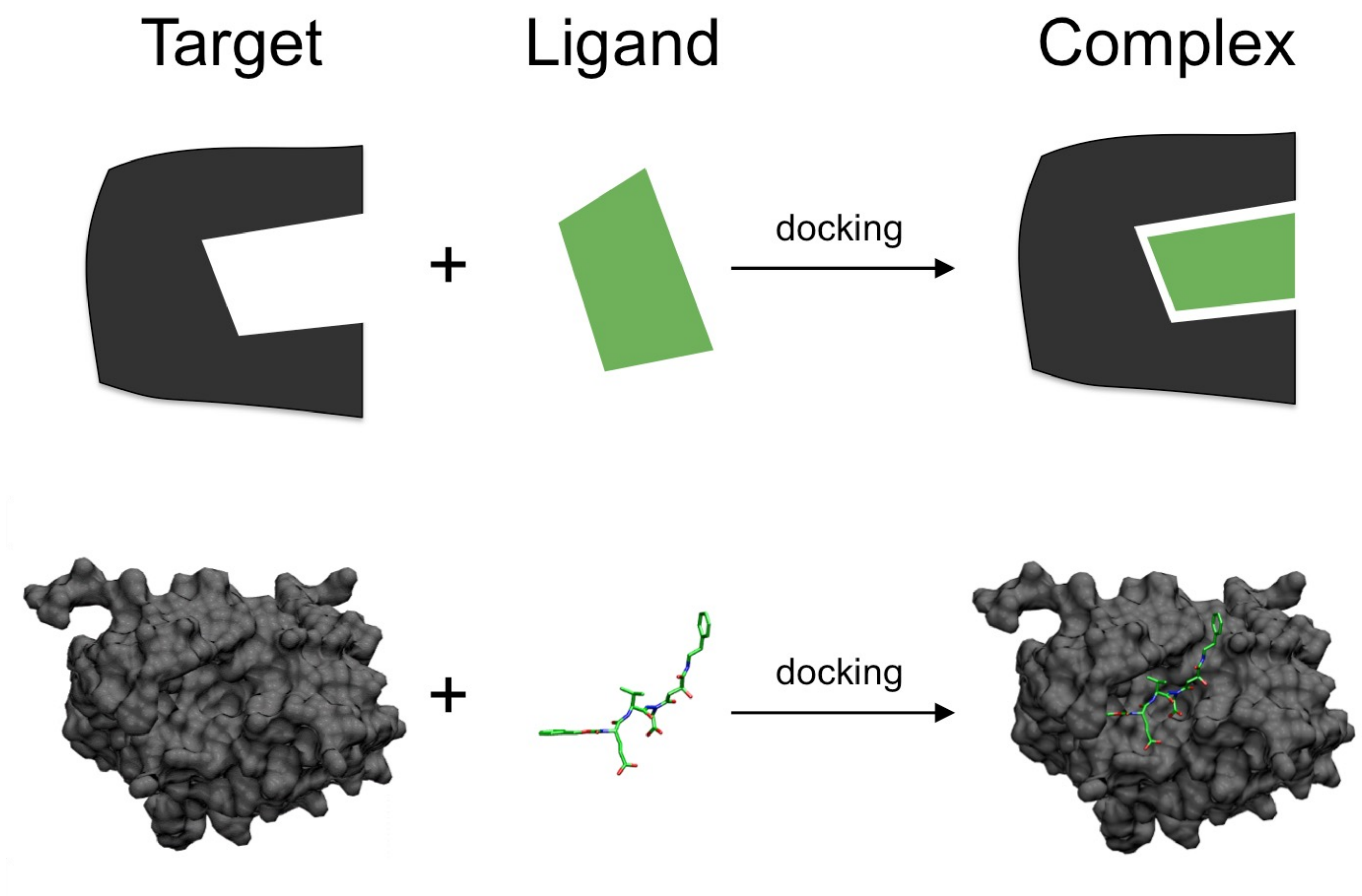


Image source – Schoichet

Protein-ligand docking

Combines 2 techniques: a scoring function and optimisation



Target

Ligand

Complex

+

docking

+

docking

Optimisation algorithm

Common example is Vina

$$\triangle\, G_{bind} = \triangle\, G_{solvent} + \triangle\, G_{conf} + \triangle\, G_{int} + \triangle\, G_{rot} + \triangle\, G_{t/t} + \triangle\, G_{vib}$$

Optimisation algorithm

Usually a mix of local and global search

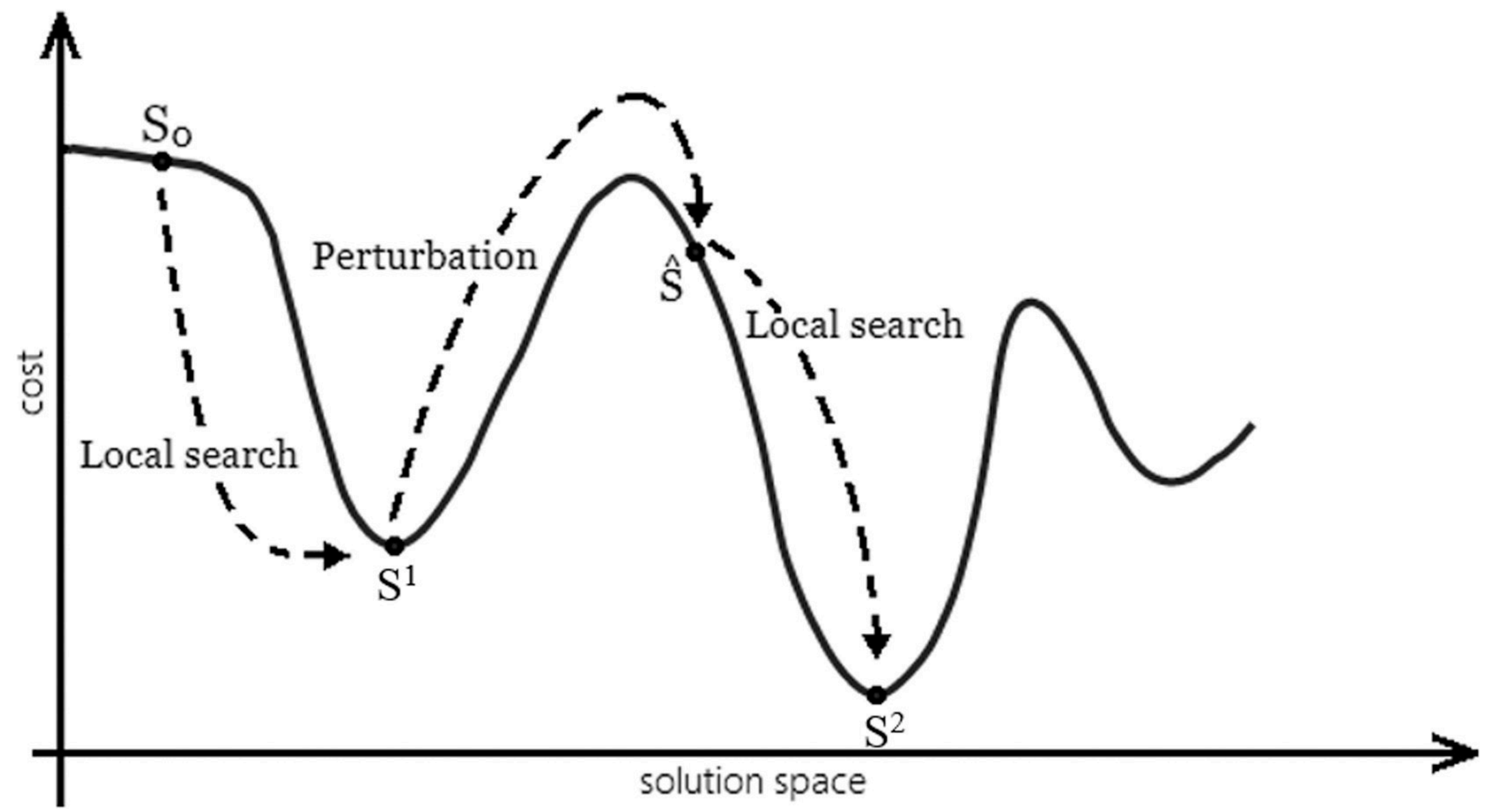
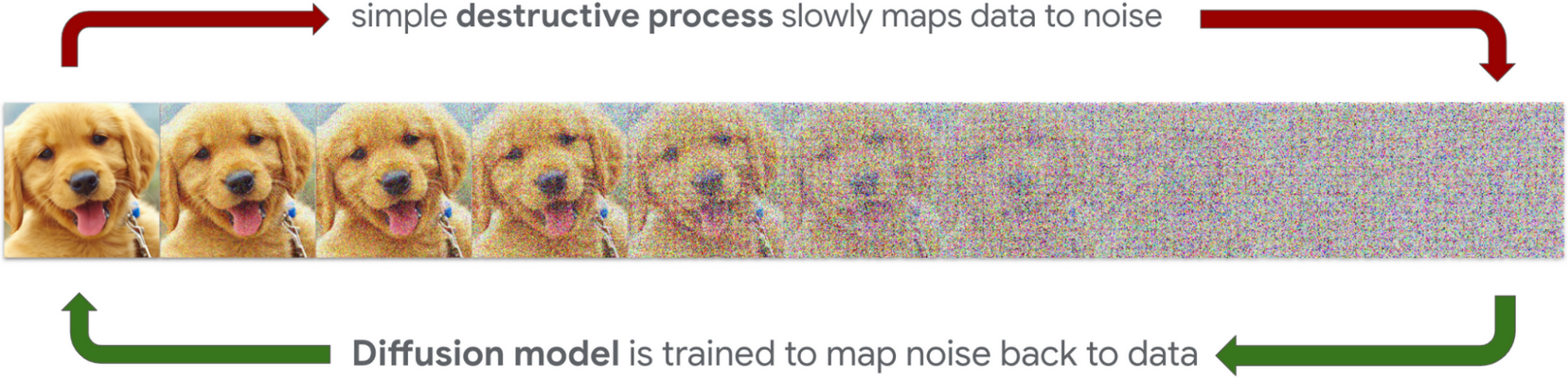


Image source – Cravo et al, 2022

1. DiffDock: Docking with deep learning

Recap: Diffusion Models

Mapping noise back to data



simple **destructive process** slowly maps data to noise

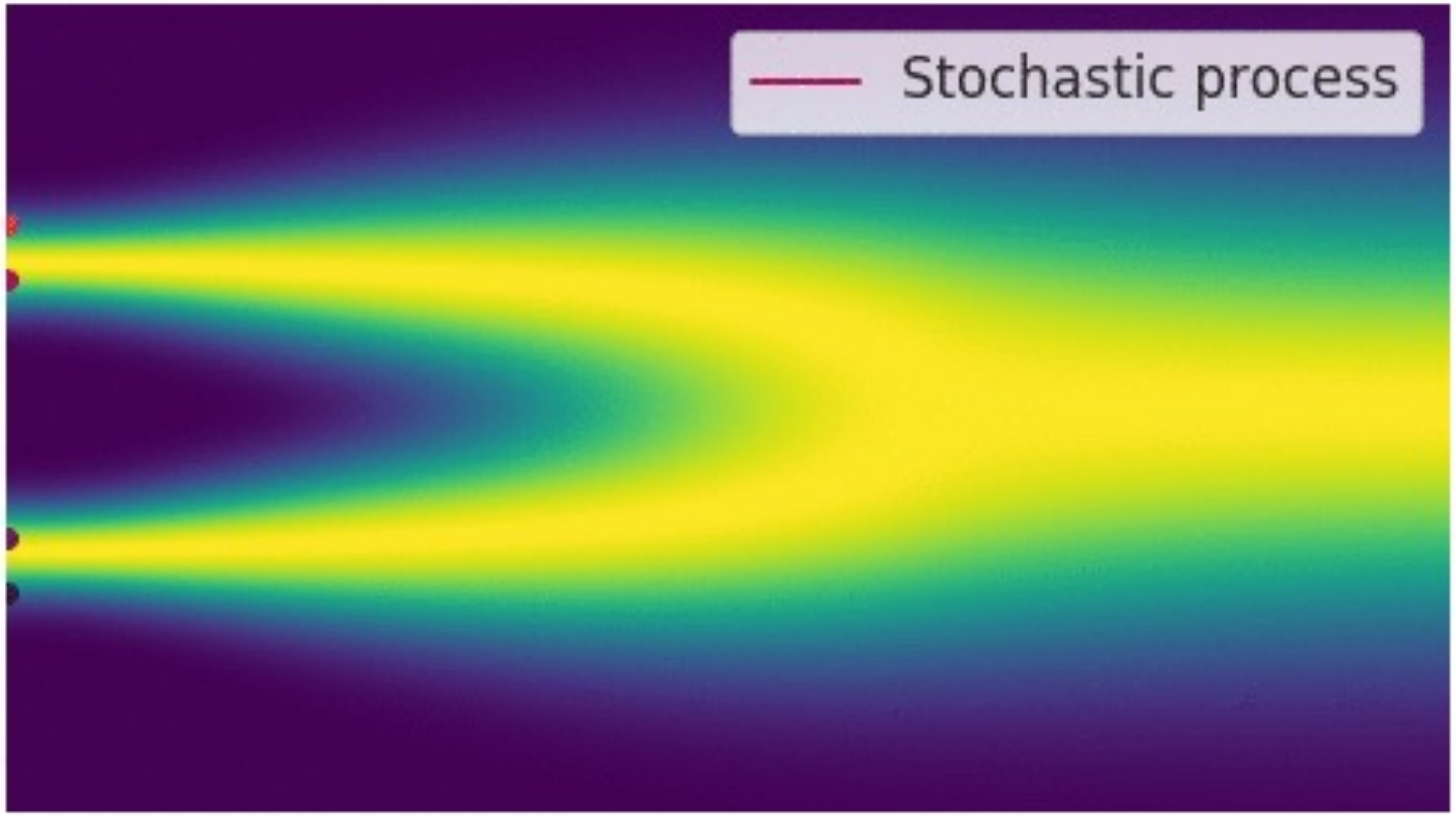
Diffusion model is trained to map noise back to data

$$L_t^{\text{simple}} = \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\boldsymbol{\epsilon}_t}\left[\left\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right\|^2\right]$$

Google Research, 2022 & Beyond: Language, Vision and Generative Models (Google Research)

Recap: Diffusion Models

Forward Process = Noising to a reference distribution

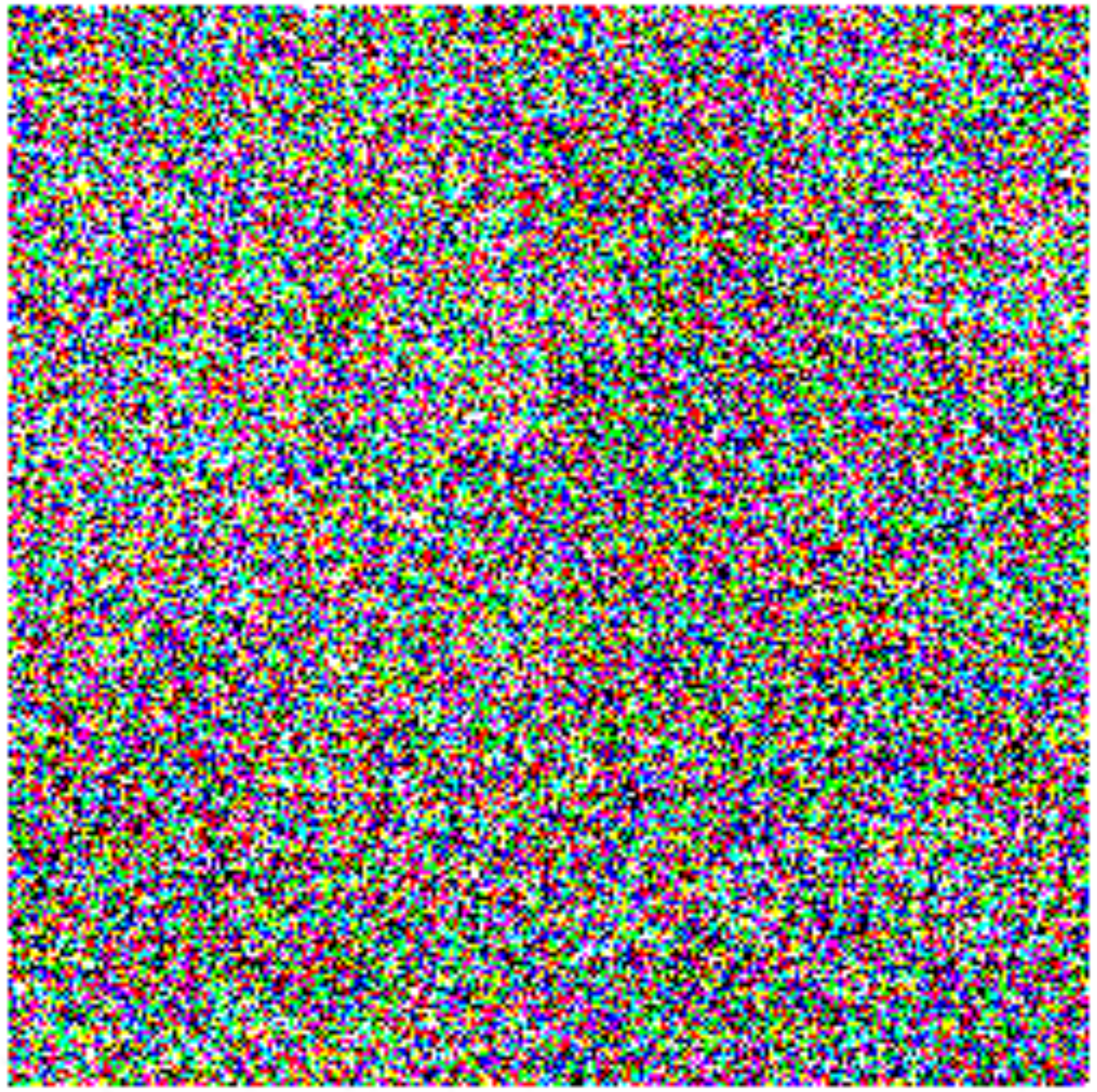


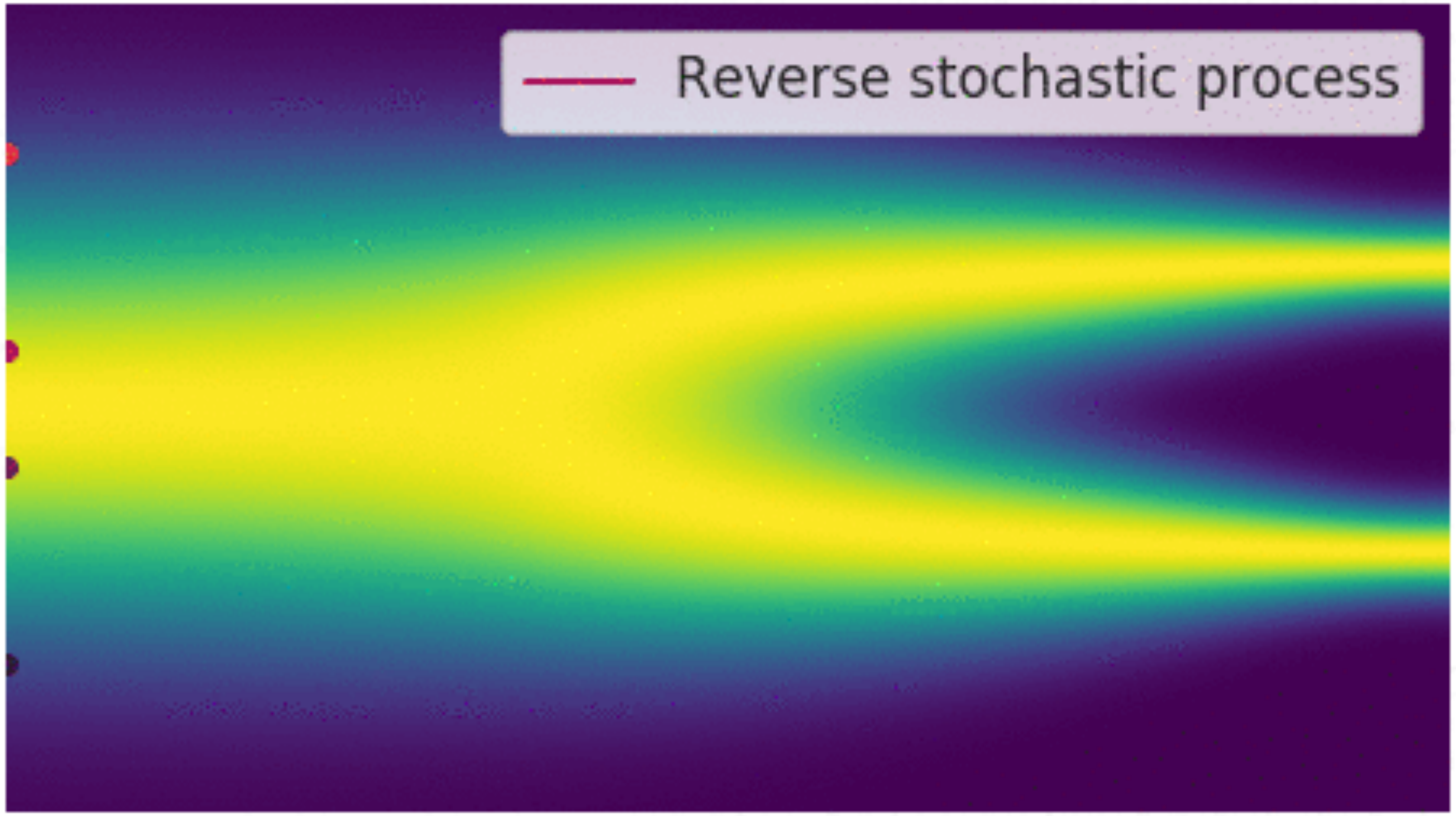
$$L_t^{\text{simple}} = \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\boldsymbol{\epsilon}_t}\left[\left\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right\|^2\right]$$

lilianweng.github.io

Recap: Diffusion Models

Reverse Process: Denoising to our target distribution

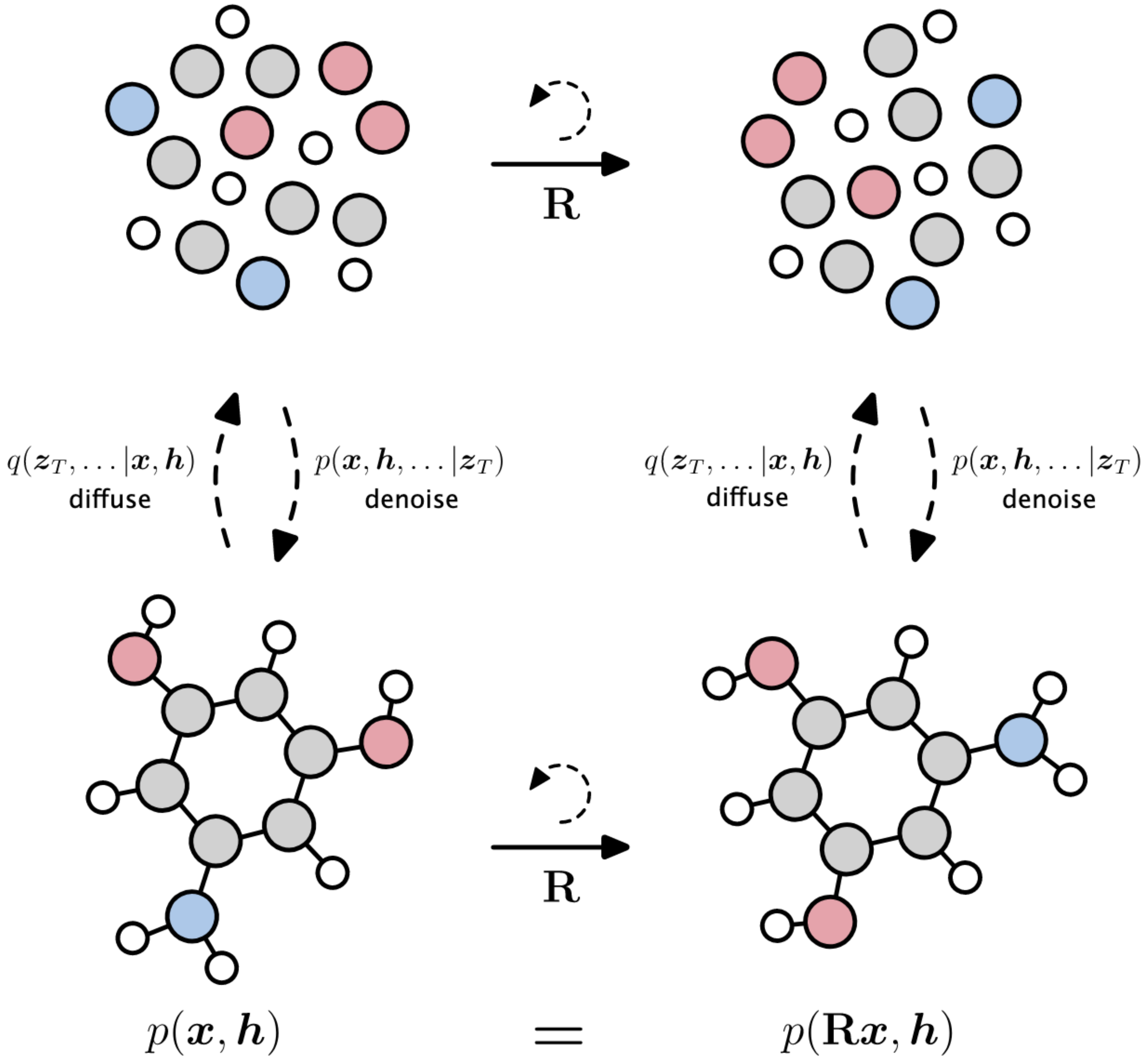




lilianweng.github.io

Recap: Geometric Deep Learning

GDL is the application of deep learning to objects that inhabit geometric domains (spaces)



R
$q(\boldsymbol{z}_T, \ldots | \boldsymbol{x}, \boldsymbol{h})$
diffuse
$p(\boldsymbol{x}, \boldsymbol{h}, \ldots | \boldsymbol{z}_T)$
denoise
$q(\boldsymbol{z}_T, \ldots | \boldsymbol{x}, \boldsymbol{h})$
diffuse
$p(\boldsymbol{x}, \boldsymbol{h}, \ldots | \boldsymbol{z}_T)$
denoise
R
$p(\boldsymbol{x}, \boldsymbol{h})$
=
$p(\mathbf{R}\boldsymbol{x}, \boldsymbol{h})$

Background

GeoDiff – early work on conformer generation using diffusion models

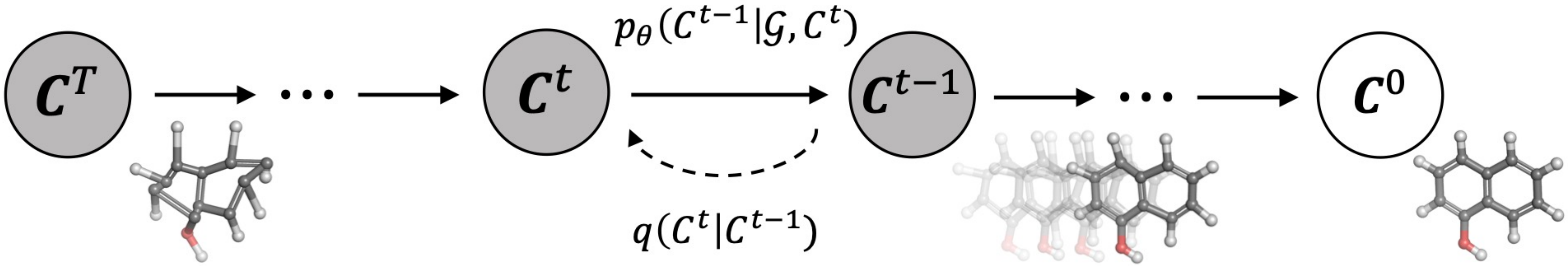


Image source – Xu et al, ICLR 2022

Torsion Diffusion

Simplified to only generate the degrees of freedom in a molecule (torsion angles)

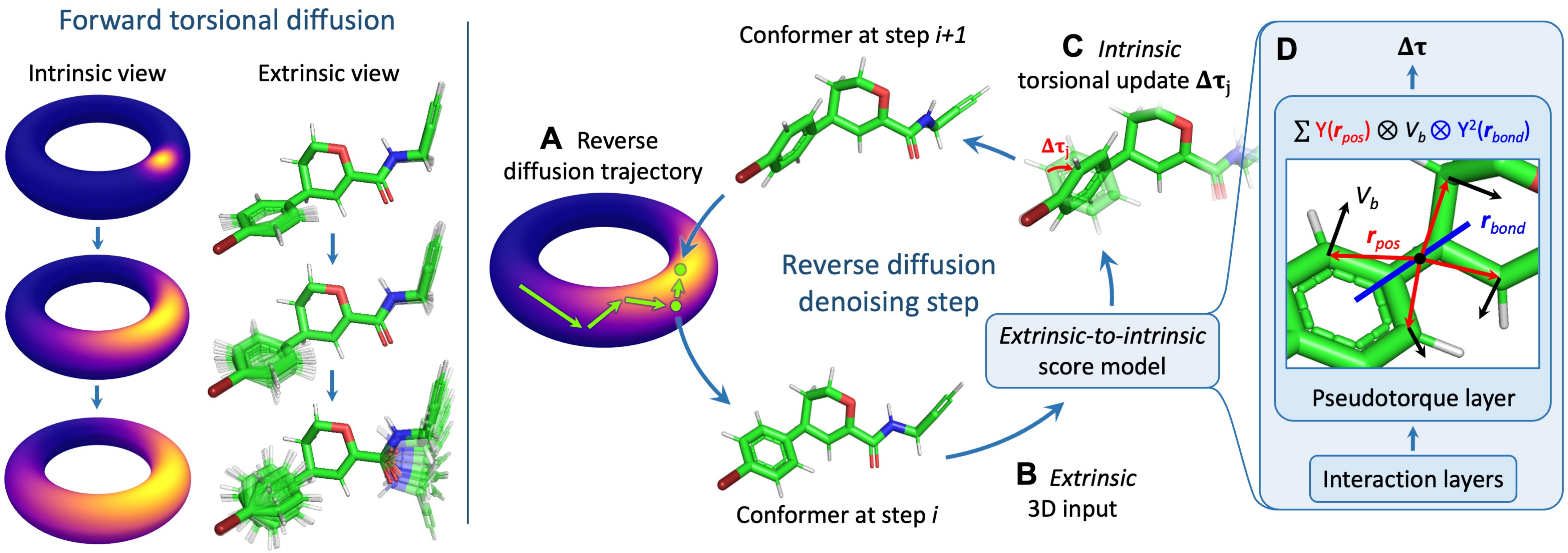
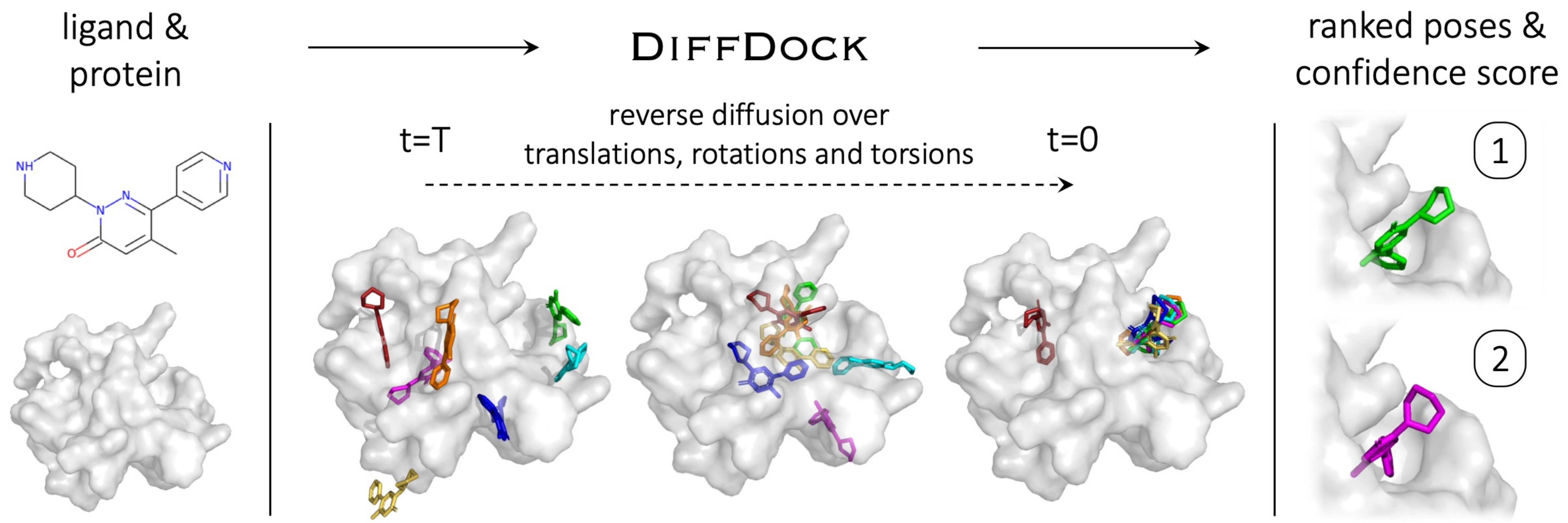

Forward torsional diffusion
Intrinsic view
Extrinsic view
A Reverse diffusion trajectory
Conformer at step *i+1*
C *Intrinsic* torsional update $\Delta\tau_j$
$\Delta\tau_j$
D
$\Delta\tau$
$\sum Y(r_{pos}) \otimes V_b \otimes Y^2(r_{bond})$
V_b
r_{pos}
r_{bond}
Pseudotorque layer
Interaction layers
Reverse diffusion denoising step
Extrinsic-to-intrinsic score model
Conformer at step *i*
B *Extrinsic* 3D input

Image source – [Jing et al, NeurIPS 2022]

DiffDock

Docking = Torsional Diffusion + SE(3) diffusion (global rotations and translations)


ligand & protein

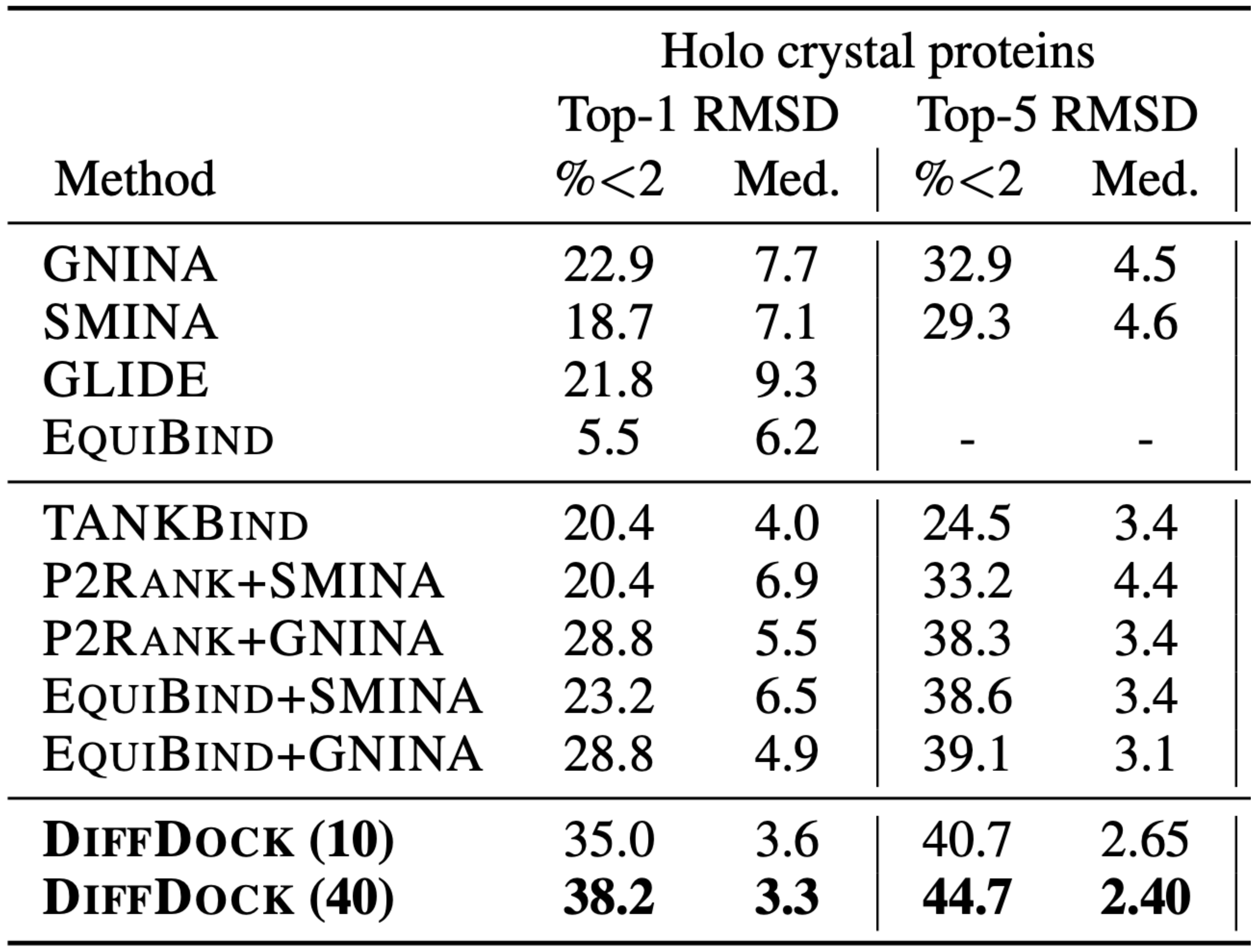
DiffDock

ranked poses & confidence score

t=T

reverse diffusion over translations, rotations and torsions

t=0

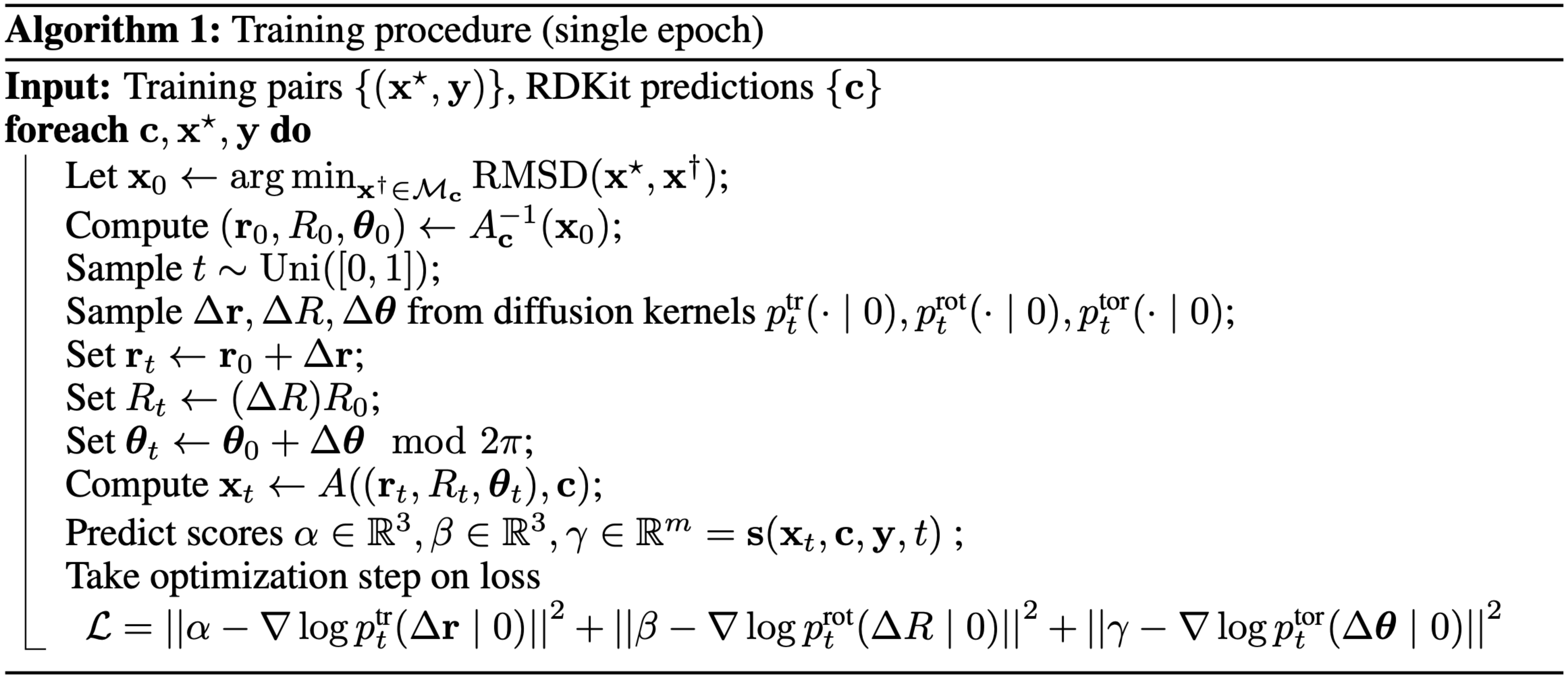


	Holo crystal proteins			
	Top-1 RMSD		Top-5 RMSD	
Method	%<2	Med.	%<2	Med.
GNINA	22.9	7.7	32.9	4.5
SMINA	18.7	7.1	29.3	4.6
GLIDE	21.8	9.3		
EquiBind	5.5	6.2	-	-
TANKBind	20.4	4.0	24.5	3.4
P2Rank+SMINA	20.4	6.9	33.2	4.4
P2Rank+GNINA	28.8	5.5	38.3	3.4
EquiBind+SMINA	23.2	6.5	38.6	3.4
EquiBind+GNINA	28.8	4.9	39.1	3.1
DiffDock (10)	35.0	3.6	40.7	2.65
DiffDock (40)	**38.2**	**3.3**	**44.7**	**2.40**

Image source – Corso et al, ICLR 2023

DiffDock

Training by learning translational, rotational and torsional diffusion kernels

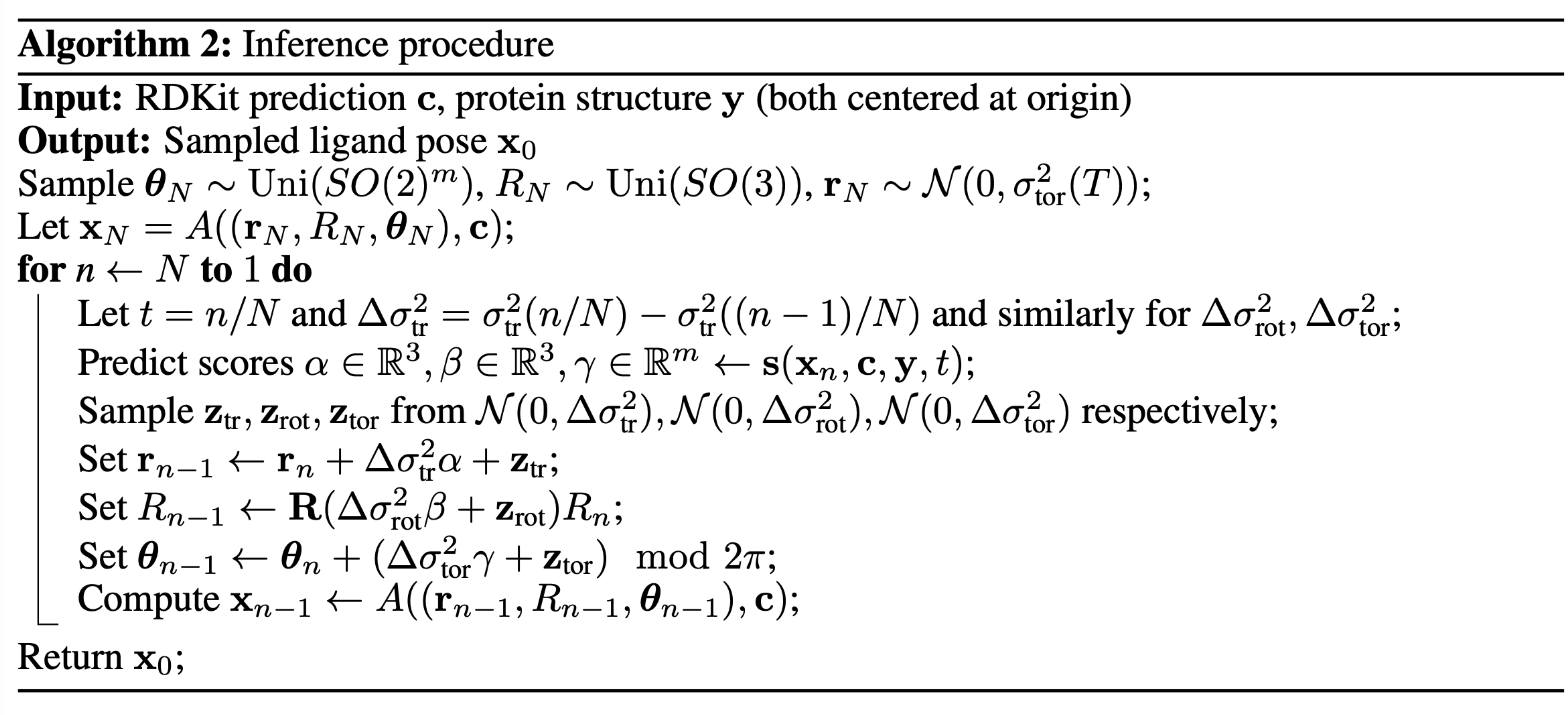


```
Algorithm 1: Training procedure (single epoch)
Input: Training pairs $\{(\mathbf{x}^\star, \mathbf{y})\}$, RDKit predictions $\{\mathbf{c}\}$
foreach $\mathbf{c}, \mathbf{x}^\star, \mathbf{y}$ do
    Let $\mathbf{x}_0 \leftarrow \arg\min_{\mathbf{x}^\dagger \in \mathcal{M}_\mathbf{c}} \mathrm{RMSD}(\mathbf{x}^\star, \mathbf{x}^\dagger)$;
    Compute $(\mathbf{r}_0, R_0, \boldsymbol{\theta}_0) \leftarrow A_\mathbf{c}^{-1}(\mathbf{x}_0)$;
    Sample $t \sim \mathrm{Uni}([0, 1])$;
    Sample $\Delta\mathbf{r}, \Delta R, \Delta\boldsymbol{\theta}$ from diffusion kernels $p_t^{\mathrm{tr}}(\cdot \mid 0), p_t^{\mathrm{rot}}(\cdot \mid 0), p_t^{\mathrm{tor}}(\cdot \mid 0)$;
    Set $\mathbf{r}_t \leftarrow \mathbf{r}_0 + \Delta\mathbf{r}$;
    Set $R_t \leftarrow (\Delta R) R_0$;
    Set $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_0 + \Delta\boldsymbol{\theta} \mod 2\pi$;
    Compute $\mathbf{x}_t \leftarrow A((\mathbf{r}_t, R_t, \boldsymbol{\theta}_t), \mathbf{c})$;
    Predict scores $\alpha \in \mathbb{R}^3, \beta \in \mathbb{R}^3, \gamma \in \mathbb{R}^m = \mathbf{s}(\mathbf{x}_t, \mathbf{c}, \mathbf{y}, t)$ ;
    Take optimization step on loss
      $\mathcal{L} = ||\alpha - \nabla \log p_t^{\mathrm{tr}}(\Delta\mathbf{r} \mid 0)||^2 + ||\beta - \nabla \log p_t^{\mathrm{rot}}(\Delta R \mid 0)||^2 + ||\gamma - \nabla \log p_t^{\mathrm{tor}}(\Delta\boldsymbol{\theta} \mid 0)||^2$
```

Image source – Corso et al, ICLR 2023

DiffDock

Training by learning translational, rotational and torsional diffusion kernels

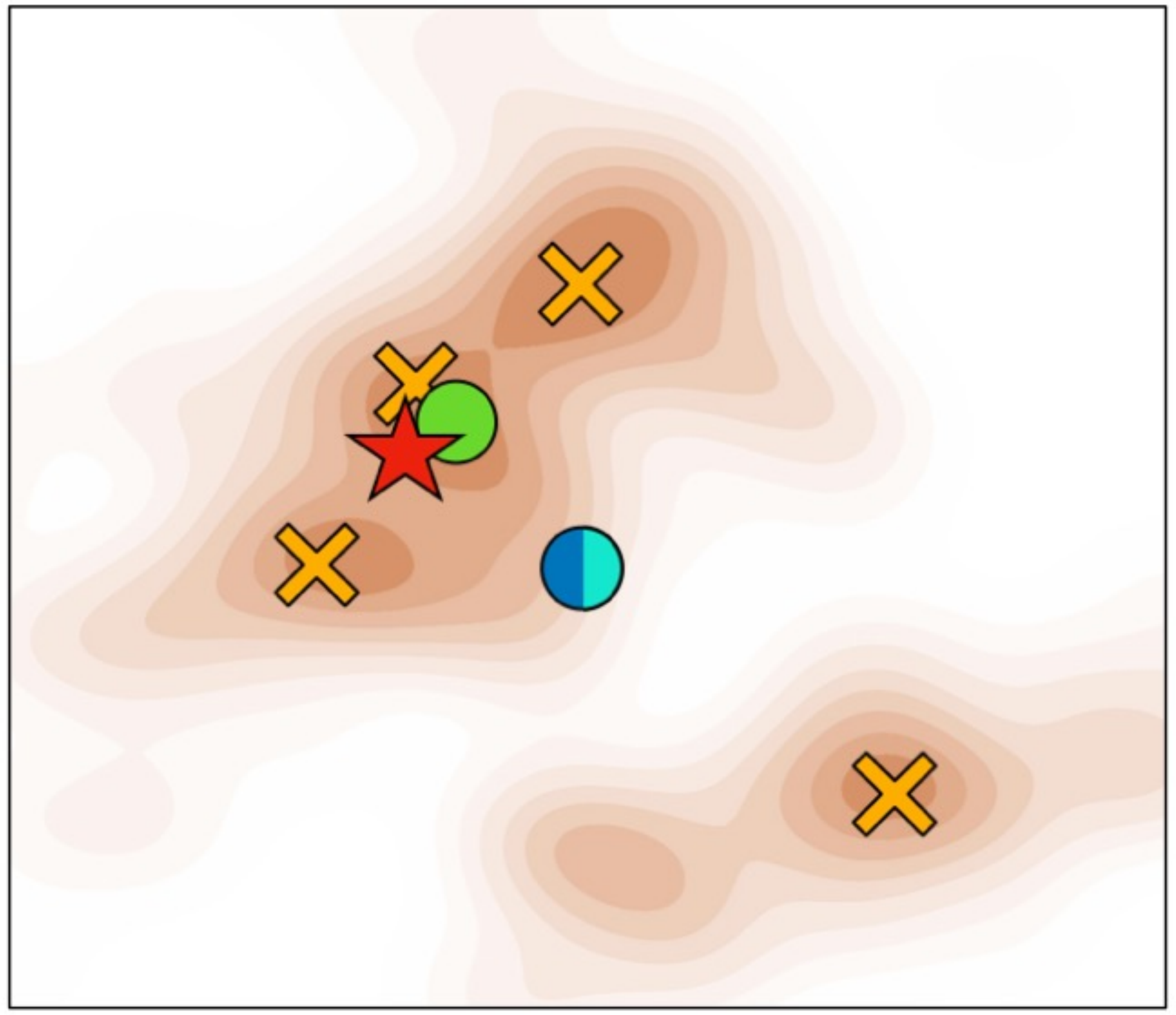


```
Algorithm 2: Inference procedure
Input: RDKit prediction $\mathbf{c}$, protein structure $\mathbf{y}$ (both centered at origin)
Output: Sampled ligand pose $\mathbf{x}_0$
Sample $\boldsymbol{\theta}_N \sim \text{Uni}(SO(2)^m)$, $R_N \sim \text{Uni}(SO(3))$, $\mathbf{r}_N \sim \mathcal{N}(0, \sigma^2_{\text{tor}}(T))$;
Let $\mathbf{x}_N = A((\mathbf{r}_N, R_N, \boldsymbol{\theta}_N), \mathbf{c})$;
for $n \leftarrow N$ to 1 do
    Let $t = n/N$ and $\Delta\sigma^2_{\text{tr}} = \sigma^2_{\text{tr}}(n/N) - \sigma^2_{\text{tr}}((n-1)/N)$ and similarly for $\Delta\sigma^2_{\text{rot}}, \Delta\sigma^2_{\text{tor}}$;
    Predict scores $\alpha \in \mathbb{R}^3, \beta \in \mathbb{R}^3, \gamma \in \mathbb{R}^m \leftarrow \mathbf{s}(\mathbf{x}_n, \mathbf{c}, \mathbf{y}, t)$;
    Sample $\mathbf{z}_{\text{tr}}, \mathbf{z}_{\text{rot}}, \mathbf{z}_{\text{tor}}$ from $\mathcal{N}(0, \Delta\sigma^2_{\text{tr}}), \mathcal{N}(0, \Delta\sigma^2_{\text{rot}}), \mathcal{N}(0, \Delta\sigma^2_{\text{tor}})$ respectively;
    Set $\mathbf{r}_{n-1} \leftarrow \mathbf{r}_n + \Delta\sigma^2_{\text{tr}}\alpha + \mathbf{z}_{\text{tr}}$;
    Set $R_{n-1} \leftarrow \mathbf{R}(\Delta\sigma^2_{\text{rot}}\beta + \mathbf{z}_{\text{rot}})R_n$;
    Set $\boldsymbol{\theta}_{n-1} \leftarrow \boldsymbol{\theta}_n + (\Delta\sigma^2_{\text{tor}}\gamma + \mathbf{z}_{\text{tor}}) \mod 2\pi$;
    Compute $\mathbf{x}_{n-1} \leftarrow A((\mathbf{r}_{n-1}, R_{n-1}, \boldsymbol{\theta}_{n-1}), \mathbf{c})$;
Return $\mathbf{x}_0$;
```

Image source – Corso et al, ICLR 2023

DiffDock

In theory, generative modelling allows us to approximate and sample from the whole binding landscape



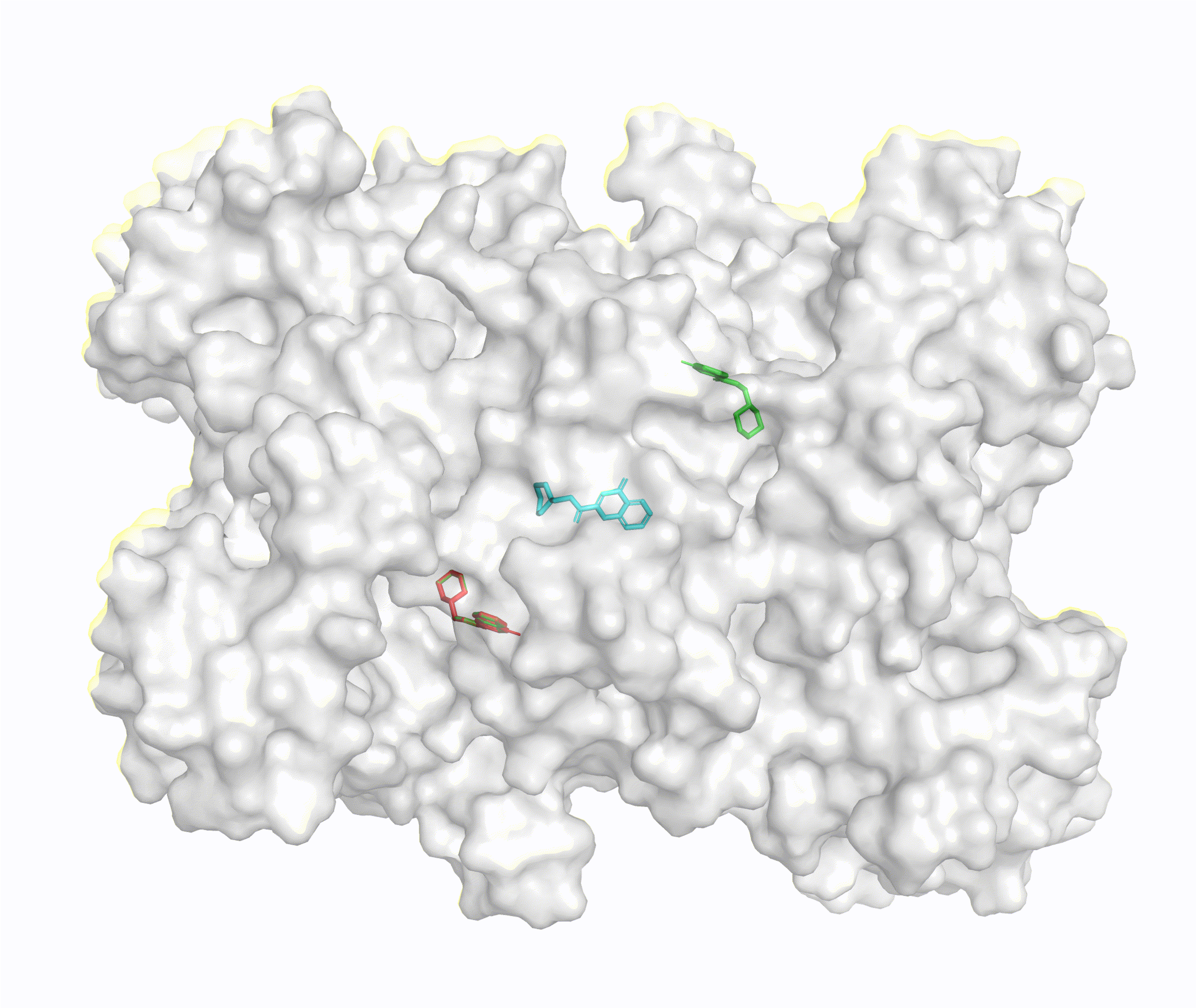


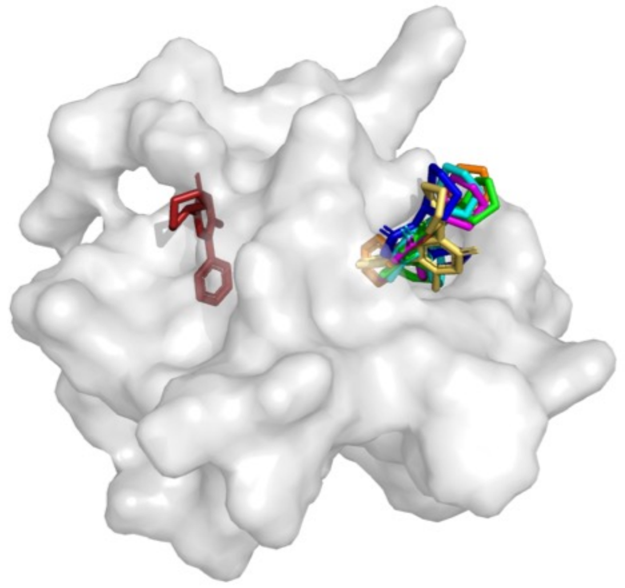
Image source – Corso et al, ICLR 2023

Limitations of DL docking are significant

Many substantial issues can be masked when only measuring performing by RMSD

Poor evaluations

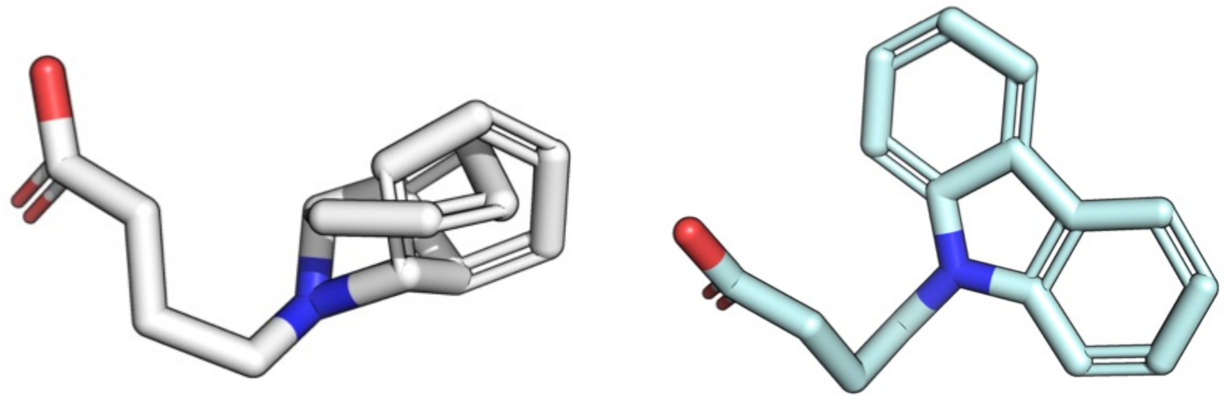
DL-based methods are often evaluated using blind docking, something conventional methods are not designed for



- When evaluated by docking molecules into known pockets, the traditional methods still outperform DiffDock
- DiffDock is actually a SOTA binding pocket prediction algorithm

Unrealistic molecules

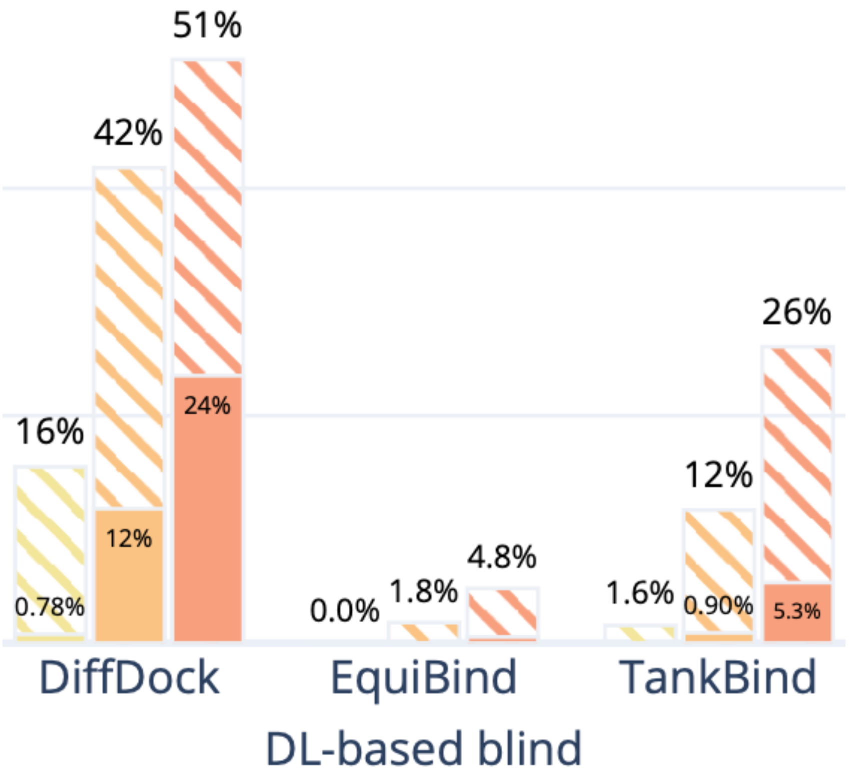
DL-based methods can have significant biophysical violations, even if docking RMSD is low



Recommended Reading:
- Martin Buttenschoen– PoseBusters

Poor generalisation

DL-based methods struggle to generalise to proteins not seen during training



- Generalisation to novel receptors and molecules is essential
- Issue also large in molecule scaffold generalisation but less studied

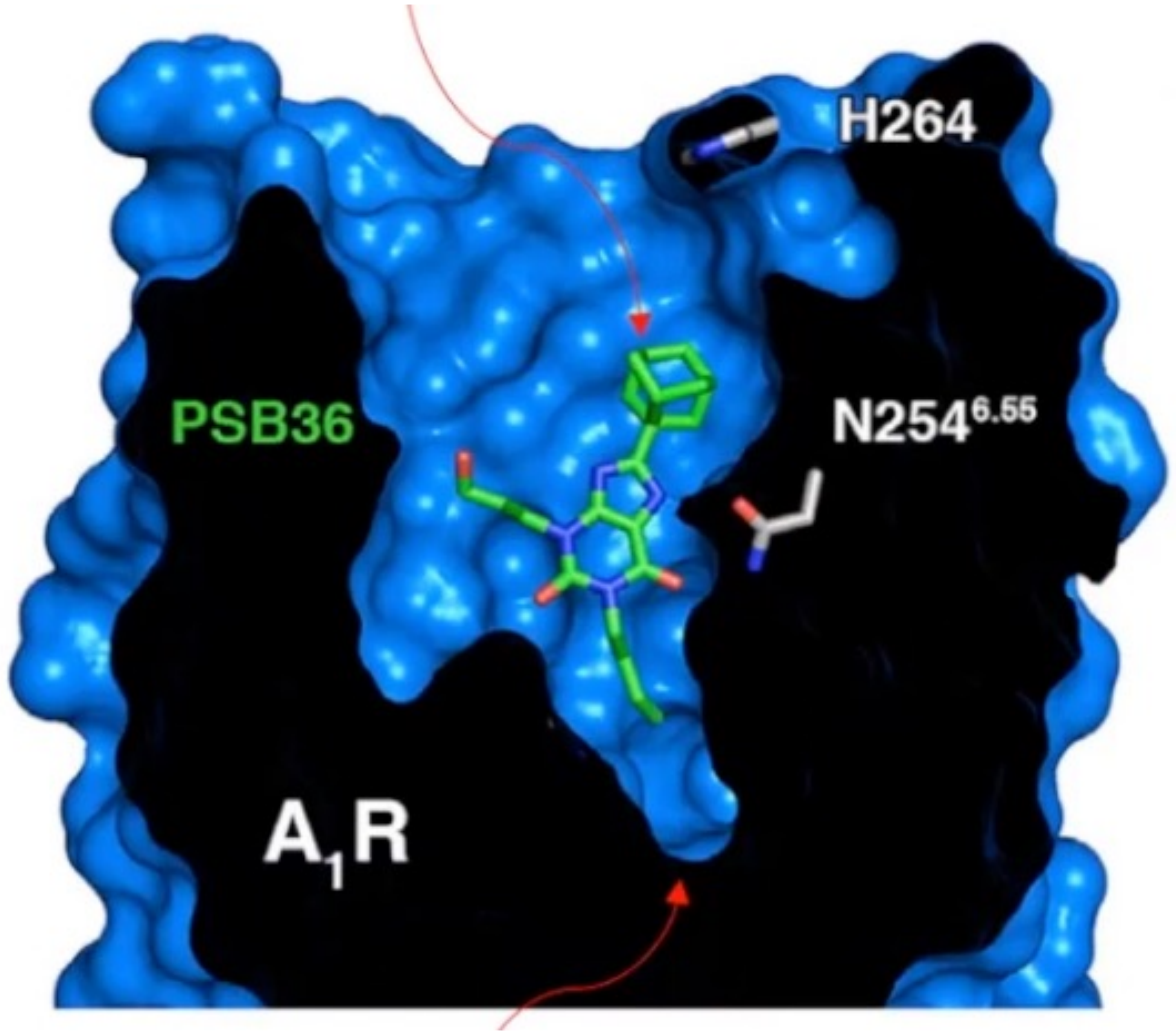
Recommended Reading:
- Martin Buttenschoen– PoseBusters

Image source – Corso et al, ICLR 2023, Buttenschoen et al, 2023

2. SBDD with Generative Models

SBDD with Generation Models

Rephrasing SBDD as learning a conditional probability distribution



$$\text{SBDD} = p(\text{molecule}|\text{receptor})$$

Image source – Chen et al, 2017

SBDD with Generation Models

Rephrasing SBDD as learning a conditional probability distribution

$$\mathrm{SBDD} = p(\mathrm{molecule}|\mathrm{receptor})$$

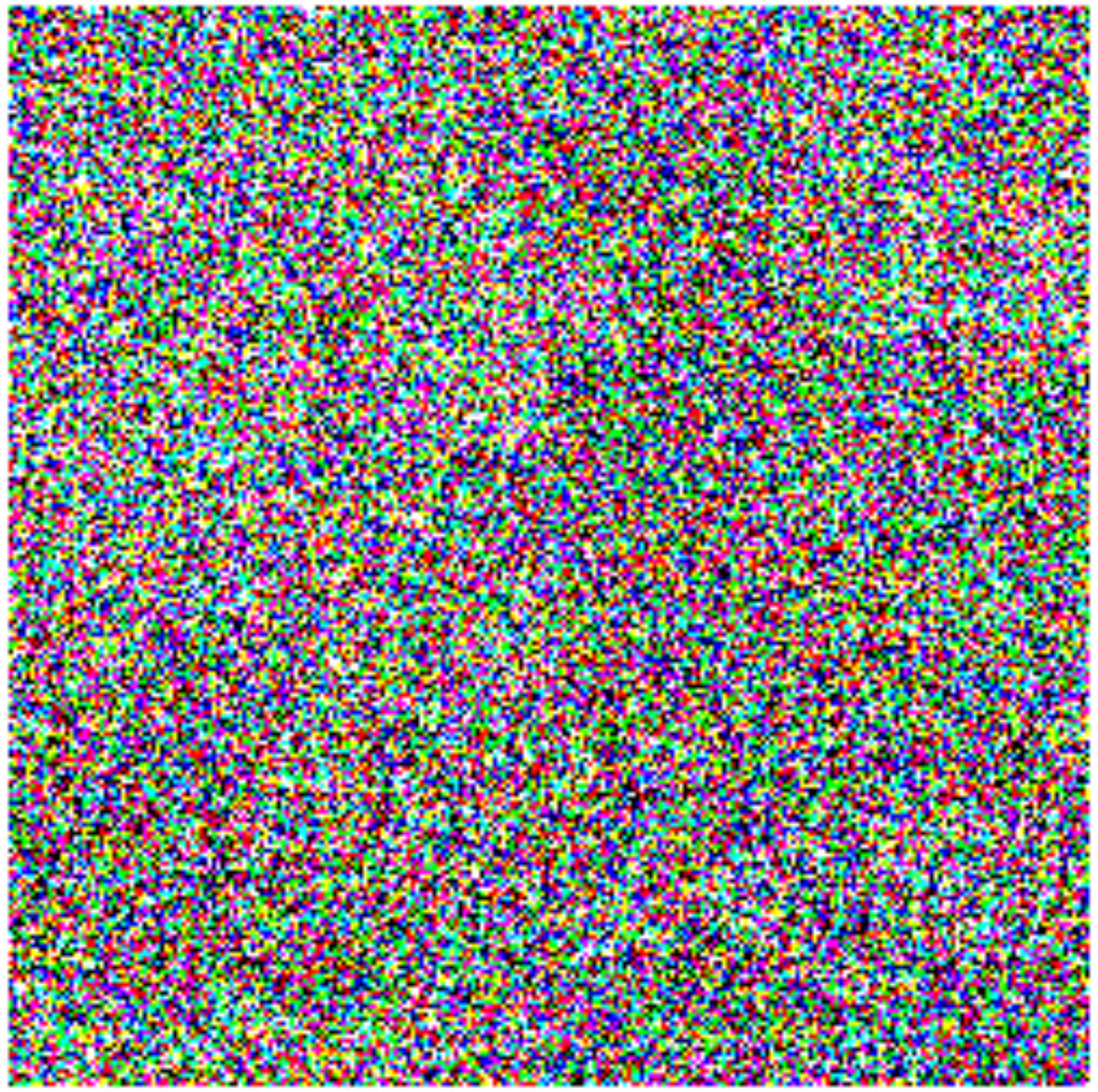
Image source – Chen et al, 2017

💡 Central idea:

Treat drug design as a condition generation problem by learning from data

SBDD with Diffusion Models

Learning to generate complimentary molecules in 3D



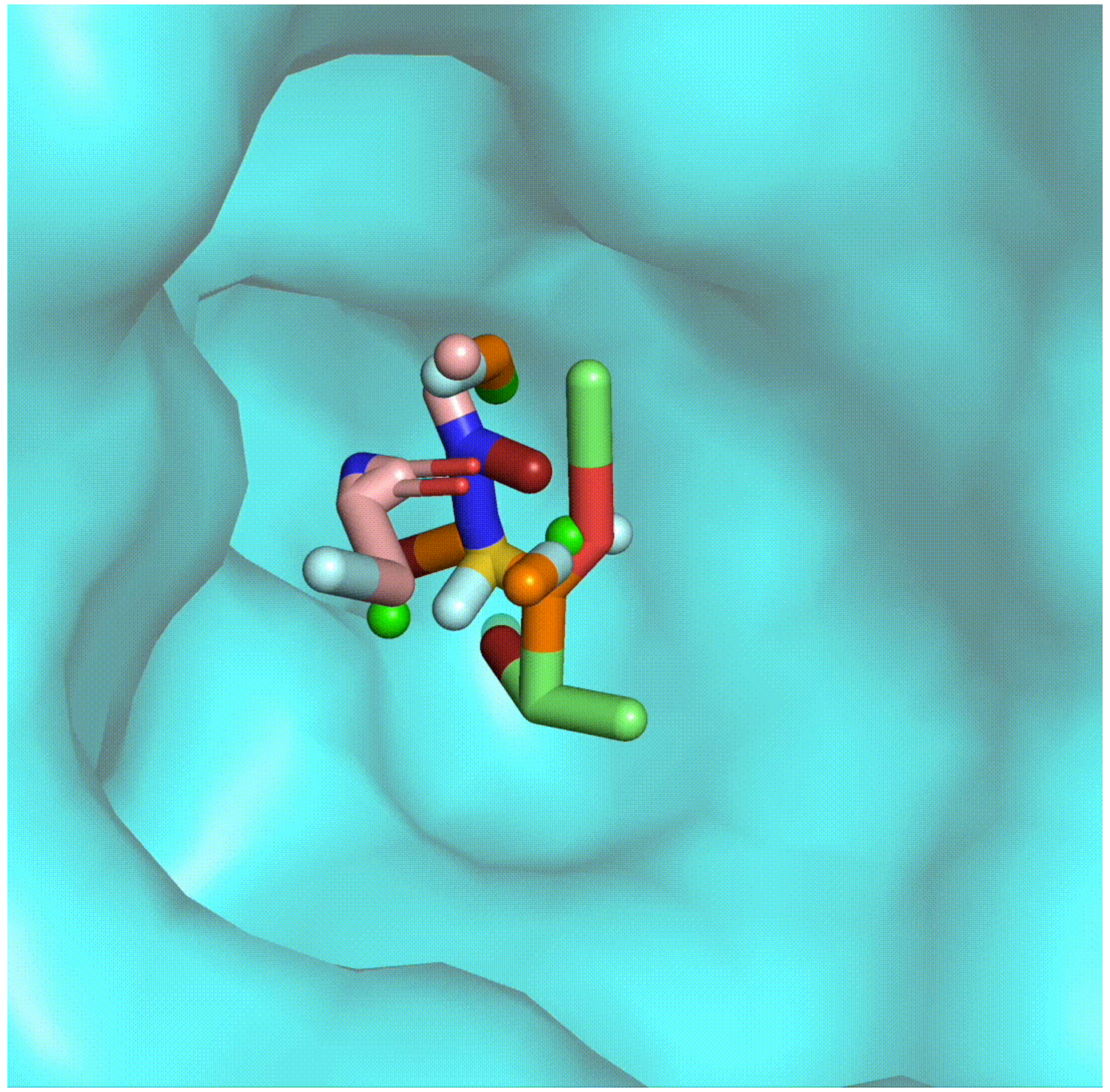
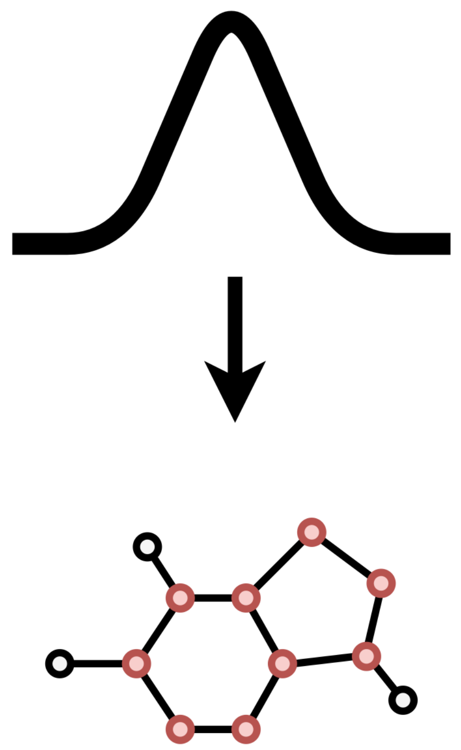


Image source – Schneuing et al, 2022

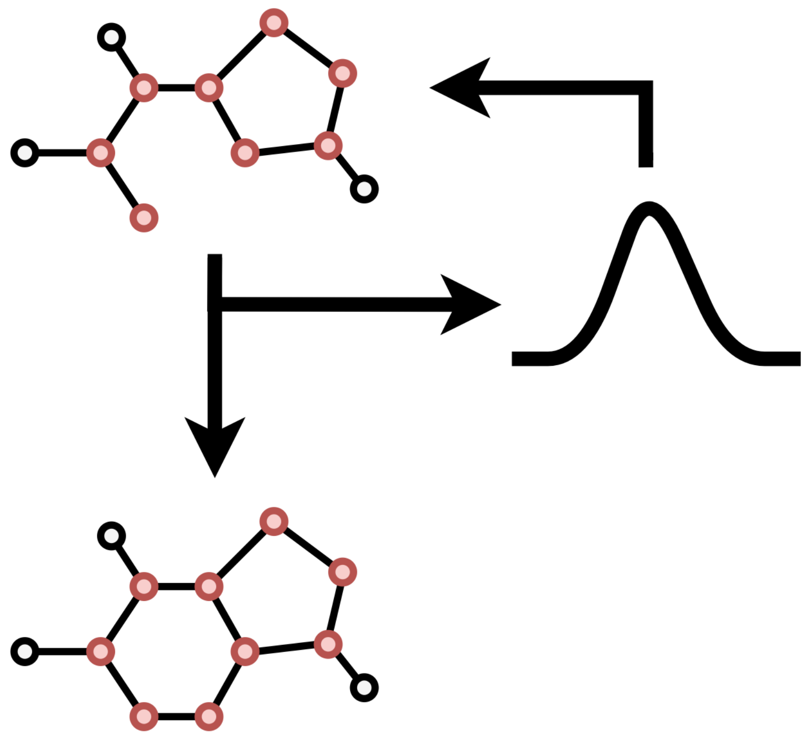
Generative Modelling for Molecule Generation

1. All-at-once (one-shot)



(Zang et al, 2020)

2. Node-by-node (autoregressive)



(Imrie et al, 2020)

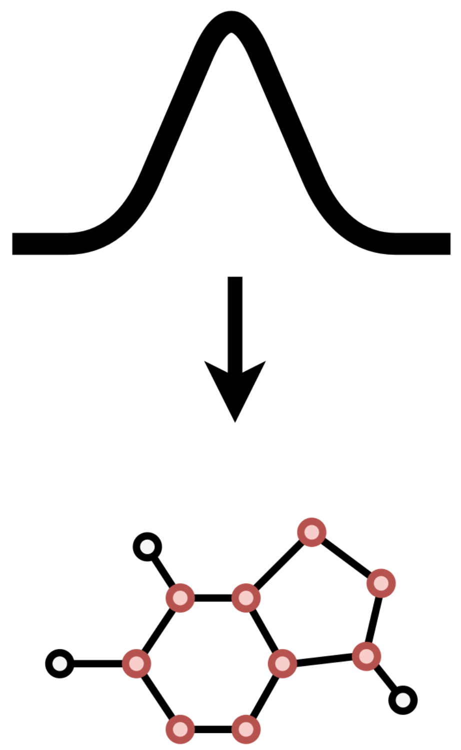
✗ Node independence assumed

✗ Arbitrary node ordering

✗ Different generation traces not equal

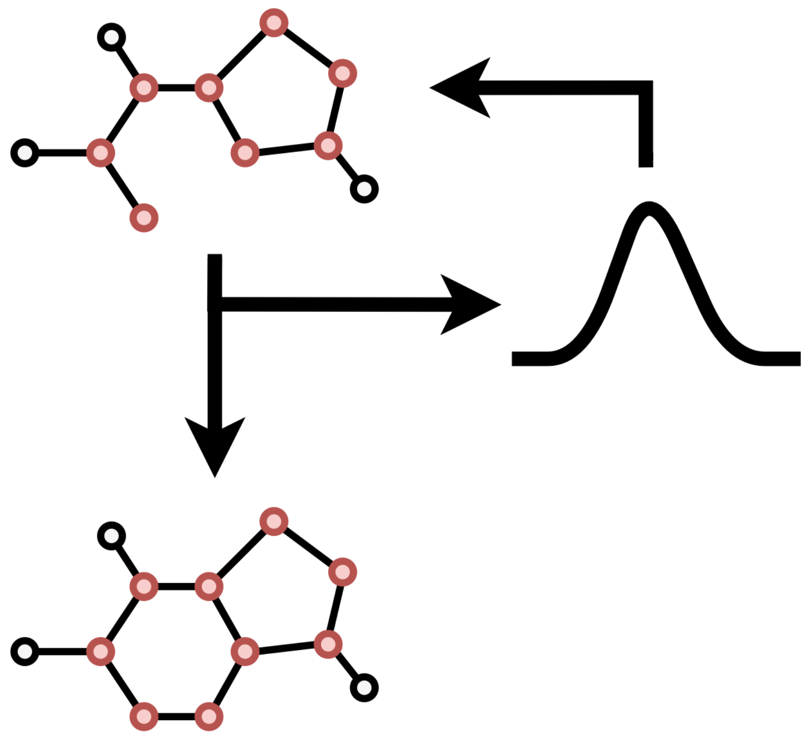
Generative Modelling for Molecule Generation

1. All-at-once (one-shot)



(Zang et al, 2020)

2. Node-by-node (autoregressive)



(Imrie et al, 2020)

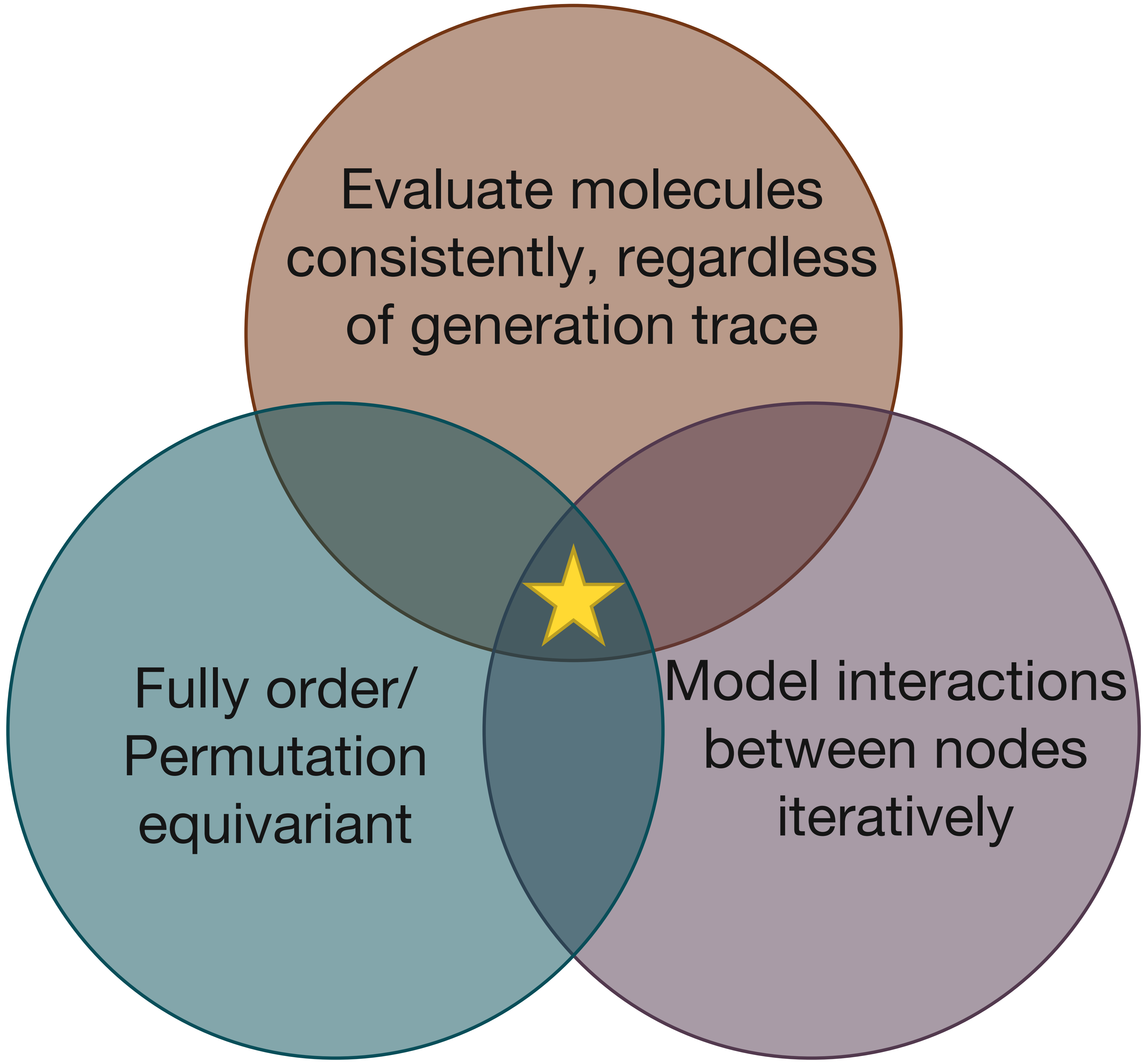
✗ Node independence assumed

✗ Arbitrary node ordering

✗ Different generation traces not equal

Generative Modelling for Molecule Generation

- ✗ Node independence assumed
- ✗ Arbitrary node ordering
- ✗ Different generation traces not equal



Evaluate molecules consistently, regardless of generation trace

Fully order/ Permutation equivariant

Model interactions between nodes iteratively

★

Structure-based Drug Design with Equivariant Diffusion Models

Arne S. (EPFL), Yuanqi Du (Cornell), Charles Harris (Cambridge), Arian J. (Cambridge), Ilia Igashov (EPFL), Weitao Du (Cornell), Tom Blundell (Cambridge), Pietro Lio (Cambridge), Carla Gomes (Cornell), Max Welling (Amsterdam/Microsoft), Michael Bronstein (Oxford/Twitter), Bruno Correia (EPFL)

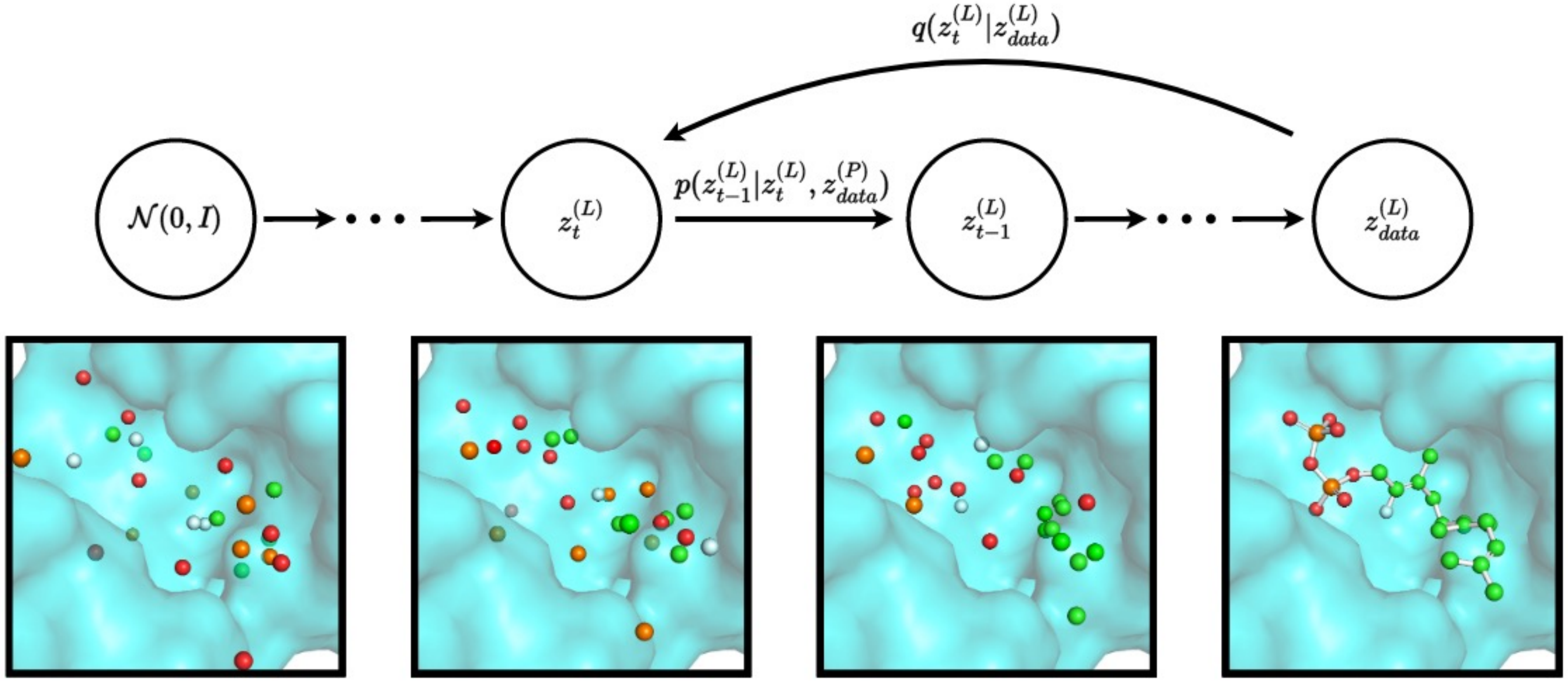
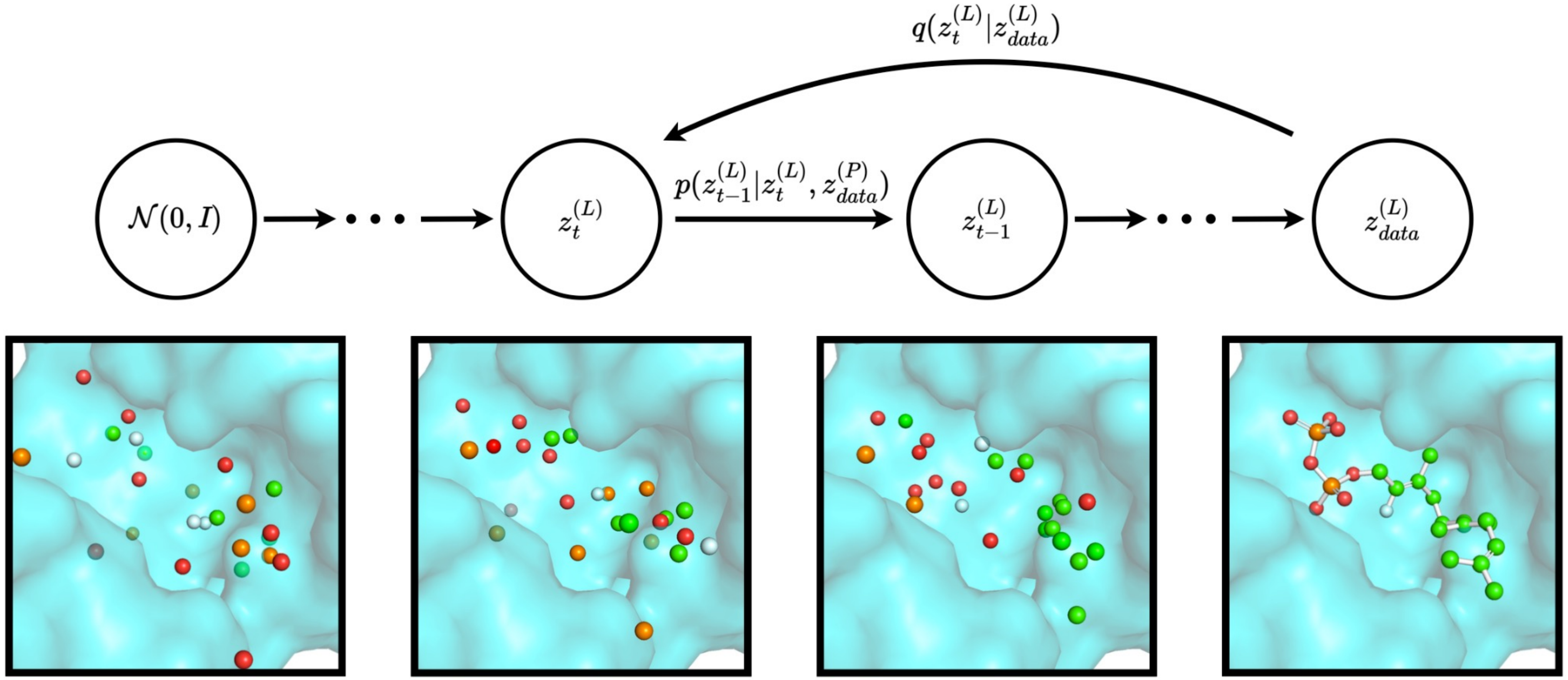

$q(z_t^{(L)} | z_{data}^{(L)})$
$\mathcal{N}(0, I)$
$z_t^{(L)}$
$p(z_{t-1}^{(L)} | z_t^{(L)}, z_{data}^{(P)})$
$z_{t-1}^{(L)}$
$z_{data}^{(L)}$

Image source – Schneuing et al, 2022

DiffSBDD:

A Diffusion Model for Structure-based Drug Design

- **Both proteins and ligands are represented as all-atom graphs**
- **Learns the transitional probability distribution $\boldsymbol{p_\theta\left(z_{t-1}^{(L)} \middle| z_t^{(L)}, z_{data}^{(P)}\right)}$**
- **Denoising network $\hat{\epsilon}_\theta$ constructs samples**



$q(z_t^{(L)}|z_{data}^{(L)})$
$\mathcal{N}(0, I)$
$z_t^{(L)}$
$p(z_{t-1}^{(L)}|z_t^{(L)}, z_{data}^{(P)})$
$z_{t-1}^{(L)}$
$z_{data}^{(L)}$

$$\boldsymbol{z}_{\text{data}} = [\boldsymbol{x}, \boldsymbol{h}]$$

$$\hat{\boldsymbol{\epsilon}}_\theta = \phi_\theta(\boldsymbol{z}_t^{(L)}, \boldsymbol{z}_{\text{data}}^{(P)}, t)$$

SE(3)-equivariant Graph Neural Network

Based on: Schneuing, Arne, et al. "Structure-based drug design with equivariant diffusion models." *NeurIPS MLSB* 2022.

DiffSBDD: Results

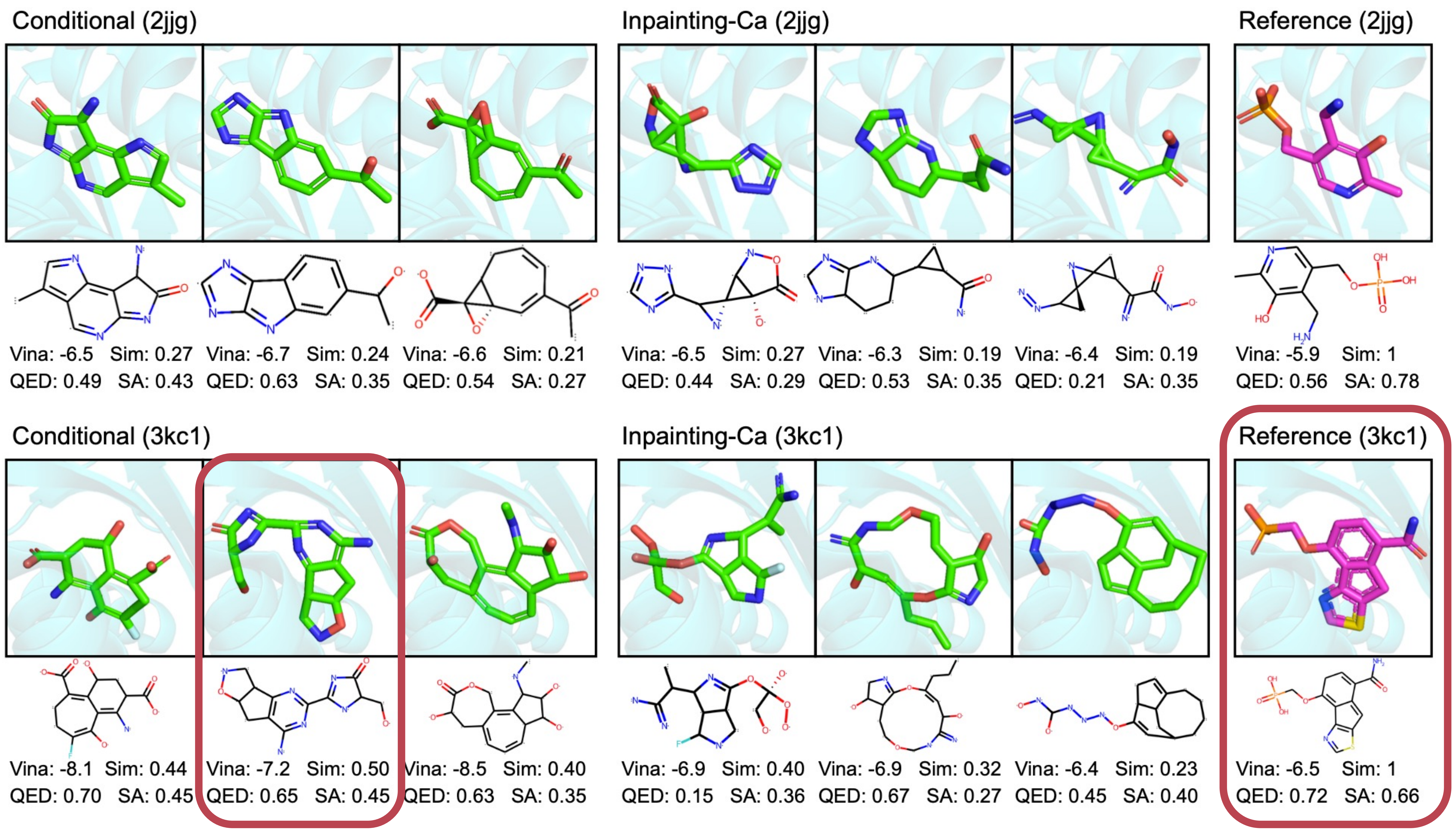
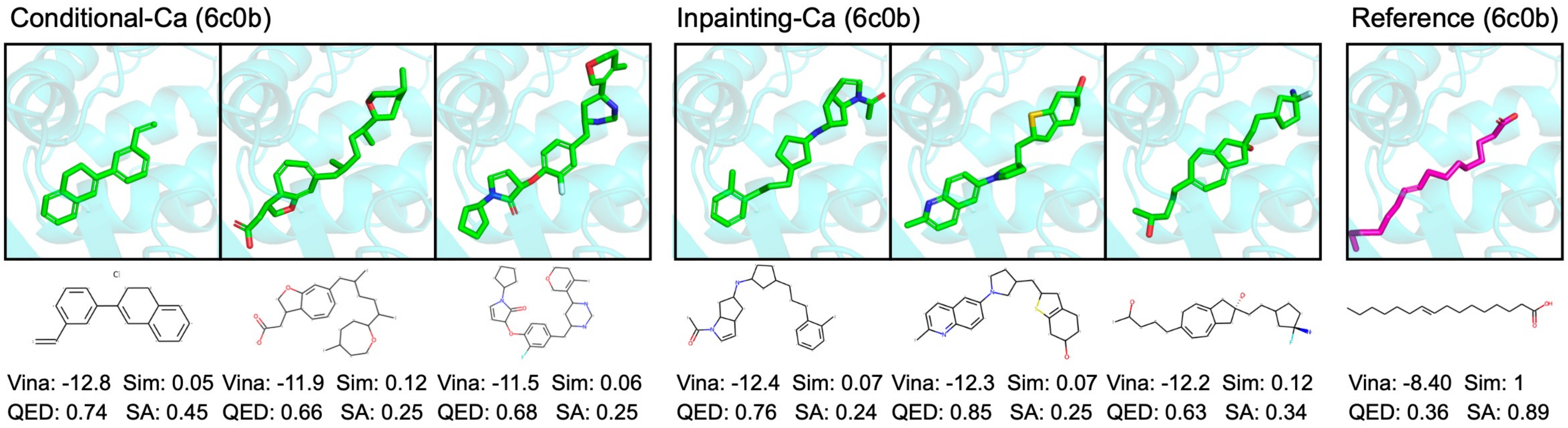


Image source – Schneuing et al, 2022

DiffSBDD: Results



Conditional-Ca (6c0b)

Inpainting-Ca (6c0b)

Reference (6c0b)

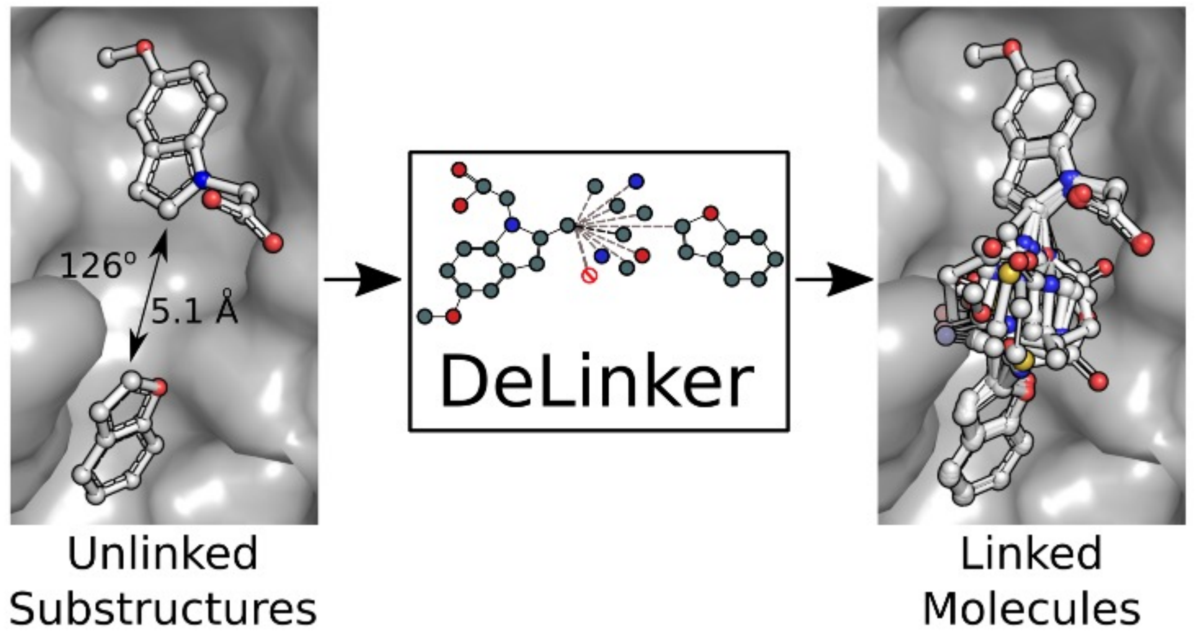
Image source – Schneuing et al, 2022

Other Strategies in SBDD

Not having to design molecule de novo simplifies the process

Fragment-based Drug Design

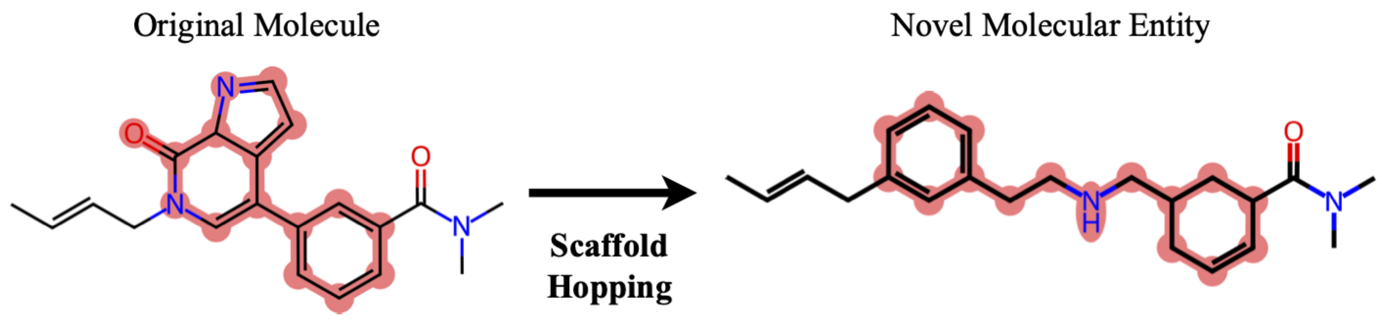
Identifying small molecular motifs (**fragments**) that bind inside a pocket and then **linking** these for form a whole lead molecule



Example: **DiffLinker**

Identifying small molecular motifs (**fragments**) that bind inside a pocket and then **linking** these for form a whole lead molecule

Scaffold Hopping



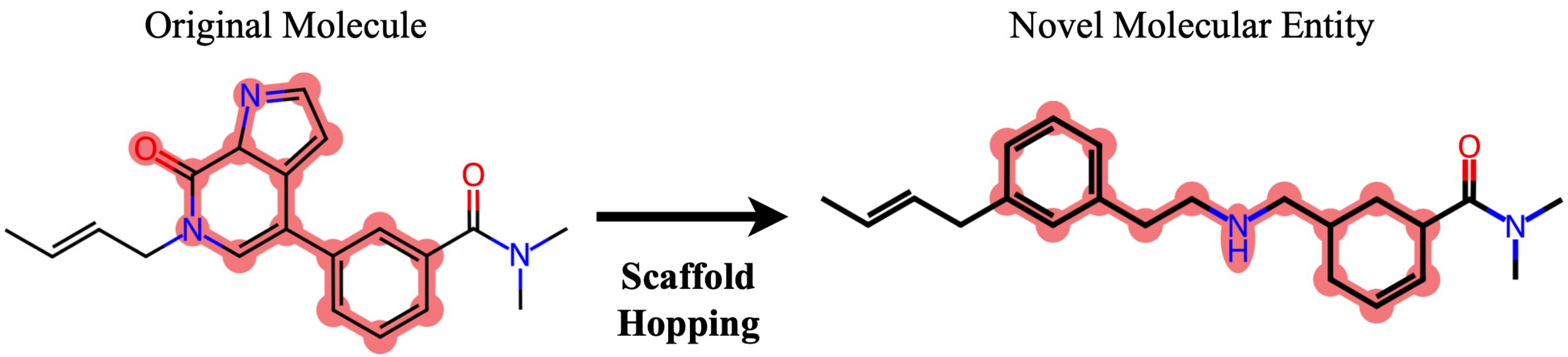
Example: **DiffHopp**

Identifying small molecular motifs (**fragments**) that bind inside a pocket and then **linking** these for form a whole lead molecule

Image source – State of AI Report

What is scaffold hopping?

Similar properties, novel topology

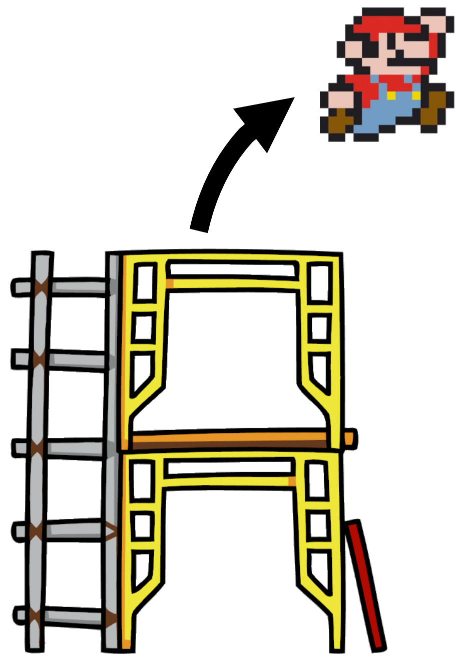


Original Molecule

Novel Molecular Entity

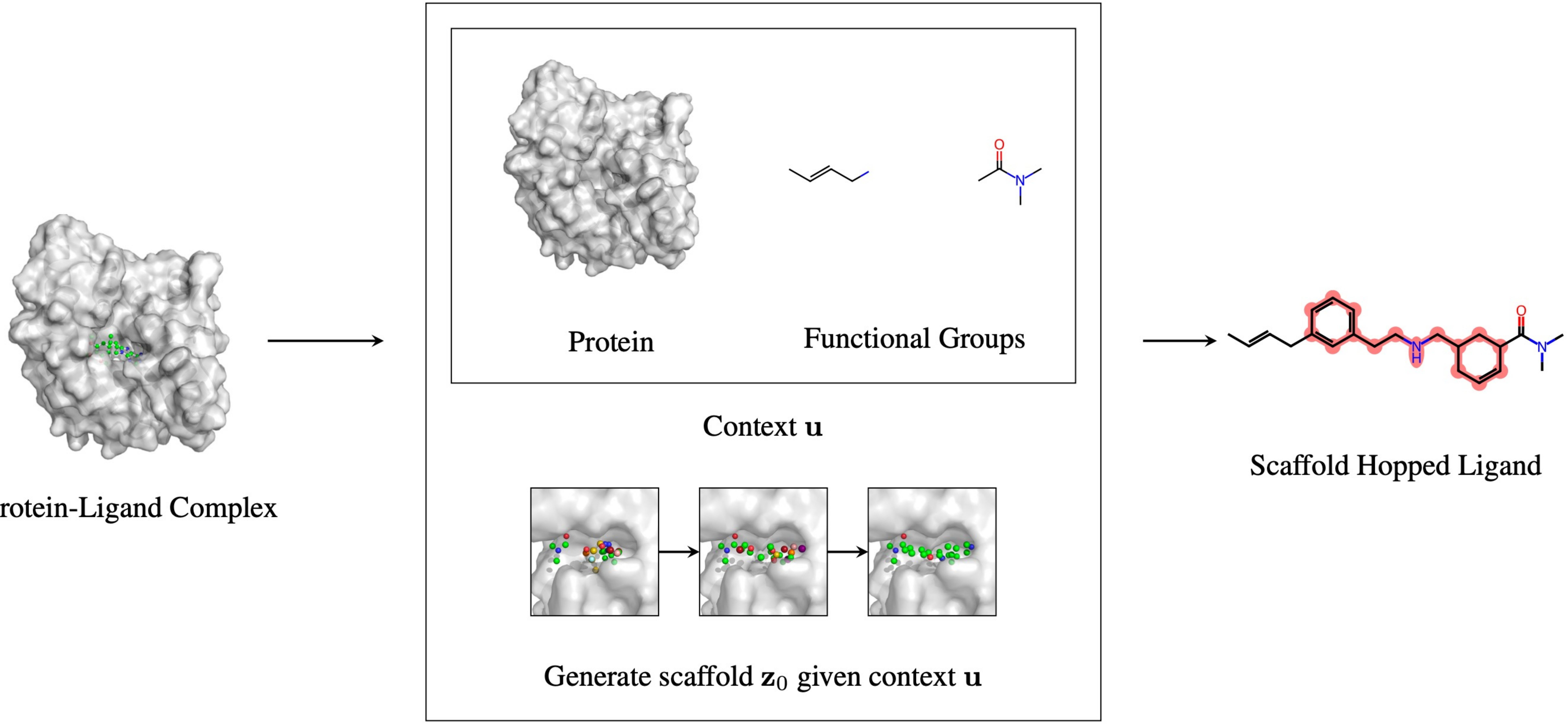
Scaffold Hopping

👋 **“molecular core”** 👋



DiffHopp

A conditional diffusion model for scaffold hopping



Protein-Ligand Complex

Protein

Functional Groups

Context $\mathbf{u}$

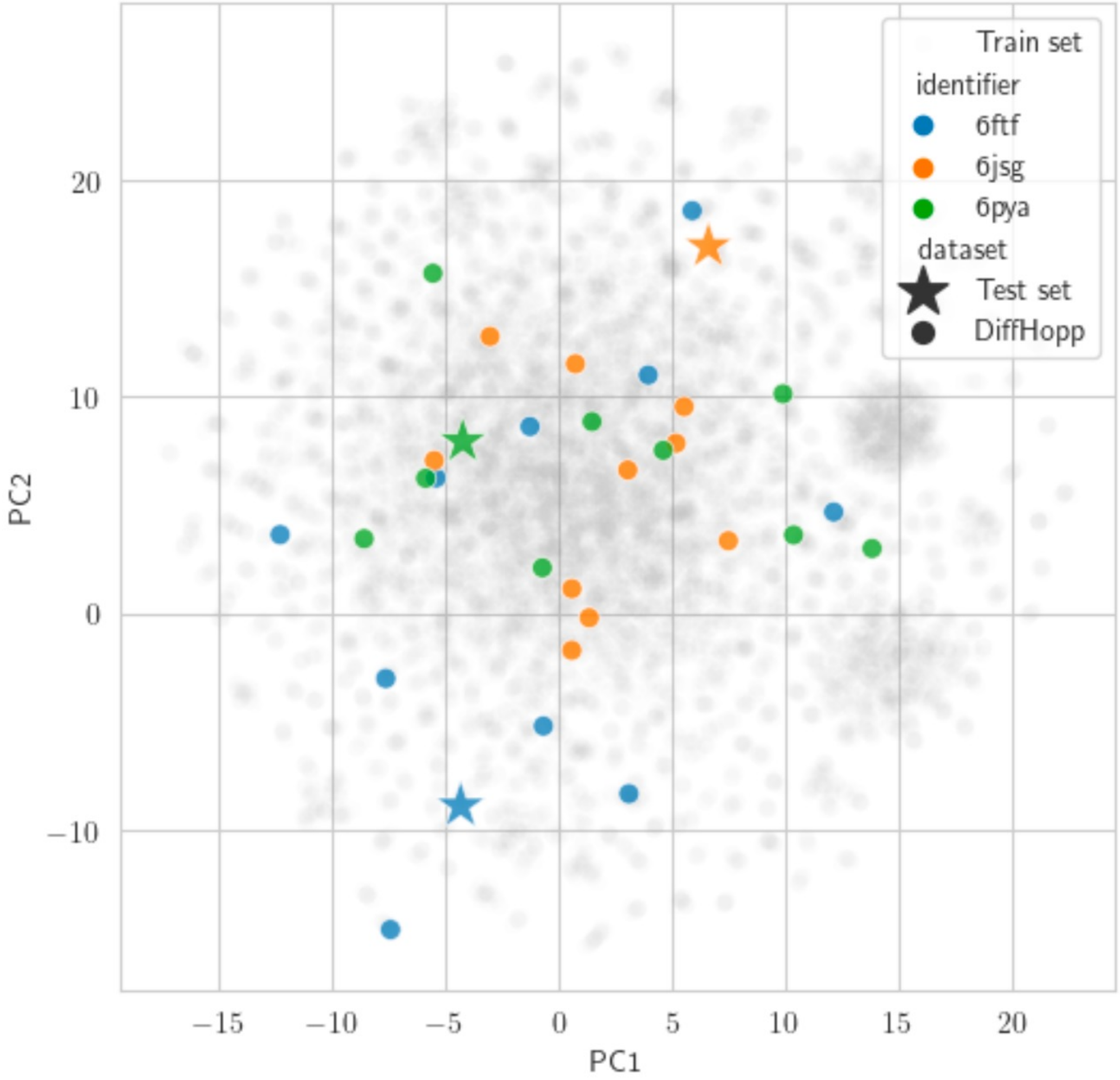
Generate scaffold $\mathbf{z}_0$ given context $\mathbf{u}$

Scaffold Hopped Ligand

Scaffold diversity analysis

Generated scaffold are as diverse as the PDB

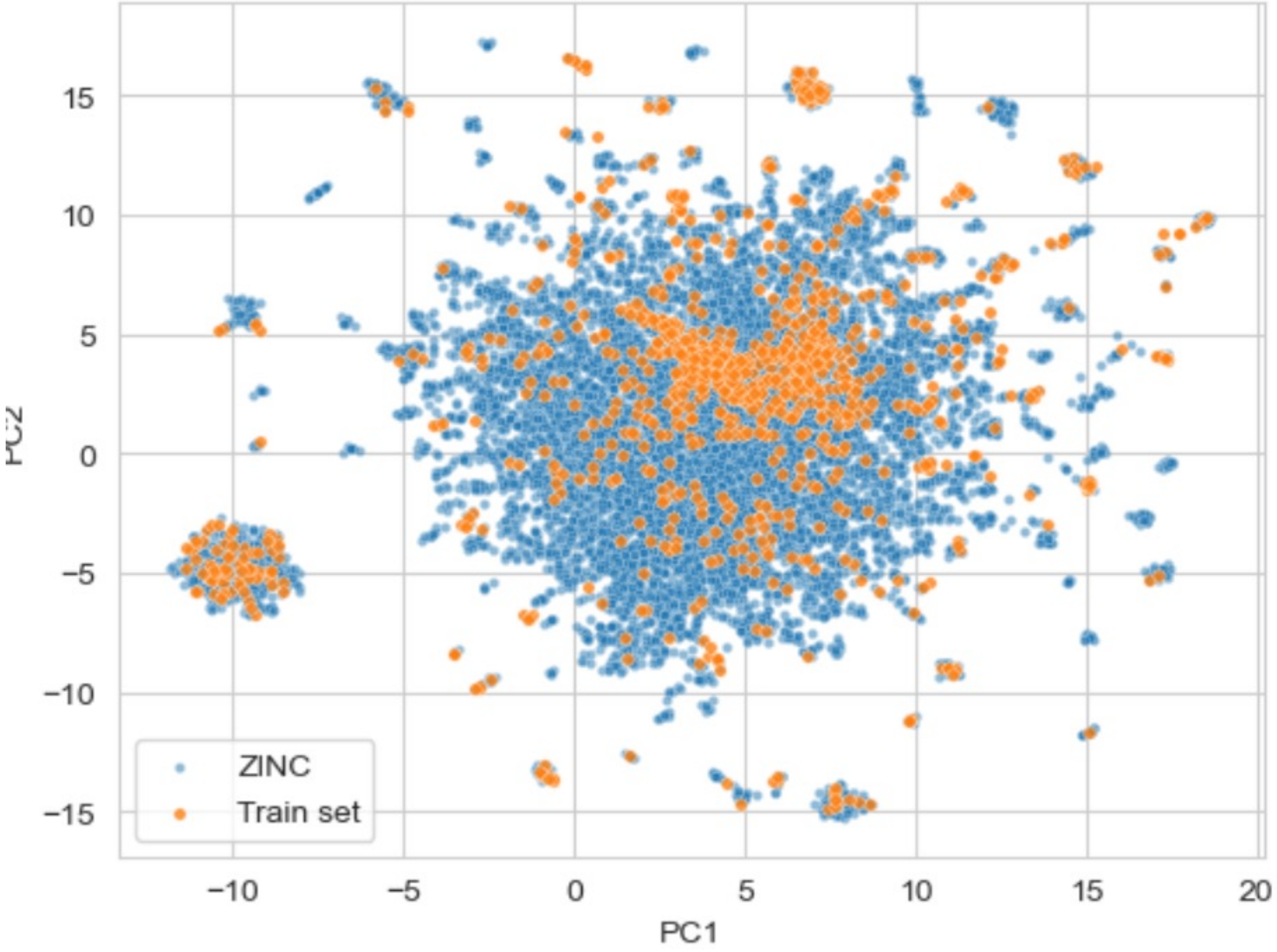
- **Can generate diverse scaffolds, regardless of the starting chemotype**
- **However, scaffold space is very large and we are limited by the PDB (<40,000)**



Scaffold diversity analysis

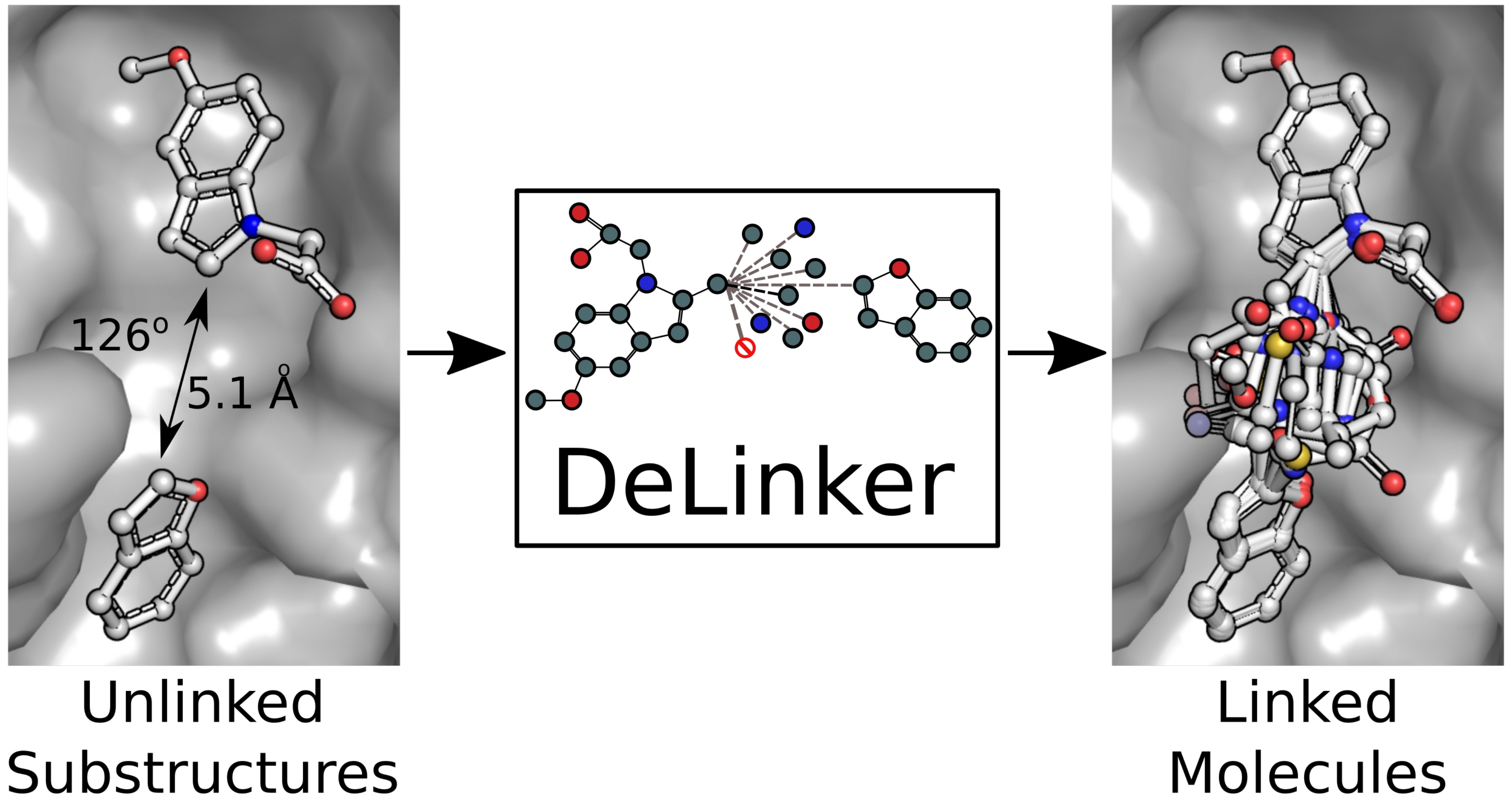
Generated scaffold are as diverse as the PDB

- **Can generate diverse scaffolds, regardless of the starting chemotype**
- **However, scaffold space is very large and we are limited by the PDB (<40,000)**



There are many sub-tasks within SBDD

e.g. Fragment-linking, scaffold hopping



(Imrie et al, 2020)

Conditional models are trained on synthetic dataset

Specialist models cannot generalise to new tasks

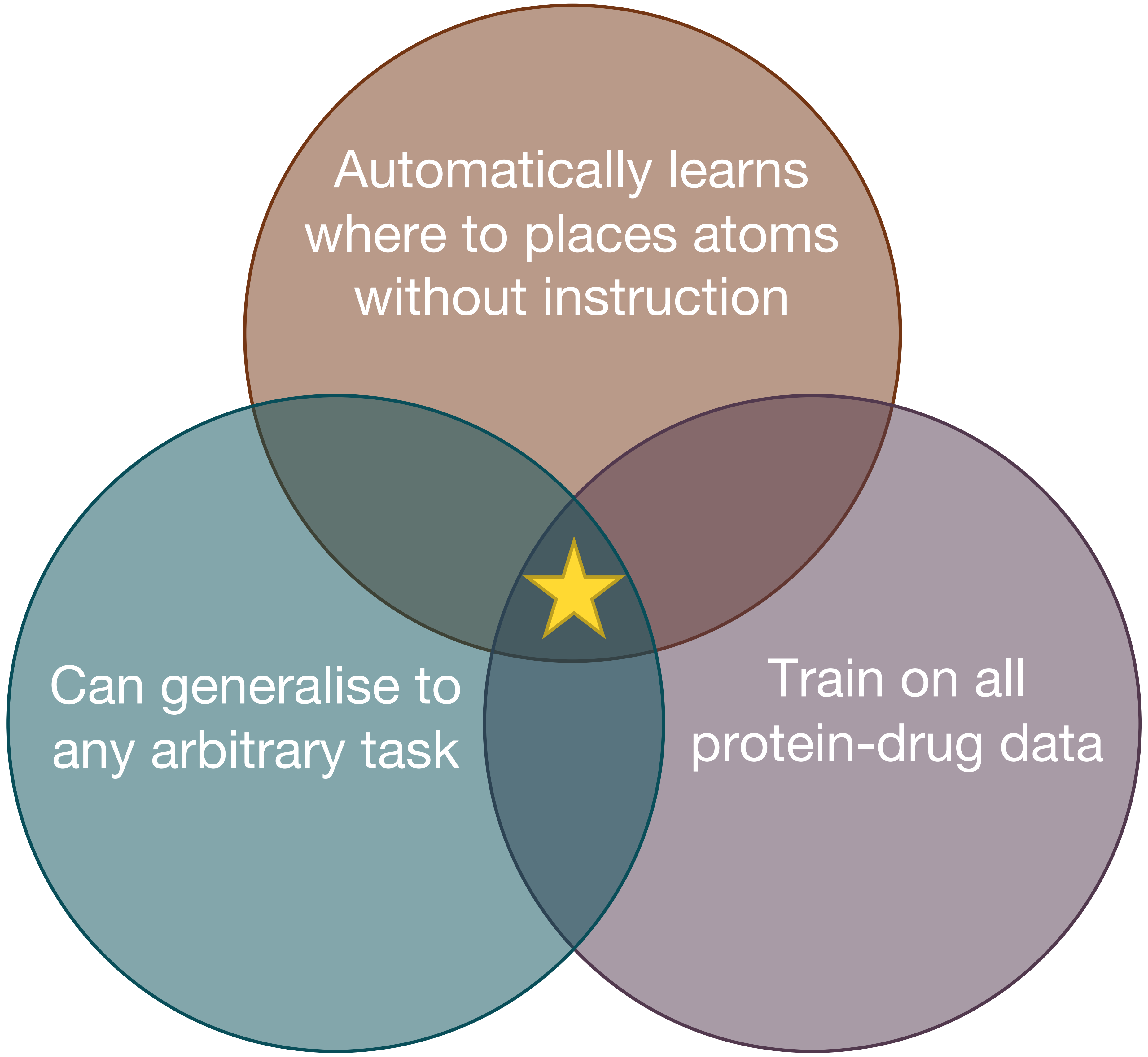
Need to prespecify atom attachment points

Limitation of conditional models

✗ Conditional models are trained on synthetic dataset

✗ Specialist models cannot generalise to new tasks

✗ Need to prespecify atom attachment points

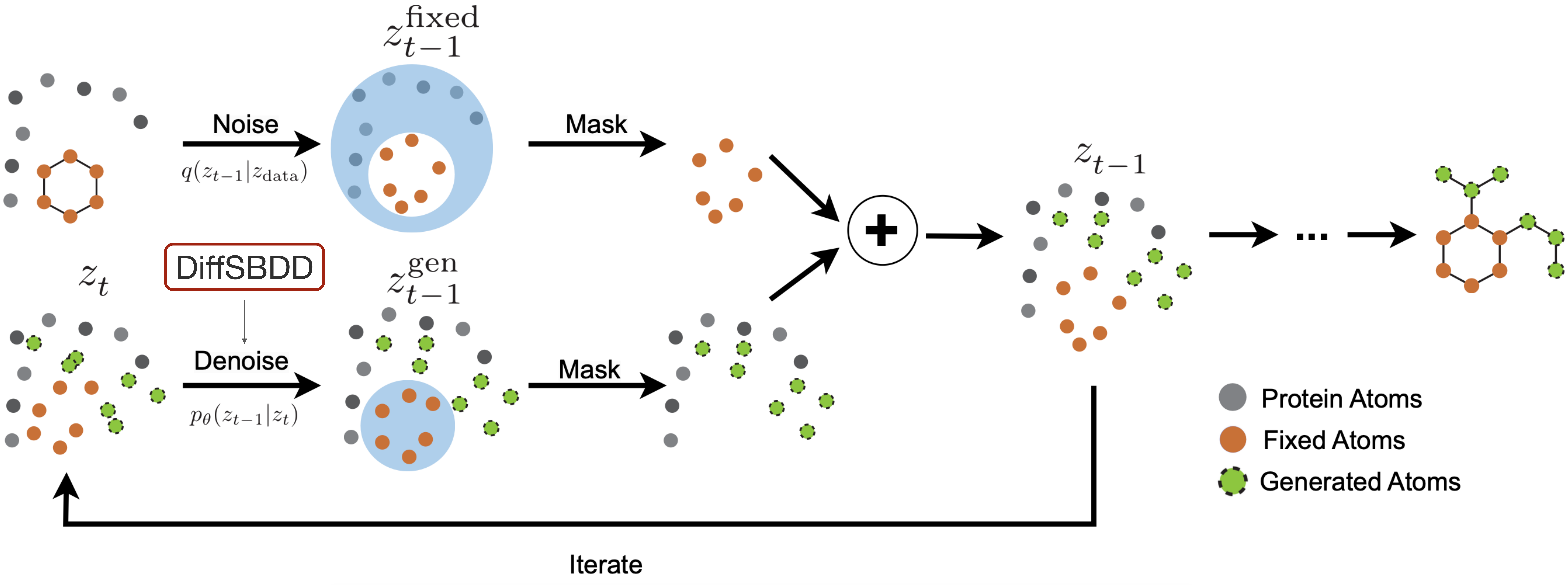


Automatically learns where to places atoms without instruction

Can generalise to any arbitrary task

Train on all protein-drug data

Molecular inpainting with DiffSBDD



Based on: Harris, et al. "*Flexible Small-Molecule Design and Optimization with Equivariant Diffusion Models*." *ICML MLDD* 2023.

🥡 Takeaway 🥡

Generative modelling **holds promise** for designing novel drugs but has **no real-world validation**