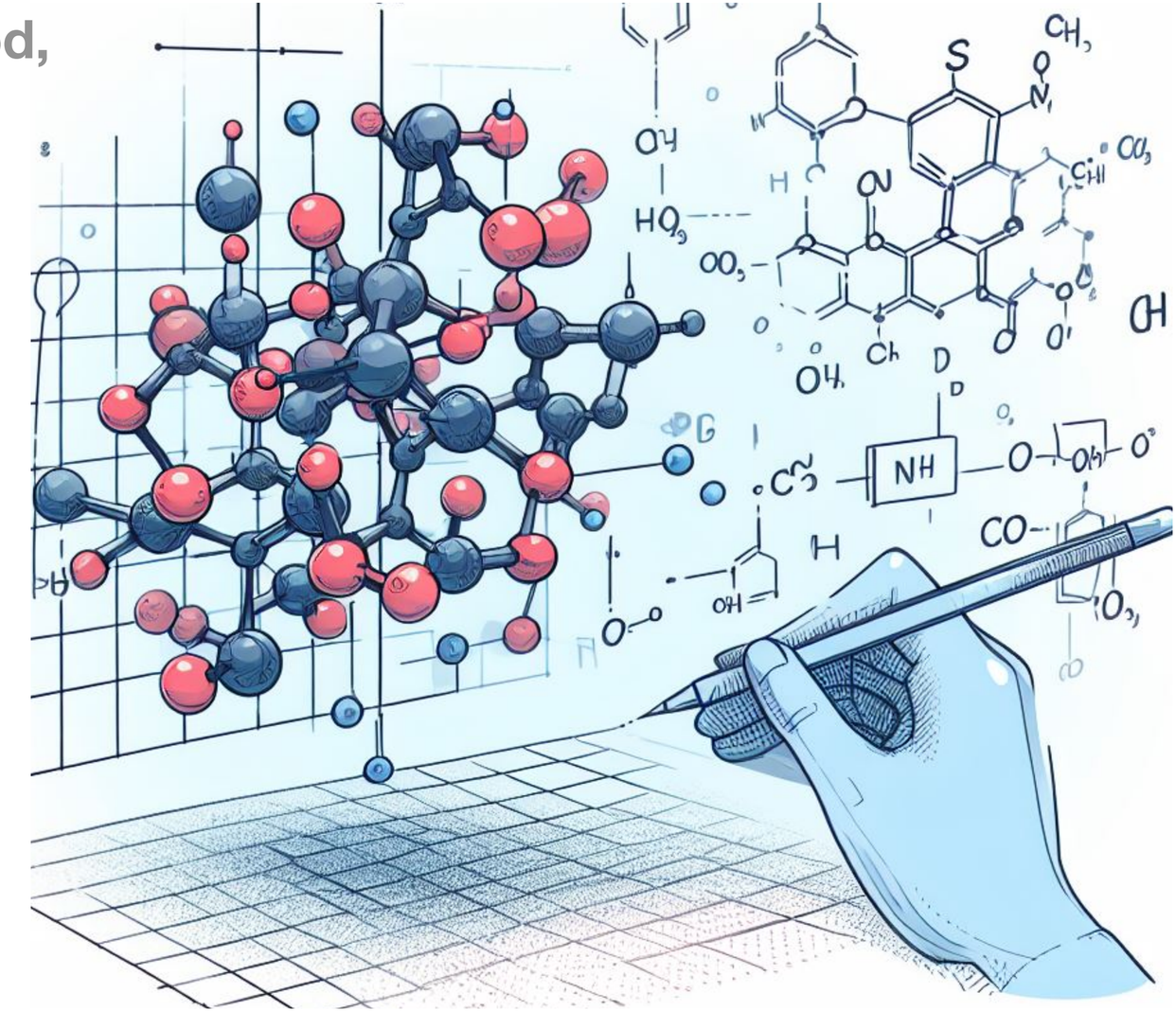
Structure-based drug design by denoising voxel grids

Pedro O. Pinheiro, Arian Jamasb, Omar Mahmood,

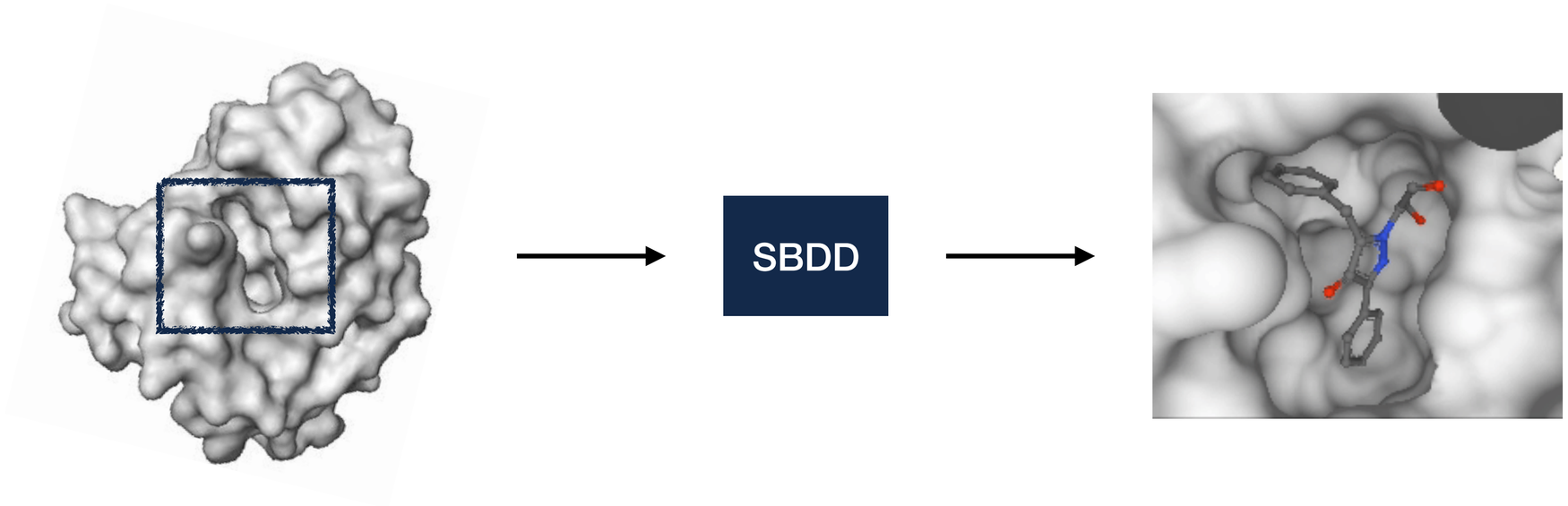
Vishnu Sresht, Saeed Saremi



Genentech

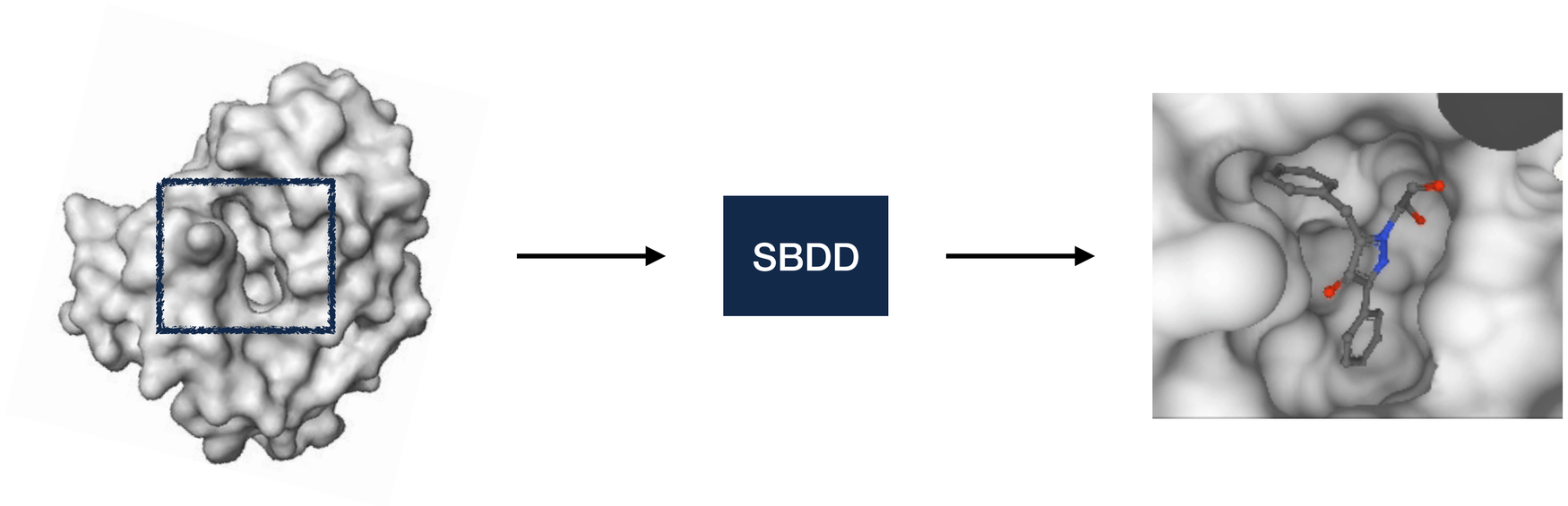
Structure-based drug design (SBDD)

Design molecules that bind to a specific 3D protein structures with high affinity and specifity (Anderson, 2003).



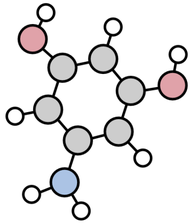
Structure-based drug design (SBDD)

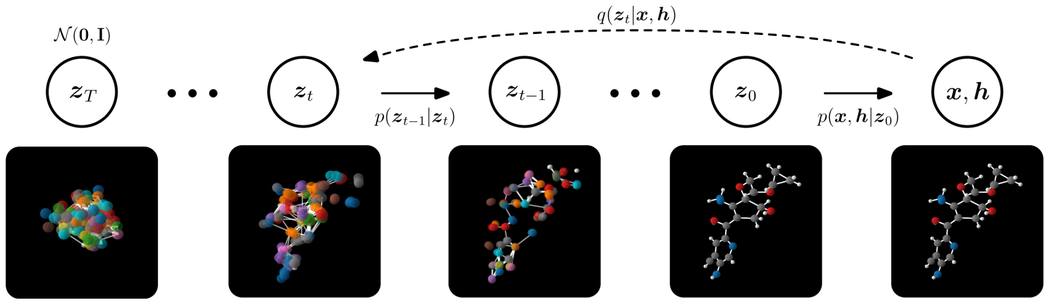
Design molecules that bind to a specific 3D protein structures with high affinity and specifity (Anderson, 2003).



3D molecule generation

Point-clouds

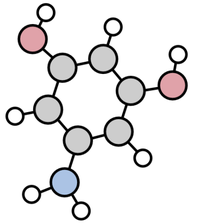


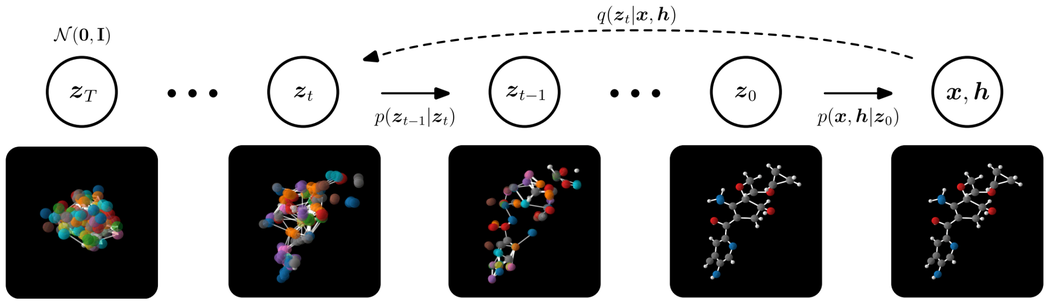


- Molecules as point-clouds
- SE(3)-equivariant GNN
- Generation with diffusion models

3D molecule generation

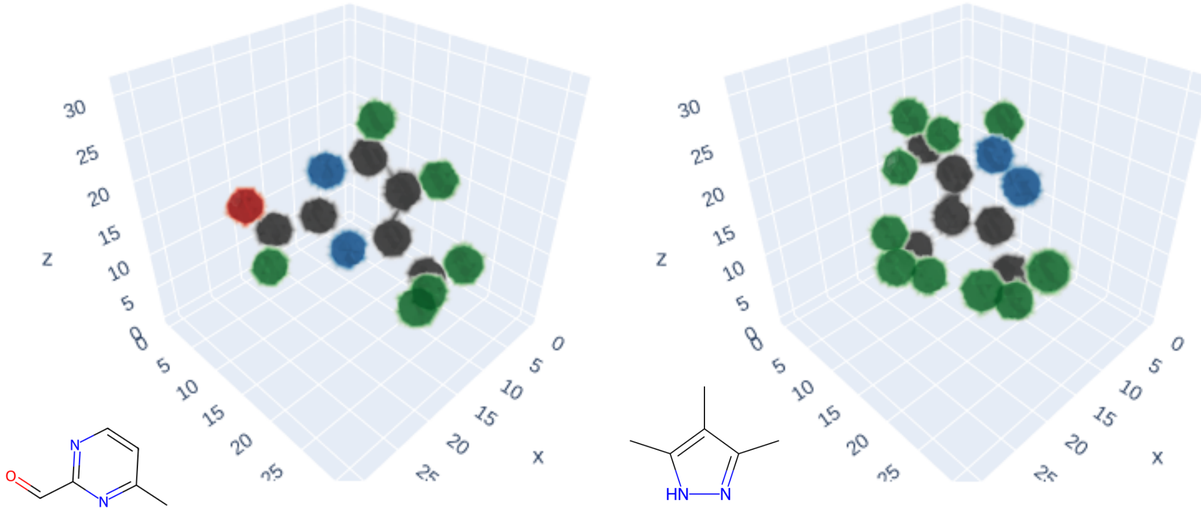
Point-clouds





- Molecules as point-clouds
- SE(3)-equivariant GNN
- Generation with diffusion models

Voxel grids



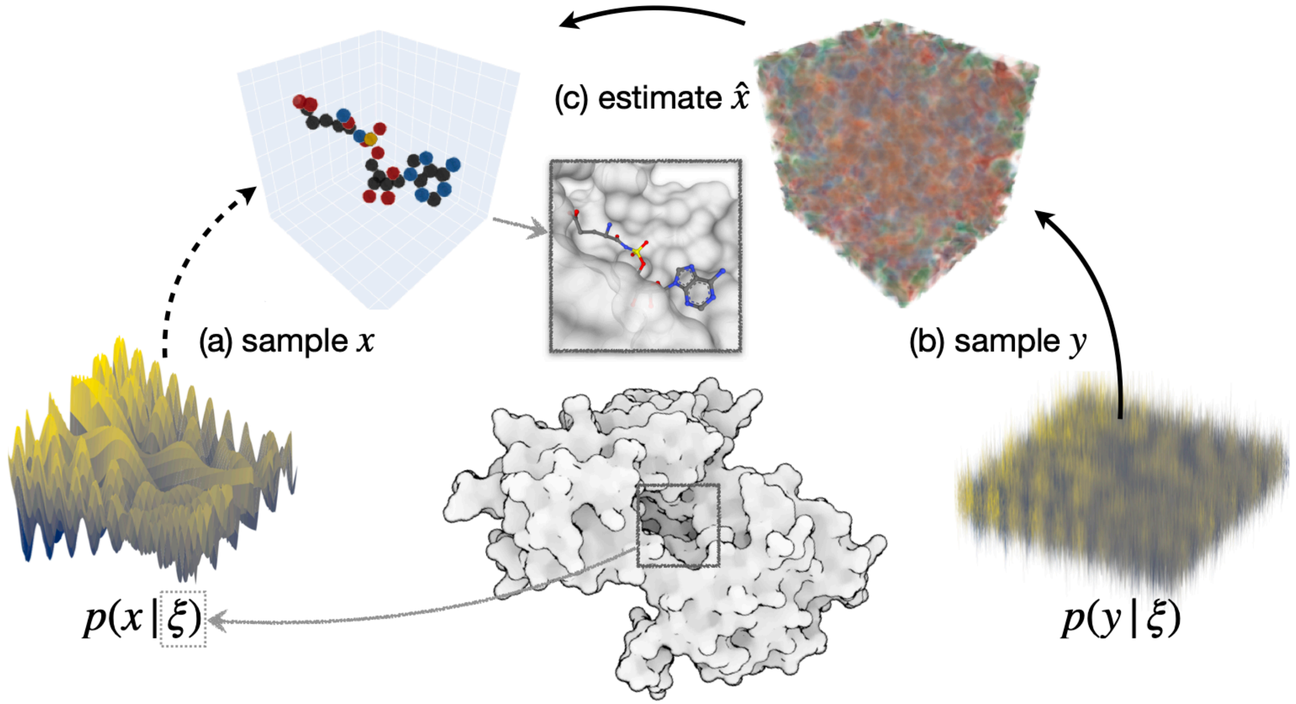
- Molecules as discrete atom densities
- Leverage success of **denoising architectures** in computer vision
- Generation through ***neural empirical Bayes***

Conditional neural Empirical Bayes

Goal: sample from $p(x \mid \xi)$, the distribution of ligands x given pocket ξ.

- **Intuition:**
  - Hard to sample $X \sim p(x \mid \xi)$ **(a)**
  - Easier to sample from

$$Y \sim p(y \mid \xi) = p(x \mid \xi) * \mathcal{N}(0, \sigma^2 I_d) \textbf{ (b)}$$



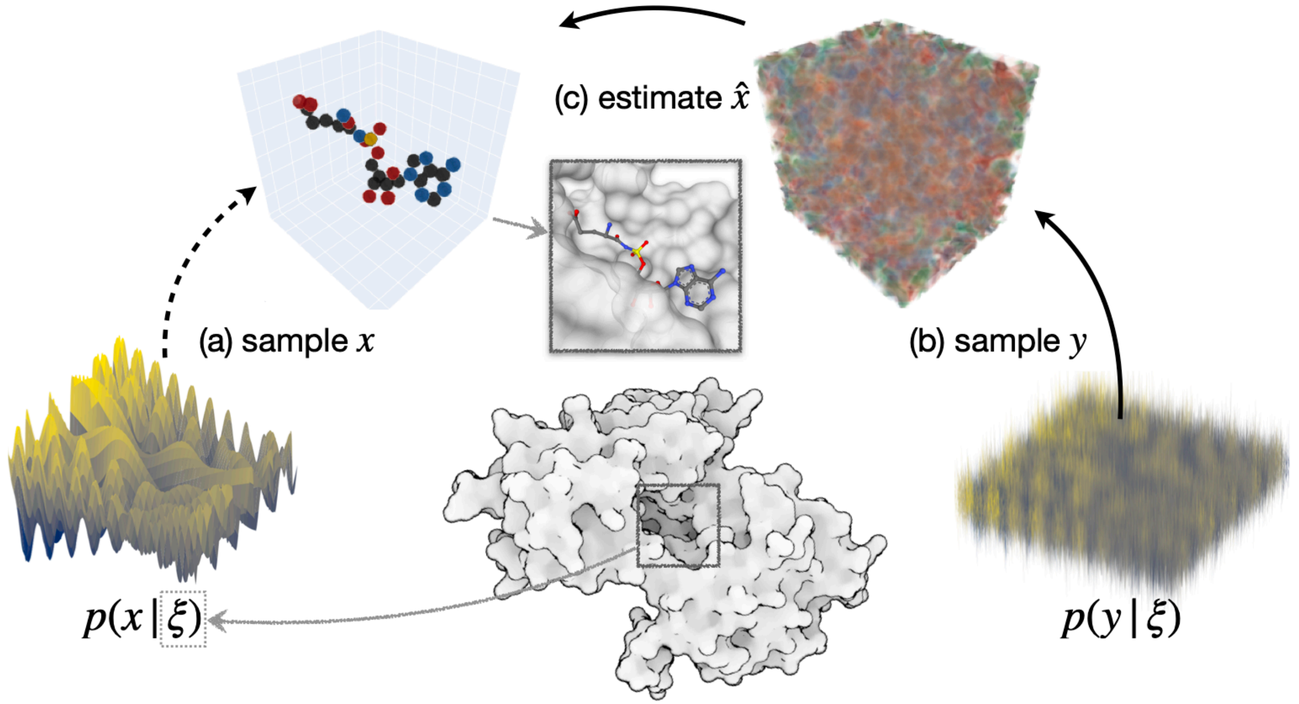
Conditional neural Empirical Bayes

Goal: sample from $p(x \mid \xi)$, the distribution of ligands x given pocket ξ.

- **Intuition:**
  - Hard to sample $X \sim p(x \mid \xi)$ **(a)**
  - Easier to sample from

$$Y \sim p(y \mid \xi) = p(x \mid \xi) * \mathcal{N}(0, \sigma^2 I_d)$$ **(b)**

- **Idea:**
  - Sample $Y \sim p(y \mid \xi)$ **(b)**
  - Estimate x with $\mathbb{E}(X \mid y, \xi)$ **(c)**

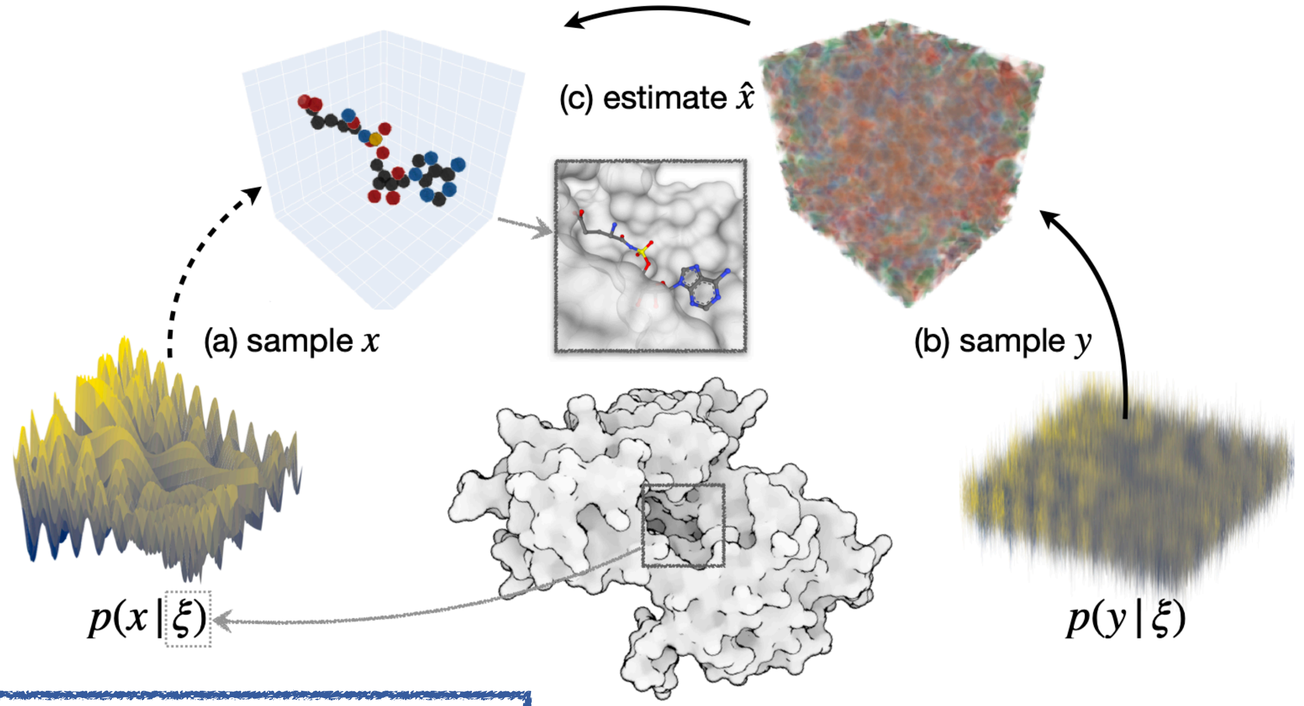


Conditional neural Empirical Bayes

Goal: sample from $p(x \mid \xi)$, the distribution of ligands x given pocket ξ.

- **Intuition:**
  - Hard to sample $X \sim p(x \mid \xi)$ **(a)**
  - Easier to sample from

$$Y \sim p(y \mid \xi) = p(x \mid \xi) * \mathcal{N}(0, \sigma^2 I_d) \text{ (b)}$$



- **Idea:**
  - Sample $Y \sim p(y \mid \xi)$ **(b)**
  - Estimate x with $\mathbb{E}(X \mid y, \xi)$ **(c)**

$$\hat{x}(y \mid \xi) = y + \sigma^2 \nabla_y \log p(y \mid \xi)$$

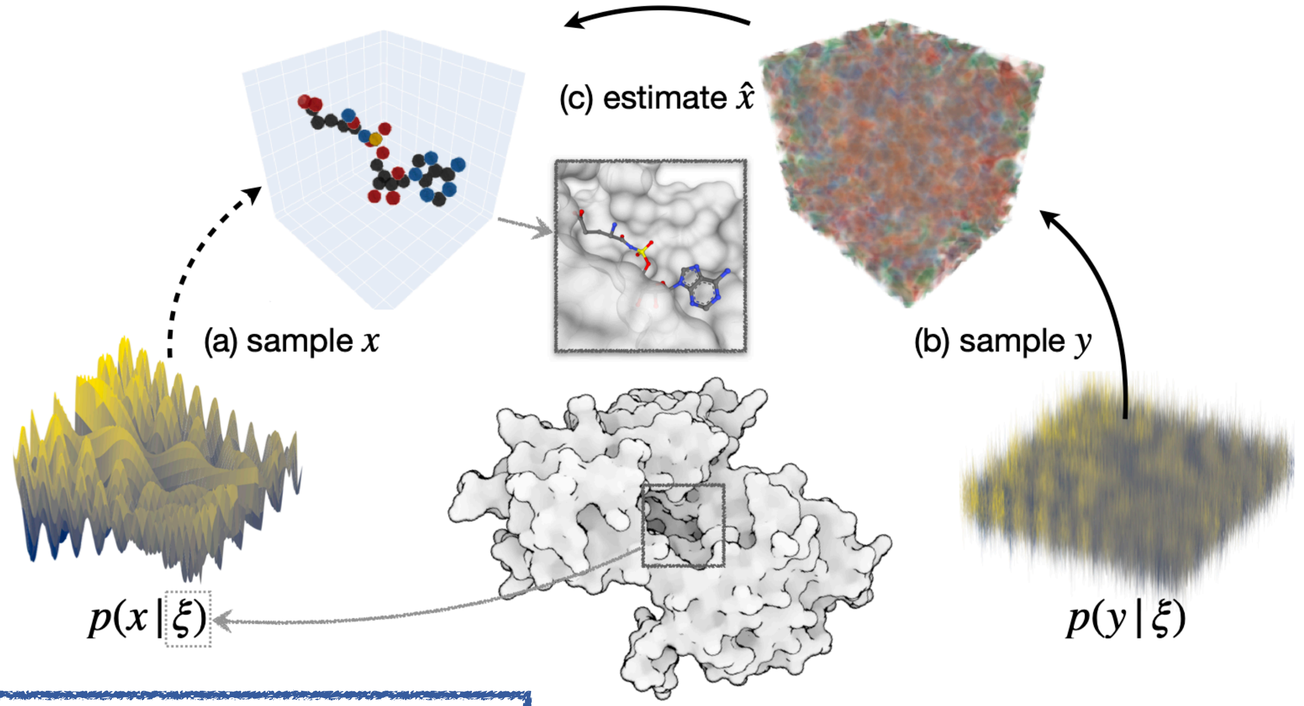
conditional version of (Miyasawa, 61)

Conditional neural Empirical Bayes

Goal: sample from $p(x \mid \xi)$, the distribution of ligands x given pocket ξ.

- **Intuition:**
  - Hard to sample $X \sim p(x \mid \xi)$ **(a)**
  - Easier to sample from

$$Y \sim p(y \mid \xi) = p(x \mid \xi) * \mathcal{N}(0, \sigma^2 I_d) \text{ (b)}$$



- **Idea:**
  - Sample $Y \sim p(y \mid \xi)$ **(b)**
  - Estimate x with $\mathbb{E}(X \mid y, \xi)$ **(c)**

$$\hat{x}(y \mid \xi) = y + \sigma^2 \nabla_y \log p(y \mid \xi)$$

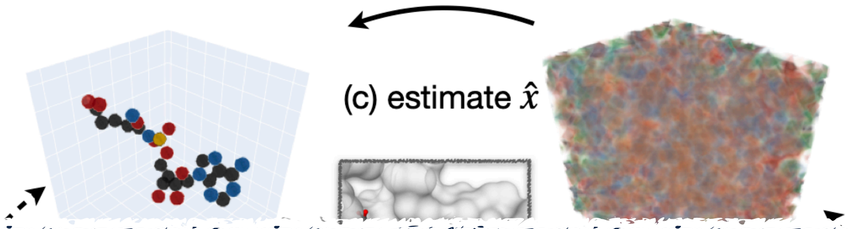
conditional version of (Miyasawa, 61)

$$\mathbb{E}(X \mid y, \xi) = \hat{x}(y \mid \xi)$$ (Robbins, 56)

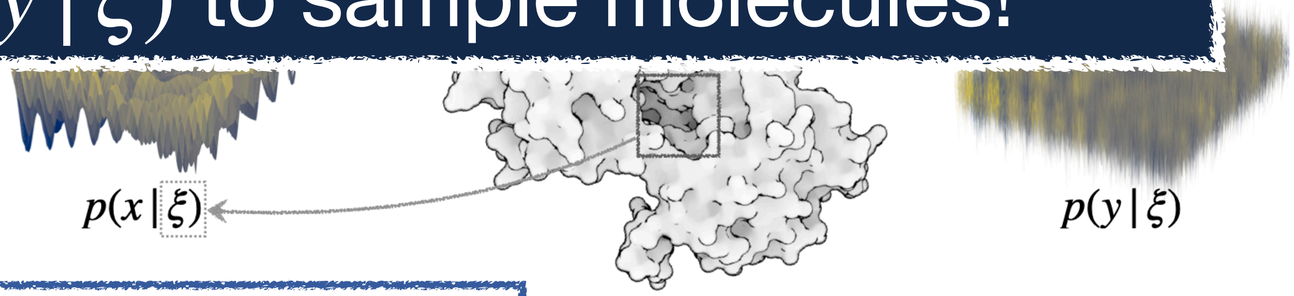
Conditional neural Empirical Bayes

Goal: sample from $p(x \mid \xi)$, the distribution of ligands x given pocket ξ.

- **Intuition:**
  - Hard to sample $X \sim p(x \mid \xi)$ **(a)**



Only need the conditional least-square estimator (ie conditional denoiser) $\hat{x}(y \mid \xi)$ to sample molecules!

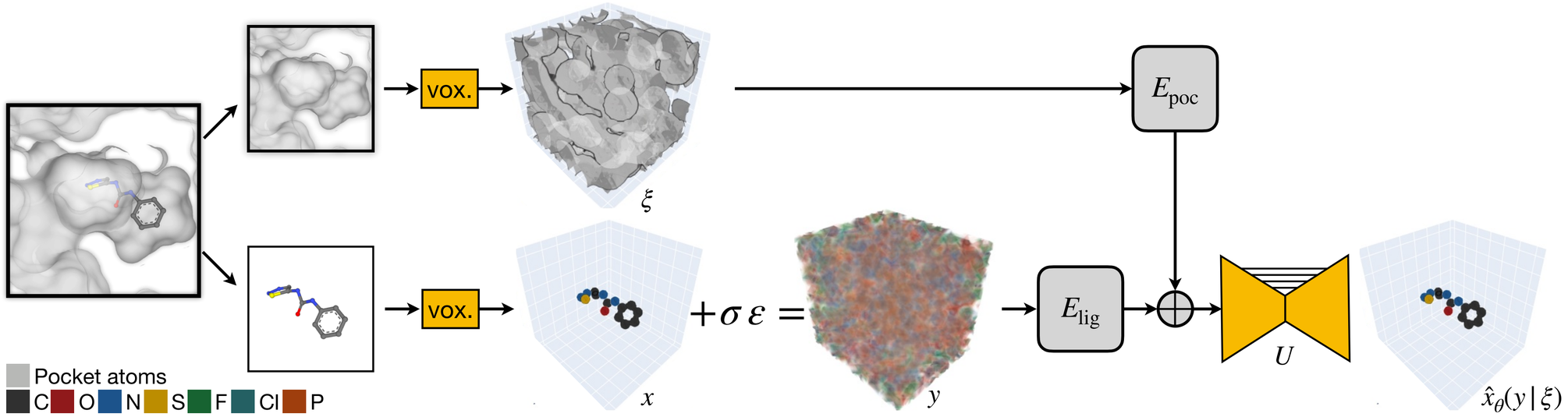


- **Idea:**
  - Sample $Y \sim p(y \mid \xi)$ **(b)**
  - Estimate x with $\mathbb{E}(X \mid y, \xi)$ **(c)**

$$\hat{x}(y \mid \xi) = y + \sigma^2 \nabla_y \log p(y \mid \xi)$$
conditional version of (Miyasawa, 61)

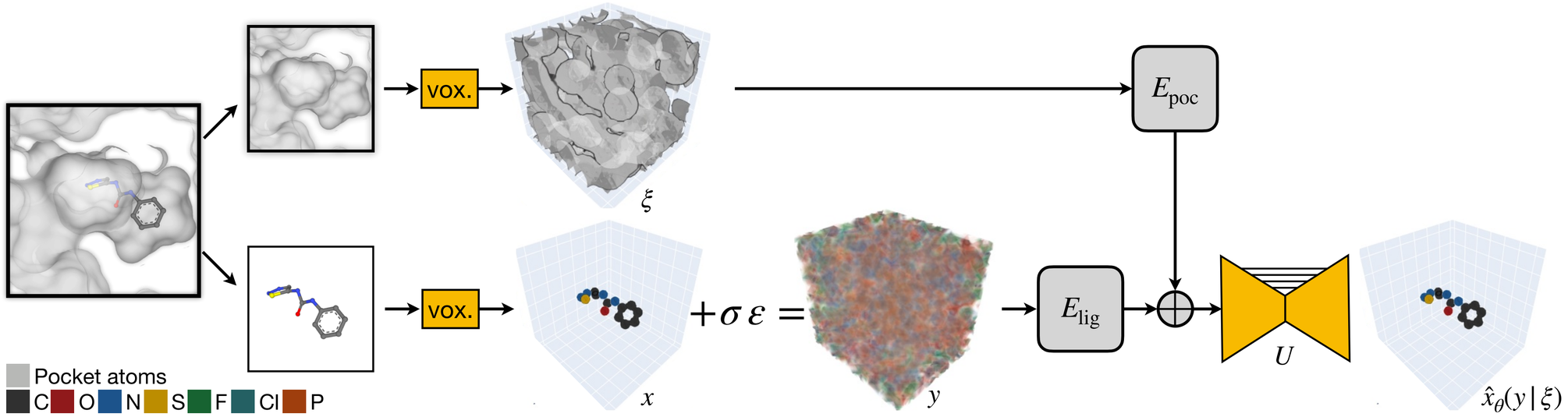
$$\mathbb{E}(X \mid y, \xi) = \hat{x}(y \mid \xi)$$ (Robbins, 56)

Training: Conditional voxel denoiser



- **Voxelized molecules**:
  - Atoms are Gaussian blobs in 3D, centered at atomic centers
  - Each atom type at different grid channels (color)
  - **Ligand:** $x \in \mathbb{R}^{d_x}$, $d_x = 7 \times 64 \times 64 \times 64$ (res. .25Å)
  - **Pocket:** $\xi \in \mathbb{R}^{d_\xi}$, $d_\xi = 4 \times 64 \times 64 \times 64$ (res. .25Å)
- Approximate **conditional least-square estimator** with neural net. $\hat{x}_\theta : \mathbb{R}^{d_x} \times \mathbb{R}^{d_\xi} \to \mathbb{R}^{d_x}$

Training: Conditional voxel denoiser



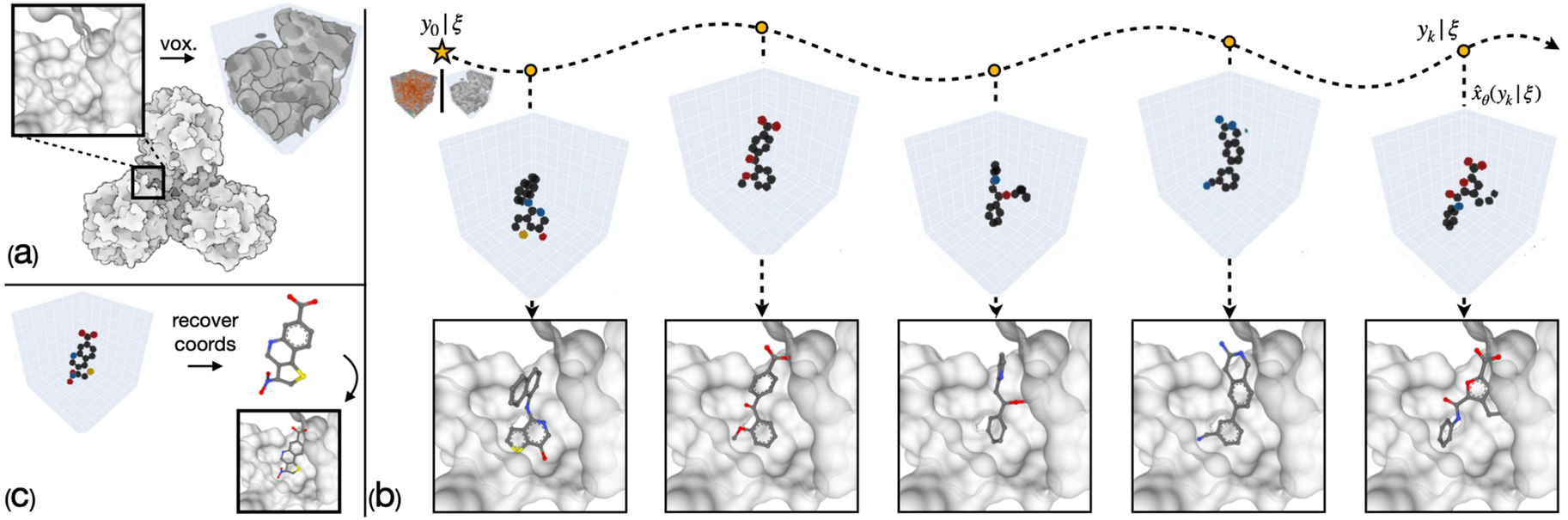
- **Voxelized molecules**:
  - Atoms are Gaussian blobs in 3D, centered at atomic centers
  - Each atom type at different grid channels (color)
  - **Ligand:** $x \in \mathbb{R}^{d_x}$, $d_x = 7 \times 64 \times 64 \times 64$ (res. .25Å)
  - **Pocket:** $\xi \in \mathbb{R}^{d_\xi}$, $d_\xi = 4 \times 64 \times 64 \times 64$ (res. .25Å)
- Approximate **conditional least-square estimator** with neural net. $\hat{x}_\theta : \mathbb{R}^{d_x} \times \mathbb{R}^{d_\xi} \to \mathbb{R}^{d_x}$

- **Denoising objective:**

$$\min_\theta \ \mathbb{E}_{(x,\xi)\sim p(x,\xi),\varepsilon\sim\mathcal{N}(0,\sigma^2 I_{d_x})} \|\hat{x}_\theta(x+\sigma\varepsilon \mid \xi) - x\|^2$$

- The **conditional denoiser** gives us:
  - $g_\theta(y \mid \xi) = \frac{1}{\sigma^2}(\hat{x}_\theta(y \mid \xi) - y) \approx \nabla_y \log p(y \mid \xi)$
  - $\hat{x}_\theta(y \mid \xi) \approx \mathbb{E}(X \mid y, \xi)$

Sampling: conditional walk-jump sampling (cWJS)



1. **(initialize)** Voxelize the pocket ξ and initialize y_0 with noise

2. ***(walk)*** sample noisy ligands $y_k \sim p(y \mid \xi)$ with Langevin MCMC*

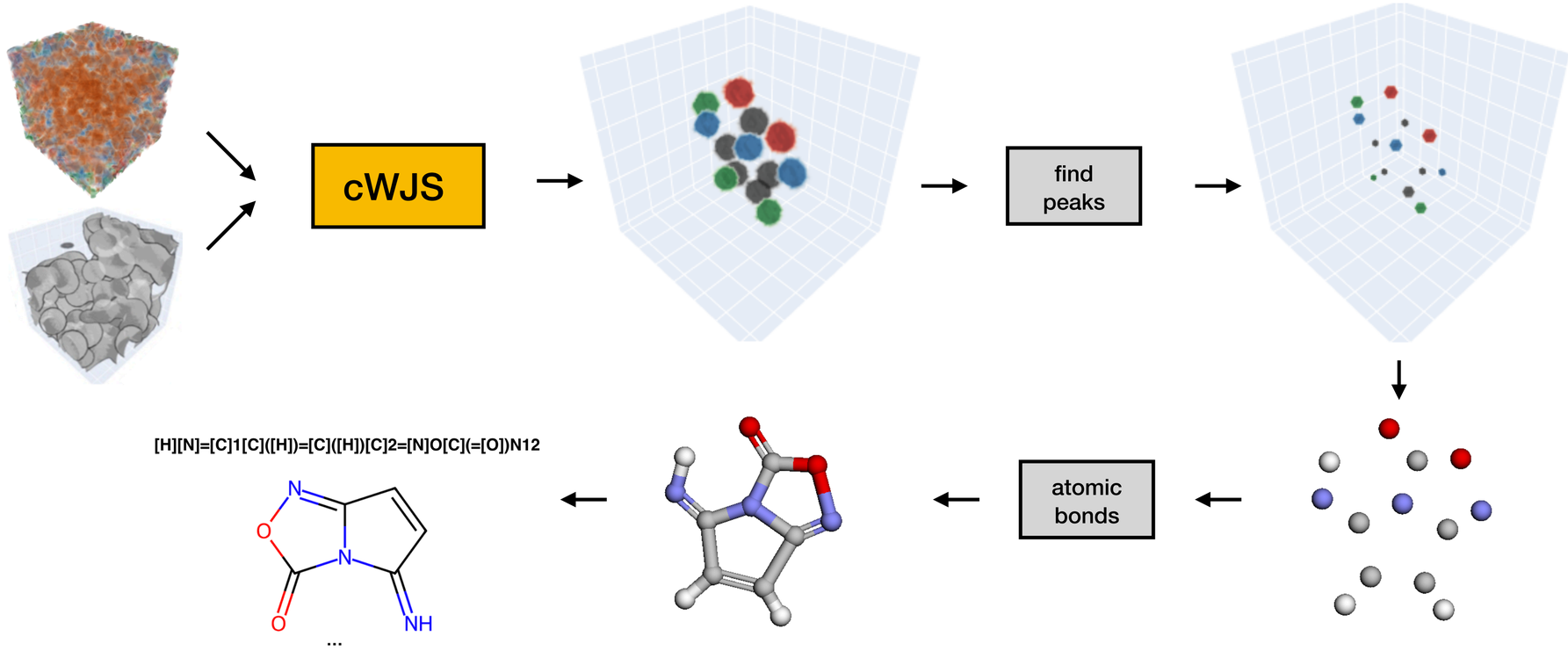
$$y_{t+1} = y_t + \delta g_\theta(y_t \mid \xi) + \sqrt{2\delta}\varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I_{d_x})$$

3. ***(jump)*** generate clean sample at arbitrary step k

$$\hat{x}_k = \hat{x}_\theta(y_k \mid \xi)$$

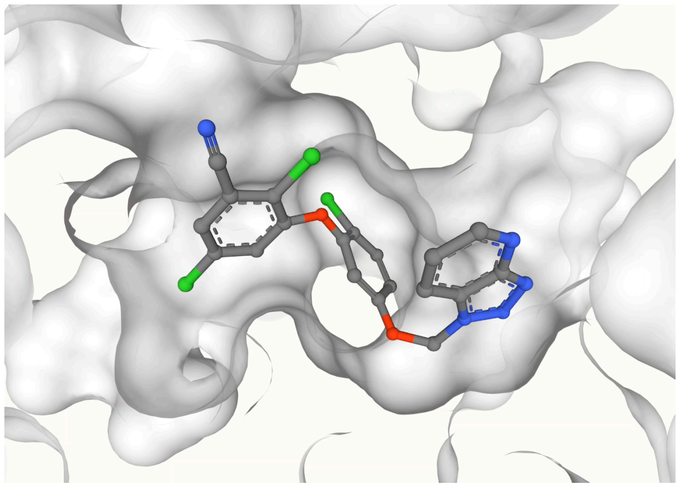
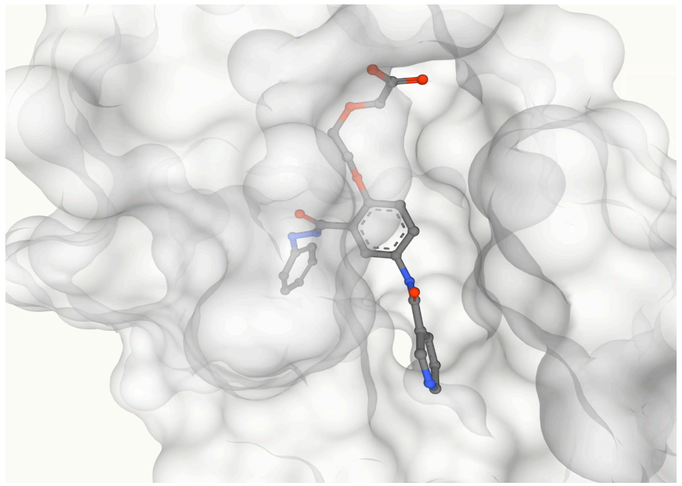
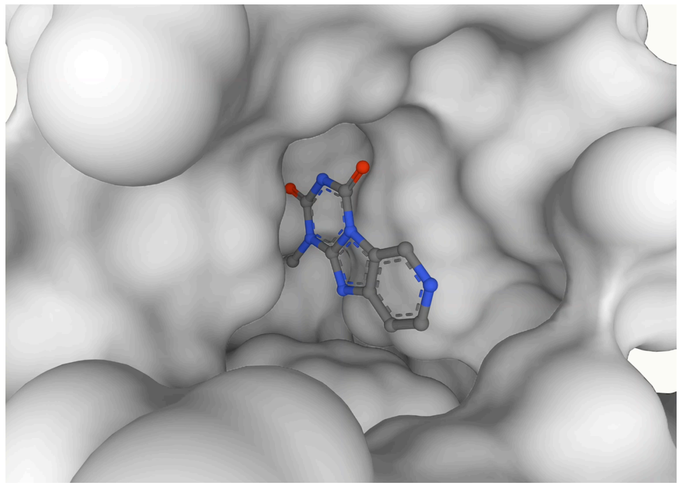
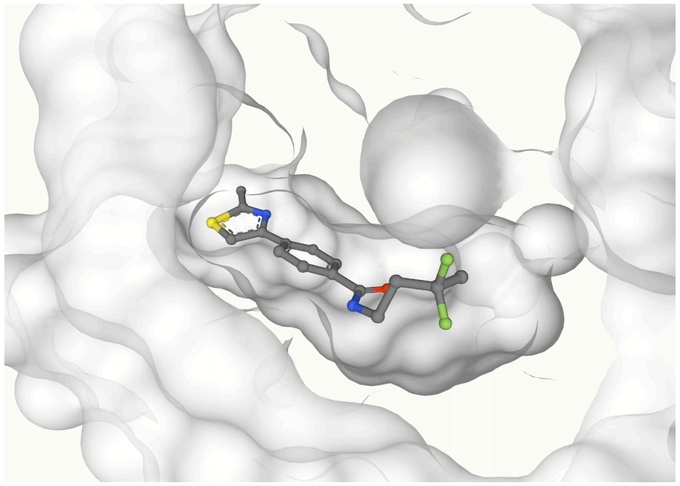
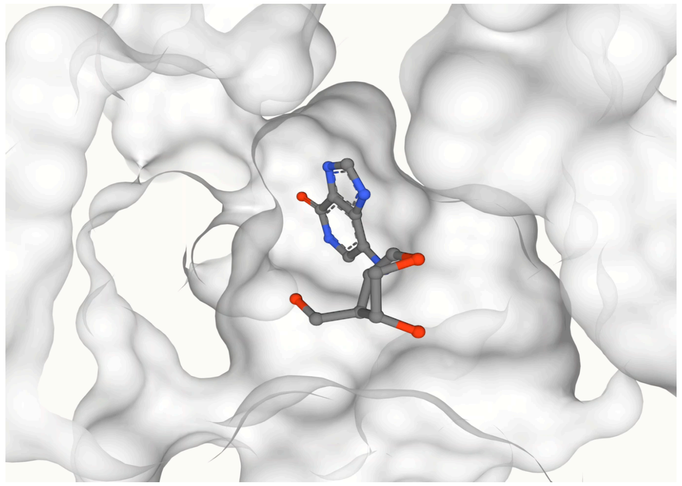
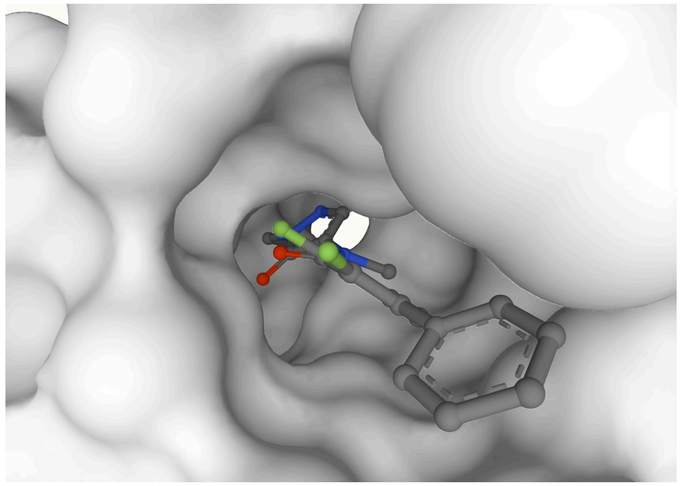
4. Voxel → atomic coords

Recovering atomic coordinates from voxel grids

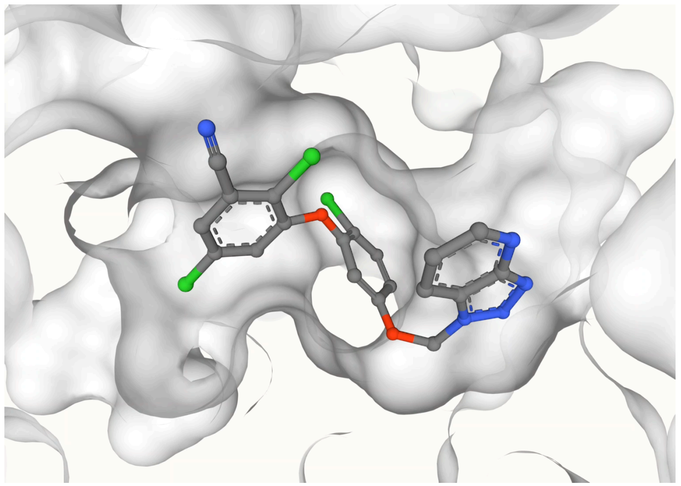
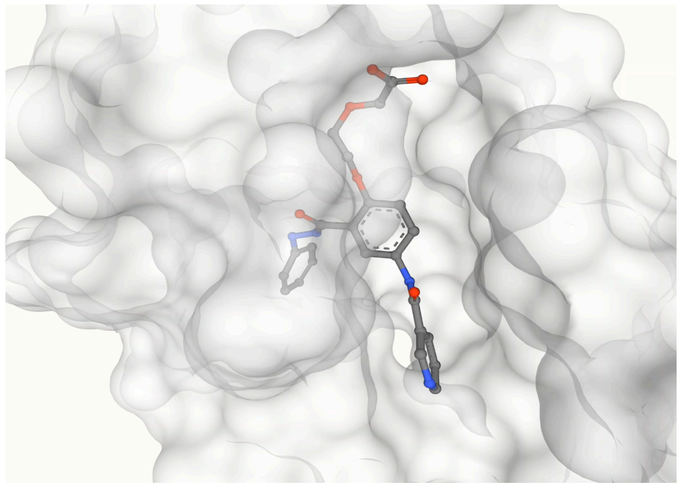
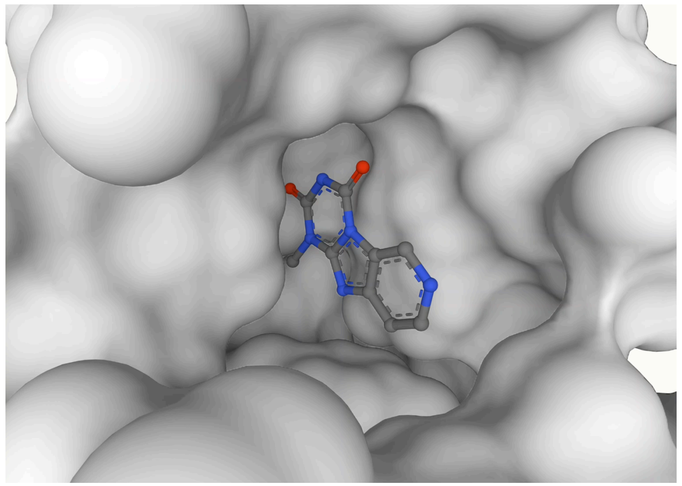
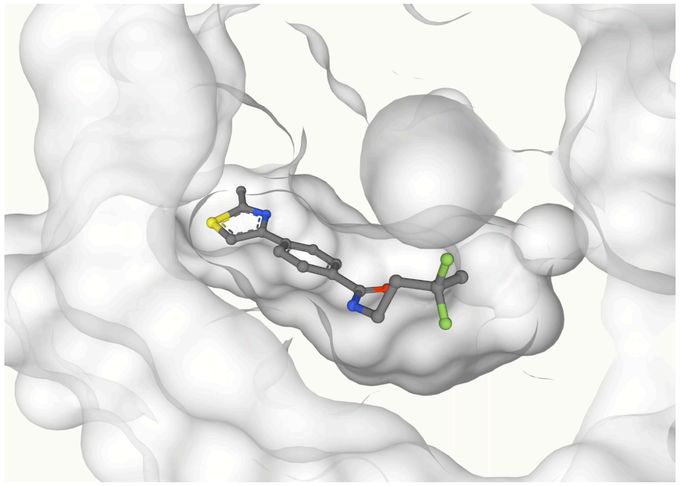
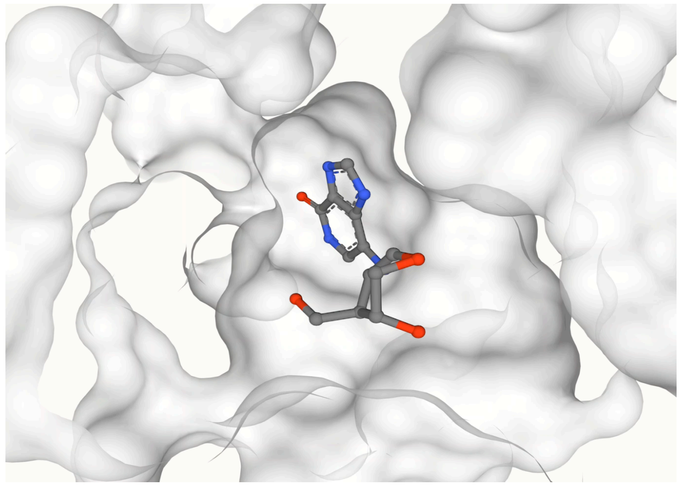
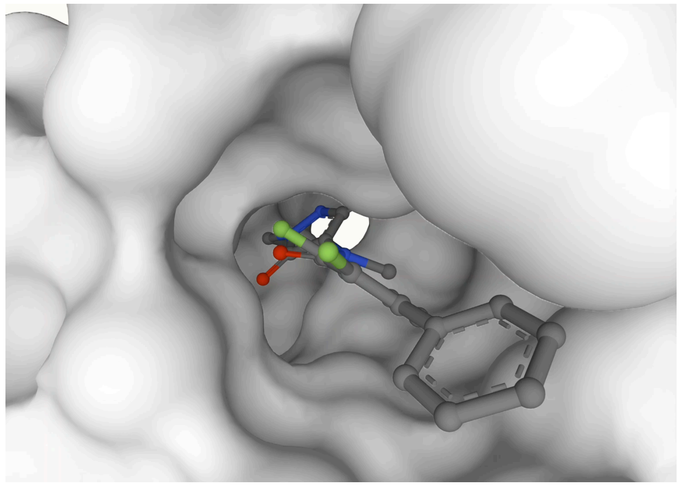


We use the same method as Pinheiro *et al*, 23

Examples of cWJS (single) chains

Examples of cWJS (single) chains

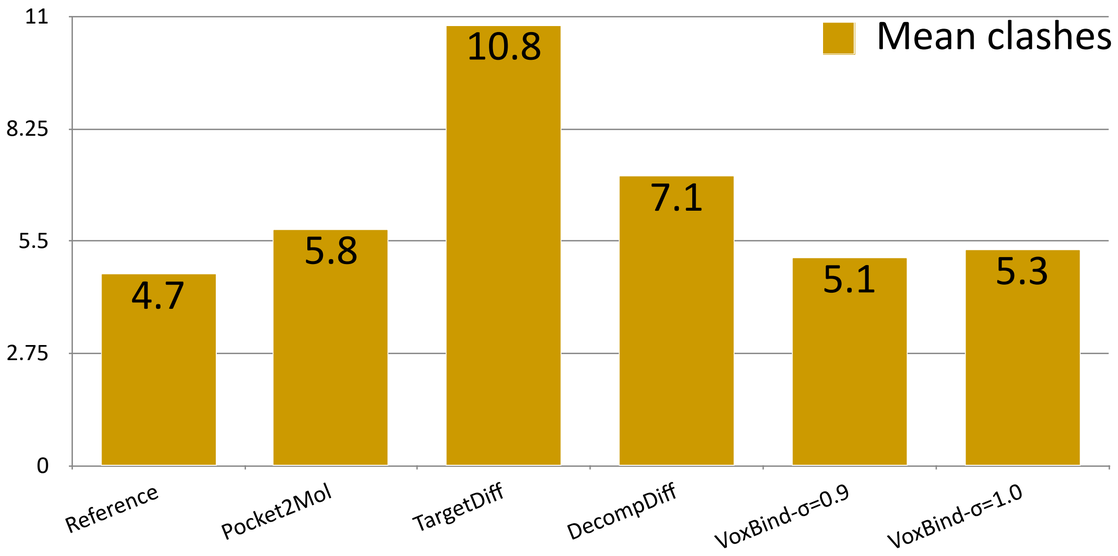
Experimental results

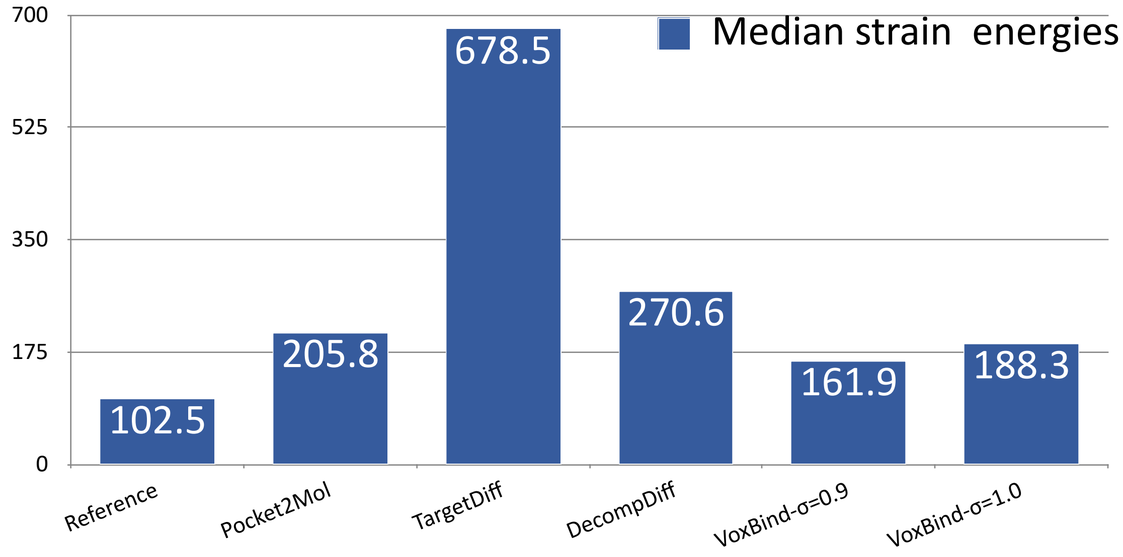
- **CrossDocked2020:** 100k/100/100 pocket-ligand binding pairs for train/val/test.
- Baselines taken from (Guan et al. 2023).
- Metrics averaged over 10k generated samples.

	VinaScore avg. (↓)	VinaMin avg. (↓)	VinaDock avg. (↓)	QED avg. (↑)	SA avg. (↑)	Diversity avg. (↑)	#atoms/ molecule	sec / mol avg. (↓)
Reference	-6.36	-6.71	-7.45	0.48	0.73	-	22.8	-
Pocket2Mol	-5.14	-6.42	-7.15	0.56	**0.74**	0.69	17.7	25.44
TargetDiff	-5.47	-6.64	-7.80	0.48	0.58	0.72	24.2	34.28
DecompDiff*	-5.67	-7.04	**-8.39**	0.45	0.61	0.68	20.9	61.89
VoxBind$_{\sigma=0.9}$	**-6.94**	**-7.54**	-8.30	**0.57**	0.70	0.73	**23.4**	4.92
VoxBind$_{\sigma=1.0}$	-6.63	-7.12	-7.82	0.55	0.69	**0.75**	21.7	4.92

Experimental results

- **PoseCheck metrics (Harris et al., 23)**
- (left) Mean steric clashes for each model (↓ better).
- (right) Median strain energy (kcal/mol - UFF) of molecules on their generated pose (↓ better).
- Metrics computed over 10k generated samples.

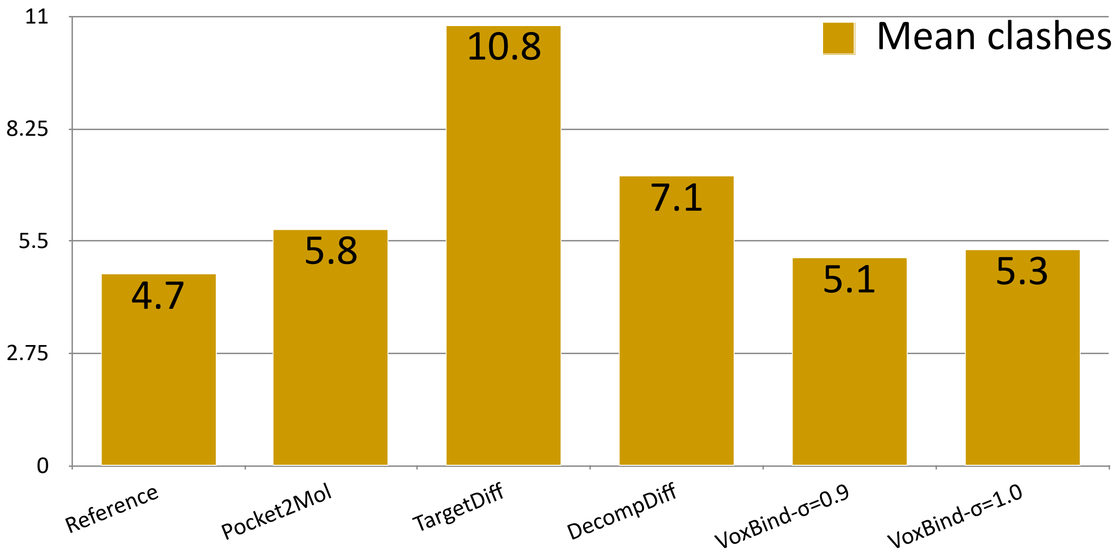


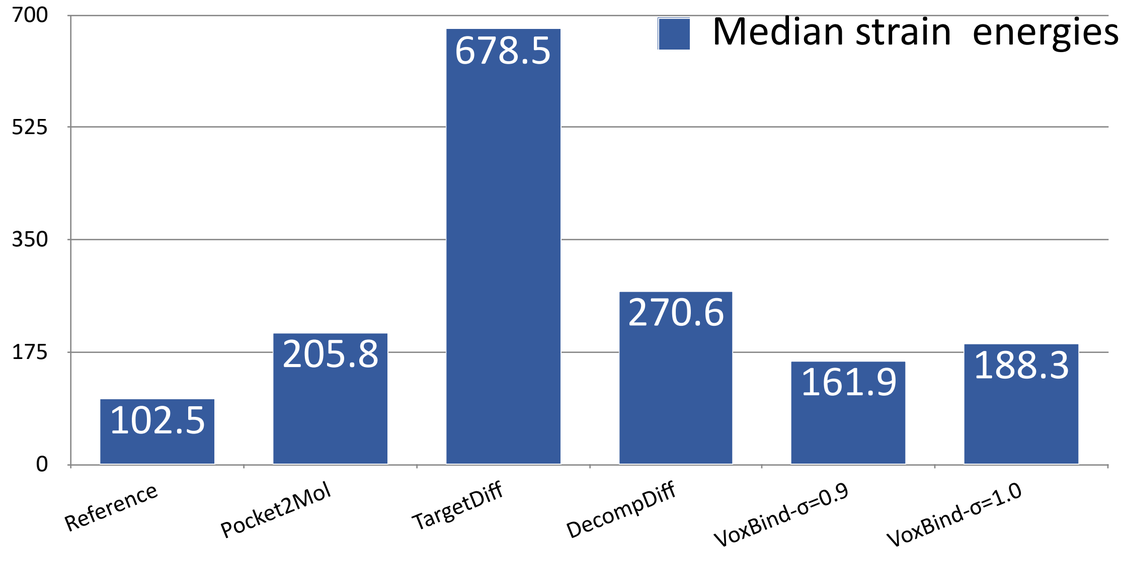


Experimental results

- **PoseCheck metrics (Harris et al., 23)**
- (left) Mean steric clashes for each model (↓ better).
- (right) Median strain energy (kcal/mol - UFF) of molecules on their generated pose (↓ better).
- Metrics computed over 10k generated samples.

More metrics on the paper!





Sampling efficiency

- From DecompDiff paper (Guan et al. ICML23):

> For the sampling efficiency, AR, Pocket2Mol, GraphBP, and TargetDiff use 7785s, 2544s, 105s, and 3428s for generating 100 valid molecules on average separately. It takes DecompDiff 5570s / 6189s on average without / with validity guidance. Similarly, the decomposed prior has a negligible impact on the sampling time. Bond diffusion results in 1.62x sampling time compared to TargetDiff, and validity guidance makes the sampling time slightly increase by 10% further.

Sampling efficiency

- From DecompDiff paper (Guan et al. ICML23):

> For the sampling efficiency, AR, Pocket2Mol, GraphBP, and TargetDiff use 7785s, 2544s, 105s, and 3428s for generating 100 valid molecules on average separately. It takes DecompDiff 5570s / 6189s on average without / with validity guidance. Similarly, the decomposed prior has a negligible impact on the sampling time. Bond diffusion results in 1.62x sampling time compared to TargetDiff, and validity guidance makes the sampling time slightly increase by 10% further.

VoxBind takes ~500 sec / 100 valid samples!

Conclusion

- New model for structure-based drug design inspired by **computer vision**
- **Better** results and **faster** sampling on standard benchmark
- **Expressivity >> built-in SE(3) equivariance**
- Code at: https://www.github.com/genentech/voxbind